Computational Interaction Frames

Michael Rovatsos

Institut für Informatik
der Technischen Universität München

Computational Interaction Frames

Michael Rovatsos

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr. Alois Knoll
Prüfer der Dissertation:		
	1.	Univ.-Prof. Dr. Dr. h.c. mult. Wilfried Brauer
	2.	Univ.-Prof. Dr. Hans-Dieter Burkhard (Humboldt-Universität zu Berlin)

Die Dissertation wurde am 24. Juni 2004 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 14. Dezember 2004 angenommen.

Zusammenfassung

In dieser Arbeit werden *Interaktionsrahmen* als neuartiger Ansatz zur sozialen Inferenz und zum Umgang mit sozialer Interaktion in Multi-Agenten-Systemen vorgestellt. Das soziologisch inspirierte Konzept der Interaktionsrahmen kann verwendet werden, um Kategorien von Interaktionsmustern zu repräsentieren. In Multi-Agenten-Systemen operierende und interagierende Agenten können mit Hilfe dieser Rahmen ihre Interaktionserfahrungen aufzeichnen und darauf basierend strategische Kommunikationsentscheidungen treffen.

Solch „sozial intelligente" *Rahmung*, die Rationalität im entscheidungstheoretischen Sinn mit empirischen Methoden zum Erlernen von Gesetzmäßigkeiten bezüglich der Kommunikation in einem Multi-Agenten-System kombiniert, ist besonders für *offene* Agenten-Gesellschaften geeignet. In solchen Gesellschaften kann die Einhaltung einer *a priori* festgelegten Semantik von Nachrichten und Kommunikations-Protokollen nicht vorausgesetzt werden. In Ermangelung absoluter Sicherheit in Bezug auf das Verhalten Anderer in zukünftigen Interaktionen, versucht der rahmenbasierte Ansatz, Unsicherheit durch Anspassung an beobachtetes Verhalten zu minimieren. Dies stellt eine erhebliche Verbesserung gegenüber der strikten Einhaltung vordefinierter Protokolle und sogenannter „Conversation Policies" dar, die – zumindest wenn man Agenten-*Autonomie* ernst nimmt – eine zu große Einschränkung darstellen kann.

Es wird zunächst die abstrakte Architektur InFFrA vorgestellt, die auf den Konzepten von Interaktionsrahmen und Rahmung beruht und als Meta-Modell für konkretere Agenten-Designs auf Basis dieser Konzepte verwendet werden kann.

Diese abstrakte Architektur wird durch das formale Modell einer konkreten, direkt implementierbaren Instanz von InFFrA ergänzt, die den Anforderungen des Meta-Modells genügt. Für dieses formale Modell kann auf der Grundlage des allgemeineren Konzeptes der *empirischen Semantik* von Agenten-Kommunikation eine formale Semantik für rahmenbasierte Kommunikation abgeleitet werden. Für diese formalisierte InFFrA-Variante werden darüber hinaus Entscheidungsmechanismen definiert und Lernalgorithmen in Anlehnung an Theorien des *hierarchischen „Reinforcement"-Lernens* entwickelt.

Eine konkrete Implementierung der formalisierten Architektur wird verwendet, um die Leistungsfähigkeit rahmenbasierter Agenten in einem realistischen Anwendungsszenario aus dem Bereich der agentenbasierten Webseiten-Verlinkung zu evaluieren. Die experimentellen Befunde stellen die Leistungsfähigkeit des Ansatzes unter Beweis und zeigen, dass Interaktionsrahmen ein mächtiges Werkzeug zum Umgang mit sozialer Interaktion in offenen Systemen darstellen.

Die breite Anwendbarkeit von Interaktionsrahmen wird schließlich durch die Darstellung weiterer Anwendungsmöglichkeiten aufgezeigt.

Abstract

This thesis introduces *interaction frames* as a novel approach for social reasoning and interaction management in multiagent systems. Interaction frames are a sociologically inspired concept that can be used to represent categories of interaction patterns. Agents operating and interacting in multiagent systems can employ these frames to record their interaction experience and to make strategic communication decisions based on this experience.

This kind of socially intelligent *framing* that combines decision-theoretic rationality with empirical methods for learning the regularities of communication processes in a multiagent system is particularly well-suited for *open* agent societies. In these societies, the adherence to an *a priori* semantics of messages and communication protocols cannot be taken for granted. In the absence of absolute certainty about the ways others will behave in future interactions, the frame-based approach relies on observation and adaptation of one's own behaviour to the observed patterns of interaction to reduce uncertainty.

This constitutes a significant improvement over the use of pre-specified communication protocols and conversation policies in a hard-wired fashion which can be too limiting, at least if we take agent *autonomy* seriously.

We present an abstract social reasoning architecture called InFFrA that is based on the concepts of interaction frames and framing and that can be used as a meta-model for concrete agent designs.

This abstract architecture is supplemented by the formal model of a concrete, ready-to-implement instance of the meta-model that complies with InFFrA requirements. For this formal model, we also establish a formal semantics based on a more general model of *empirical semantics* for agent communication. Furthermore, we define decision-making procedures for this formal version of InFFrA and develop learning algorithms that borrow from the theory of *hierarchical reinforcement learning*.

An implementation of the formal social reasoning architecture is used to evaluate the performance of frame-based agents in a realistic application scenario taken from the domain of agent-based Web linkage. The experimental results prove the effectiveness of our approach and show that interaction frames can be successfully used as a powerful tool for reasoning about interaction in open systems.

Finally, the broad applicability of frames is illustrated by a discussion of various further applications.

Acknowledgements

Above all, I wish to thank my supervisor Prof. Wilfried Brauer who has contributed to this thesis in many ways. First of all, by providing a highly productive and pleasant working environment with enough room for every idea to grow. His unswerving belief that students and collaborators should be given as much freedom as possible to pursue their own research directions has been confirmed by the high quality of the research done in his group – I hope that this thesis is yet another piece of work that has benefited from this liberal spirit. Also, his advice and his often unconventional outlook on things, stemming from his tremendous experience in computer science and artifical intelligence, has been immeasurably valuable and helpful over the years.

Also, I wish to thank Prof. Hans-Dieter Burkhard for his willingness to act as a second referee and to devote a lot of time to reading and commenting on the thesis. The feedback he provided on earlier drafts significantly contributed to improving the structure and overall presentation. As it is not an easy task to deal with material based on the research one was not directly involved in, I appreciate his effort very much.

A number of people have actively contributed to the research described in this thesis as close collaborators. Gerhard Weiß, leader of the AI/Cognition group has contributed through his wide and profound knowledge of the field of multiagent systems and also by being open to new ideas (at peak times, we would come up with the idea for a new field of research every other week). Matthias Nickles, my co-investigator in our Socionics research project had a similar influence, albeit by expressing views that were often very different from mainstream multiagent systems research. His often radical ideas lead to endless but very fruitful discussions that helped shape many novel concepts and perspectives on seemingly clear and well-understood research issues.

A major part of the research presented here was carried out in the project “Conflict resolution and structural change: social theory as a ‘construction manual’ for adaptive multiagent systems” which made up part of the Socionics priority program. I am grateful to the German National Research Foundation DFG for funding this program and our project (and for paying my salary, of course). In the context of this project, our sociological collaborators from the Technical University of Hamburg have introduced us to social theory, but also to new ways of thinking computer scientists are rarely confronted with. Kai Paetow and Marco Schmitt deserve my gratitude for introducing me to the works of G. H. Mead and E. Goffman, but also for backing our interdisciplinary work with in-depth sociological expertise. And for being very patient with answering each and every question of mine.

Several students have directly contributed to the work presented here. Marco Wolf realised the first real system based on the principles of interaction frames and framing. Felix Fischer assisted in the development of the formal semantics of the InFFrA architecture. He also implemented the learning algorithms, conducted some of the simulation experiments, and was an invaluable source of help in the final stages of implementation and testing. Stanislavs Bardins, Johann Duscher and Christian Pacher implemented parts of the LIESON simulation testbed used for evaluating the usefulness of our approach. Without their efforts, many of the ideas this thesis is based on would never have been turned into real software.

A big “thank you” goes to Michael Schillo from DFKI, for being a stern critic in scientific matters (and a very good friend in all other respects). Jürgen Lind, after introducing me to multiagent systems while being my masters’ thesis advisor, taught me that “complexity remains”. This insight has – contrary to his original intentions, probably – spurred my drive to master complexity and this thesis is an account of another step in this direction.

Volker Baier, Dieter Bartmann, Claudia Brand, Felix Brandt, Clemens Kirchmair, Alexandra Musto and Klaus Stein, my fellow (present and former) colleagues in the AI/Cognition group always provided a pleasant working environment (and ensured high catering quality at group parties and other gatherings). It has also been a pleasure to share that same environment with members from the theoretical computer science group, especially Markus Holzer, Stefan Katzenbeisser, Astrid Kiehn, Barbara König, Peter Rossmanith, Christian Schallhart and Helmut Veith despite the fact that their “hard” science was a constant threat to our “soft” AI methods.

I should also not forget to thank colleagues from the multiagent systems community for providing valuable interaction through e-mail discussions, meetings at conferences and workshops, etc. Some names that come to my mind are: Stuart Chalmers, Kobi Gal, Fabien Gandon, Claudia Goldman, Franziska Klügl, Emma Norling, Andrea Omicini, Lin Padgham, Adrian Pearce, Enric Plaza, Alessandro Ricci, Mirko Viroli and Osher Yadgar.

The members of the Agents Group at the University of Melbourne and, in particular, Liz Sonenberg and Iyad Rahwan helped make my research stay with them very productive and inspiring. My participation in several conferences was co-sponsored by the European Network of Excellence “AgentLink”, and the research visit to Melbourne was generously supported by the Leonhard Lorenz Foundation. The anonymous reviewers of MAAMAW’01, AOSE’01, AAAI Spring Symposium ’02, ECAI’02, AAMAS’02, AAMAS’03, AAMAS’04, IJCAI’03, MATES’03, CIA’04, and ECML’04 have contributed through their valuable comments on papers, which have certainly added to the quality of the results presented in this thesis. Thanks to Felix Brandt, Matthias Nickles, Oliver Plaehn, Michael Schillo, Gerhard Weiß, and, in particular, Felix Fischer, for a great deal of proof-reading. Finally, my family have continued their relentless support in whatever I pursue, for which I will always be indebted to them.

Contents

List of Figures

List of Tables

Once you eliminate the impossible, whatever remains, no matter how improbable, must be the truth.

Sherlock Holmes

1. Introduction

The past decade has witnessed dramatic changes in the way we see computing. After information technology had been continuously evading all areas of everyday life in the 1980s and early 1990s, the advancement of the Internet and of other digital communication technologies marked an unexampled leap towards a new era: the age of global computer communications. Apart from the impact of these changes on society in general, they certainly also bear strong implications on the way we think about computers and software. If we compare, for example, typical software systems twenty or thirty years ago to those of today, striking differences become obvious with respect to the requirements imposed on these systems. In those days, it was perfectly acceptable for a system to perform the tasks assigned to it in a particular application domain (which is a difficult thing to achieve in its own right). Today, we additionally expect of many systems to be available online on a permanent basis, to provide intuitive and robust user interfaces, to interoperate with other applications (such as legacy systems, online databases or services), to ensure security of internal data against malicious attacks, etc. More generally speaking, we are increasingly interested in building systems that are able to operate in *dynamic*, uncertain domains and that are able to *interact* with users and other systems effectively.

The *agent* metaphor is often considered useful for this kind of software systems. It marks a shift from the traditional picture of systems as relatively closed and fairly deterministic, controllable engineering artifacts to a perspective that views software components as autonomous entities. Agents perceive and act upon an environment, and they communicate and interact with other agents and humans in *multiagent systems* (MAS) in pursuit of their goals. It is widely believed that the ability to implement intelligent agents capable of operating in complex environments will contribute substantially to coping with the requirements of software in a “connected” world.

This thesis is about *interaction frames* for computational agents. Interaction frames are a novel, sociologically inspired concept which can be used to endow agents with knowledge about different types of interaction situations. This knowledge can be used to guide agents' behaviour when interacting with others by improving their *social reasoning* capabilities. Apart from its intuitive appeal and its implications for furthering our understanding of social cognition, our approach constitutes one possible answer to the practical problems mentioned above. It allows agents to manage their interactions effectively in dynamic, complex environments inhabited by other agents. Thus, it significantly contributes to research on *open systems* that will become increasingly important in the future and confront the field of computing with new challenges.

We present an agent architecture based on interaction frames, develop a formal model for a concrete instance of this architecture, and provide algorithms for reasoning and learning in frame-based agents. Simulation experiments in complex domains underpin the usefulness of our approach, and a variety of applications is discussed together with concrete examples.

1.1 An Illustrative Example

Before describing our approach, its main contributions and its relationship with other research themes, we would like to motivate it with a simple yet illustrative example for the class of applications mentioned above: the *web linkage* world.

The reader is certainly familiar with the way hyperlinks are used in World Wide Web pages to refer to other sites. Although these links provide a simple and natural way to link Web content, readers will agree that, most of the time, this linkage is not realised in an optimal way. Rather than semantically linking related content (information items, opinions and comments on similar topics, related individuals or institutions) together, links often exist between seemingly unrelated pages. On the other hand, those links that would be useful when looking for relevant information are missing. A good example for this suboptimal linkage is the usual habit of personal homepages to provide pointers to weather forecast sites, press and media or humour sites. Although these links may occasionally inform a reader about sites he had not known of, they are, in general, utterly useless for a Web user seeking specific information.

Since searching the Web for related sites and creating respective links is a rather tedious business for the Web site owner (let alone keeping track of modifications to the link targets), it would be desirable to automate this task, at least partially. Ideally, this would contribute to the evolution of a so-called Semantic Web (Berners-Lee et al. 2001) by making semantic relationships between sites transparent.

An agent-based approach seems suitable for such an application. Each Web site owner would be represented by an agent, and this agent would search for appropriate sites, decide whether to lay links towards these sites and monitor their evolution to revise previous decisions, if necessary. However, as high Web traffic is considered an asset particularly by commercial sites, it would be irrational for such agents to lay links unless it is also profitable for their own site. Therefore, such a linkage agent will be interested in influencing other Web site owners' behaviour, and this can best be achieved through communication.

Once the need for such interaction has been recognised, it becomes obvious how this application example fits into the class of open, dynamic multiagent systems in uncertain domains: Firstly, a linkage agent (and with it, its designer) does not know anything about the internal structure, the motives and reasoning capabilities of other agents. Therefore, it cannot be designed to respond appropriately to a fixed behaviour on the side of other agents. Secondly, the search space of all potential linkage structures, even among a limited number of sites is gigantic, and such structures are volatile, as links may be created and deleted arbitrarily often at virtually no cost. Furthermore, the society of agents taking part in this quest for optimal linkage is neither fixed nor limited in size, but open to new participants. In fact, in this particular application, it is not even possible to exclude anyone from linkage activities except by legal intervention in very extreme cases. So, an agent can merely try to achieve *satisficing* (rather than optimal) behaviour, especially because the social exchange required to achieve coordinated action incurs an additional computational cost.

Interaction frames come into play when we think about how this complexity, that is induced by the *autonomy* of components and by the *openness* of the environment, can be dealt with. Not knowing what other agents will do, it seems reasonable to design our agent in such a way that he observes their behaviour and tries to make sense of it, with the goal of adapting his own behaviour in a way that will allow for achieving optimal linkage with others. Since the agent is interested in aspects of others' behaviour that he can influence by communicating with them, he will concentrate on learning something about communicative (rather than general) behaviour. Assuming that other agents might do the same, it is also reasonable to observe regularities in one's own behaviour, since peer behaviour may be a reaction to it. As the computational resources of the agent will be limited, he will have to categorise the observed behaviour to determine which differences between individual behaviours matter and which do not. If the linkage situation necessitates completely new behaviours, the agent will have to devise new forms of communication to achieve this.

This is precisely what agents who employ interaction frames and framing do, and what this thesis is about:

> *The learning and strategic use of categories of interaction patterns by socially intelligent agents in open systems, in which no information is available about agents' future behaviour other than what is known from interaction experience.*

The Web linkage scenario will serve as an example throughout the remaining chapters to illustrate different aspects of the methods presented. In fact, it also constitutes the application scenario that has been implemented in the LIESON system to evaluate our findings (cf. chapter 6) and is therefore much more than a hypothetical scenario: a representative and concrete example of a new kind of socially adaptive, agent-based, real-world applications.

1.2 Main Contributions

This thesis advances the state of the art in multiagent systems research in different ways. Before embarking, we would like to list the most important of these. Clearly, these contributions also relate our work to a number of other research themes, for a discussion of which we refer the reader to section 2.3.

Abstract social reasoning architecture

We propose a new social reasoning and learning architecture called InFFrA (the Interaction Frames and Framing Architecture) that serves as an abstract model for building socially intelligent agents who construct, maintain and modify interaction frames and apply them strategically to achieve effective interaction behaviour.

InFFrA is based on sociological theories from the school of *symbolic interactionism* (Blumer 1962, Strauss 1993). More specifically, the concepts of *interaction frames* and *framing* are derived from Erving Goffman's micro-sociological theories (Goffman 1974) and the formation and employment of social expectations is based on George Herbert Mead's general theory of social action (Mead 1934) in which symbolic interactionism as a whole is deeply rooted. This connection is not only interesting with respect to interdisciplinary exchange.[1] Also, the fact that these theories are well-established in sociology underpins the plausibility of using them to design computational architectures resting on their principles, such as InFFrA .

[1] Much of the research presented here has been conducted in the context of the Socionics research program (Malsch 1998, Malsch, Müller and Schulz-Schaeffer 1998, Malsch 2001, Fischer and Florian 2003).

The InFFrA architecture can be used as a conceptual model for developing concrete frame-based systems. It suggests a general model of interaction frames and identifies the central functional elements that are needed to implement intelligent framing processes. Thus, the agent designer is provided with guidance regarding

1. the components of the design he should focus on when pursuing a frame-based approach,

2. how these components have to be combined to obtain comprehensive frame-based social reasoning capabilities,

3. the key design decisions and critical aspects of this design process.

To the best of our knowledge, InFFrA is the first abstract social reasoning architecture that deals with learning and applying regularities in "high-level" agent communication, meaning communication in which utterances can explicitly refer to non-communicative objects, such as physical objects, actions, agents and mental states (goals, plans, intentions etc.).

In that, it is deliberately designed to exploit the full potential of all relevant modelling dimensions, which results in a complex framework that allows for a wide range of possible implementations. As a natural consequence of this complexity, InFFrA does not lend itself to direct implementation but is rather suitable as a conceptual foundation for the development of concrete architectures. Thus, while it explores the landscape of "what can be modelled using frames and framing" it is clear that any concrete system based on it will have to focus on a subset of these possibilities through specific design decisions. The suitability of InFFrA for this purpose is underlined by the fact that it has been successfully applied to the analysis and design of systems (Rovatsos, Weiß and Wolf 2002) other than the formal model and implementation presented in this thesis (chapters 4, 5 and 6).

Beyond these practical considerations, the development of InFFrA also has some theoretical implications for our understanding of human social cognition, and thus for the field of artificial intelligence (AI) in general. Furthermore, the computational operationalisation of sociological theories may, in turn, have an impact on sociological discourse by making distinctions precise and using (semi-)formal representations. However, a philosophical or sociological treatise of these implications is not provided here. We refer the interested reader to (Paetow and Rovatsos 2003) for a more principled and general discussion of some of these aspects.

Formalisation of a concrete instance of this architecture

From the standpoint of computer science or artificial intelligence, an abstract architecture is of little value, as long as it cannot be turned into a computational model. Only fully formalised models can be shown to have certain properties, either through mathematical analysis (proofs) or through empirical evaluation (simulation and case studies).

In the present thesis, such a formal model is provided through m^2InFFrA[2]. m^2InFFrA realises the principles of InFFrA in a ready-to-implement fashion.

[2] The "m^2" stands for "double-Markov" and is due to the relationship between this formal model and the two-level Markov Decision Process (MDP) view of framing it provides. See also chapter 5.

Thereby, m^{2}InFFrA highlights certain elements of the abstract architecture while others are only present in the most basic form. For this reason, the fully formalised architecture must be regarded as a particular *instance* of what is possible in principle when using InFFrA as a basis for the development of concrete frame- and framing-based architectures and is certainly not a complete formalisation of InFFrA.

m^{2}InFFrA is characterised by the following distinctive features:

1. m^{2}InFFrA agents are *knowledge-based* agents that maintain and manipulate symbolic representations of their environment to reason about the world. As they use fairly common inference mechanisms and logical representations, this adds to the applicability of the model in different domains.

2. Communication in m^{2}InFFrA takes place at a speech-act level. Messages are perceived as publicly visible actions with performatives related to their desired outcomes, and by being parametrised with logical content they enable agents to refer to objects outside communication (physical objects, agents, actions, mental states, etc.).

3. The communicative expectations recorded in m^{2}InFFrA frames have probabilistic semantics defined in terms of the potential consequences of certain communicative actions. This enables us to use decision-theoretic principles and reinforcement learning methods for decision-making in m^{2}InFFrA.

4. The probabilistic semantics employ "reasoning by analogy" since frames generalise from particular interaction experiences, and "contextualisation" since certain frames are only relevant in certain situations. m^{2}InFFrA realises these principles in a fashion that resembles *case-based reasoning* (Kolodner 1993, Aamodt and Plaza 1994, Watson and Marir 1994).

The advantages of such a formal model are that (i) mathematical rigour necessitates clarification of concepts that are (inevitably) general and vague in the abstract architecture, (ii) it enables us to describe concrete reasoning and learning algorithms (and, ultimately, to implement them) and (iii) it allows connecting the frame-based approach to traditional AI methods. As concerns this last aspect, the relationship to reinforcement learning (Sutton and Barto 1998, Kaelbling, Littman and Moore 1996) and in particular to hierarchical reinforcement learning methods (Barto and Mahadevan 2003) places our work in the context of contemporary mainstream AI research (and shows how the notions we introduce can be grounded in established concepts).

Implementation and evaluation in a realistic application domain

As described in the opening paragraphs of this chapter, the foremost aim of our research is to develop methods that may help to tackle the problems of complex, open application domains. This "engineering" stance prohibits experimentation with our methods in toy domains, although these domains are very valuable for more theoretical contributions.

To make sure we can adequately cope with the problems of frame-based architectures that arise in practice, we have implemented a full-fledged simulation system for the aforementioned web-linkage scenario. This system called LIESON (cf. section 6.1) is a distributed testbed that simulates a multiagent system populated with communicating agents and a hypothetical web linkage environment in which these agents are embedded. It captures the essential aspects of the domain and reflects the difficulties associated with achieving fruitful coordination in it: incomplete information regarding the linkage status of sites, volatility of links, heterogeneity in the preferences, motives and experience of different Web site agents, to name but the most important.

Focusing on the practical side of frame-based methods bears several methodological implications:

- *Algorithms have to be applied in a computationally feasible way*
  In practical implementations, the *bounded rationality* aspect is not merely theoretical – many algorithms, though correct in principle, impose time and space constraints that are simply too prohibitive to be used in real-world application.[3] Hence, we are forced to

  1. use fairly simple logics and inference procedures, because we require that they can be *used* by InFFrA agents themselves. In contrast to many logic-based approaches in AI which use logic to describe system aspects from the designer's viewpoint, this means that we use an "internal" rather than an "external"[4] approach to logic.
  2. apply heuristics where complete solution methods are intractable. For example, we do not maintain the full state spaces for reinforcement learning algorithms but employ abstraction heuristics to reduce them. Also, limits in frame storage capacity of the implemented agents force us to use heuristics to decide which frames to ignore and which to store in the long term.

- *Mixture of different methods*
  Since we are building "full" agents with complex social reasoning (but also other) capabilities, we must combine different methods to make the individual components work together. For example, LIESON agents will have to perform planning and decision-making steps that are not directly related to social reasoning. Therefore, a local planning and goal-directed reasoning component has to be included in their implementation (in our case, a BDI (Bratman, Israel and Pollack 1988) architecture).

  Strictly speaking, this module does not make part of the architecture we have devised, but integrating different components to yield a comprehensive application system is a great challenge that has to be faced when building real systems.[5]

We believe that the advantages of realism at the level of practical applications for the acceptance of architectures as the one proposed here, but also of the MAS paradigm in general, clearly outweigh the restrictions listed above.

The relationship between the LIESON system and the m^2InFFrA model bears some similarity to that between m^2InFFrA and the abstract InFFrA framework. Again, not everything that is possible within m^2InFFrA has been experimented with, and – although the agents

[3] Some would say that this is particularly true of AI methods.
[4] We use the terminology of Fagin et al. (1995) here.
[5] This is part of the reason why die-hard followers of any of the methods we employ (decision theory, reinforcement learning, case-based reasoning) will probably find that neither of these techniques has been exploited to the maximum. Where necessary, "making it work" was given priority over "being optimal".

implemented in LIESON are full-fledged m^2InFFrA agents – we can only validate the architecture for particular kinds of interactions (namely, negotiation) and social settings. This iterative "narrowing down" of the scope of material covered from chapter 3 to chapter 6 constitutes an essential aspect of our methodology, as it shows how starting from the most abstract sociology-inspired concepts we can manage to produce concrete software that solves a practical problem.

Exploration of new approaches to multiagent learning

Learning interaction frames, which is one of the main activities in frame-based social reasoning, adds a new perspective to *multiagent learning* (Weiß 1996, Weiß and Sen 1996, Weiß 1997, Weiß 1998, Weiß and Dillenbourg 1999). This perspective lies in regarding classes of interaction as the object of learning, rather than the behaviour of (groups of) particular agents. The underlying assumption is that, in open, large-scale MAS it does not pay to learn something about individuals, since encounters with them will be often only occasional. Instead, agents should be concerned with learning something about recurring patterns of interaction that apply in a society regardless of concrete interaction partners.

This view has lead to the formulation of two novel properties of multiagent learning methods that have been largely overlooked in the current research landscape:

1. *Social abstraction*
   Models of agent behaviour must abstract from details concerning instances of certain interaction patterns in order to cover a variety of new cases. They must generalise over concrete agents and interaction situations to express "societal regularities" that apply throughout an entire social context.

2. *Transient social optimality*
   To ensure the reliability of established social procedures, they have to be enacted at the cost of sacrificing profit that would be possible if an agent optimised his behaviour constantly. This transient optimality (in contrast to *permanent* or *total* optimality) is necessary, because agents must fit into a certain generalised set of expectations themselves to be understandable for others. Thus, different from agent-based learning, acting in consistence with generalised expectations in frame-based learning implies giving up the ideal of achieving optimal outcomes for oneself at every single point in time.[6]

What is most interesting about these notions, is that they not only parallel views commonly held in sociology, but that they result naturally when using interaction frames as building blocks for learning algorithms.

A new theory of agent communication semantics

Frames involve communication, and, in a frame-based approach, communication only obtains its meaning through its use in the context of certain interaction frames. We contribute to the field of agent communication language (ACL) semantics by claiming that the

[6] At least this is the case if we adopt an *active learning* view, in which agents have to act during "online" learning. If learning merely consists of passive observation, utility-based rationality criteria are not jeopardised by transient social optimality.

meaning of inter-agent messages is given by the expectations associated with these messages as represented by the interaction frames agents have at their disposal.

This outlook on ACL semantics is radically *constructivist* as it only depends on the views of the parties involved in communication, *empirical* since these expectations are formed on the grounds of past interaction experience and *consequentialist* since – no further *a priori* semantics assumed – all that matters to agents regarding communicated messages are their consequences. Such an outlook on ACL semantics necessitates an *evolutionary* view of semantics, since agents may alter the consequences of utterances through their actions in the course of interaction.

Also, as we will see, our view differs largely from most work done in the area of agent communication language (ACL) research (Labrou, Finin and Peng (1999), Kone, Shimazu and Nakajima (2000), Dignum and Greaves (2000), and Chaib-draa and Dignum (2002) provide recent overviews) and thus constitutes a major contribution to the field (although it is just a "by-product", so to speak, of our endeavour to build socially intelligent agents).

To sum up, this thesis presents an abstract social reasoning architecture, a formal model of a specific instance of this architecture and an implementation in a realistic application domain that serves as the basis for the empirical evaluation of our approach. Methodologically, these steps follow the "standard procedure" commonly employed in artificial intelligence. The implications for multiagent learning and agent communication semantics, however, stand in contrast to this "engineering intelligence" stance. They constitute insights that lie beyond the original aims of this work, and point at new directions for research towards which this thesis is only a first step.

1.3 About This Thesis

The remainder of this thesis is organised as follows:

Chapter 2 provides the background on the two fields that are most relevant to our work. After a brief introduction to Distributed Artificial Intelligence and in particular to open systems and issues of agent interaction and coordination, we discuss elements of *interactionist socionics*, i.e. the application of interactionist social theories to multiagent systems. This chapter also contains an overview of related research themes.

Chapter 3 lays out the abstract social reasoning architecture InFFrA in full detail. The architecture can be used as a schema for building socially intelligent agents that employ interaction frames and framing to coordinate their activities with those of other agents.

Chapter 4 introduces m^{2}InFFrA, the formal model of a specific instance of InFFrA with probabilistic communication-predicting semantics. Using this formal foundation enables us to link InFFrA to the theory of Markov Decision Processes and hierarchical reinforcement learning.

Chapter 5 describes the learning algorithms and heuristics that are later used in the implemented m^{2}InFFrA agents. This chapter also formalises the notions of social abstraction and transient social optimality, which become evident in our use of hierarchical reinforcement learning algorithms.

Chapter 6 presents extensive experimental results obtained with the LIESON system. The LIESON system itself is, of course, also described in detail in this chapter. Furthermore, we report on the development of different kinds of negotiation frames for our application scenario.

Chapter 7 discusses further application areas of interaction frames and framing that have already been explored, and others that have a great potential and deserve our attention in the future.

Chapter 8 summarises our main results, gives an outlook on possible future work and closes with some general conclusions.

Note

Some of the material presented herein has been published and presented elsewhere before. A first version of the InFFrA architecture was outlined in (Rovatsos 2001), and shorter overviews appeared in overviews of InFFrA provided in (Rovatsos and Weiß 2001, Rovatsos et al. 2002, Rovatsos, Weiß and Wolf 2003b, Rovatsos and Paetow 2004).

The empirical semantics view and our model of communication was first articulated in (Rovatsos, Nickles and Weiß 2003a) and (Rovatsos, Nickles and Weiss 2004); elements of it also appeared in (Nickles and Rovatsos 2004). The notions of social abstraction and transient social optimality were coined in (Rovatsos and Paetow 2004). Our account of the formal semantics of m^2InFFrA is based on the model introduced in (Rovatsos and Paetow 2004). The general foundations of interactionist socionics were discussed at length in (Paetow and Rovatsos 2003).

The web linkage application scenario originates in ideas first laid down in an unpublished internal memo (Rovatsos 2000). A longer and much more detailed research report (Malsch, Paetow and Rovatsos 2002) extended these ideas and discussed the scenario from both a sociological as well as a multiagent systems perspective. A technical description of the LIESON system itself has been provided in the LIESON manual (Rovatsos 2002–2004) which is available online and is being constantly updated.

Many of the learning algorithms and heuristics were worked out in cooperation with Felix Fischer, and some experimental results have already appeared in his diploma thesis (Fischer 2003).

As concerns other applications, the ADHOC system for opponent classification was developed by Marco Wolf (2002) in his diploma thesis. Initial experiments with ADHOC appeared in (Rovatsos and Wolf 2002), and a more detailed account of the results was given in (Rovatsos et al. 2003b). This system was also used as a case study for the usefulness of InFFrA as a method for analysis and design in (Rovatsos et al. 2002). The relationship to the *communication systems* framework was first formalised in (Nickles and Rovatsos 2004) and (Nickles et al. 2004b), and later improved in (Nickles, Rovatsos and Weiss 2004a). The EXPAND framework is described in (Brauer et al. 2001), the RNS schema in (Nickles, Rovatsos and Weiss 2002) and (Weiß, Rovatsos, Nickles and Meindl 2003).

2. Background

This chapter provides an introduction to the two foundations of our research. We first outline some core elements of the field of Distributed Artificial Intelligence (DAI) and Multiagent Systems (MAS) with a particular focus on (i) autonomy and open systems and (ii) interaction, communication and coordination, since these are the research issues to which our methods contribute.

Then, we introduce *interactionist socionics*, an emergent sub-discipline that grew out of the Socionics research endeavour and that our work is based on. We give a short introduction to interactionist sociology (more specifically, to the theories of G. H. Mead and E. Goffman), review central assumptions made in multiagent systems that borrow from these theories and discuss implications for the design of these systems that result from this interdisciplinary view. Finally, related research themes are surveyed and their relationship to the work presented here is described.

2.1 Distributed Artificial Intelligence

According to Weiß (1999),

> DAI is the study, construction, and application of multiagent systems, that is, systems in which several interacting intelligent agents pursue some set of goals or perform some set of tasks.

To understand how the work presented here relates to this field, we need to define what agents are, how they interact in multiagent systems (MAS), and which problems and research questions the interaction frames and framing approach addresses in this context.

The presentation contained in this section largely draws upon the opening chapters of (Weiß 1999).[1]

2.1.1 Intelligent agents

There is an ongoing debate about what (intelligent) agents are (Foner 1993, Franklin and Graesser 1997). The minimal consensus is reflected by the often cited definition of Russell and Norvig (2003) which states that "an agent is anything that can be viewed as perceiving

[1] The literature on DAI abounds: early collections of DAI papers can be found in (Huhns 1987, Bond and Gasser 1988b, Demazeau, Müller and Muller 1990, Demazeau, Müller and Muller 1991, Demazeau and Werner 1992). Wooldridge and Jennings (1995b), O'Hare and Jennings (1996), Huhns and Singh (1998b), and Weiß (1999) provide more recent overviews of the current DAI research landscape. Well-known introductory articles are those of Bond and Gasser (1988a), Gasser and Huhns (1989), Wooldridge and Jennings (1995a), Wooldridge and Jennings (1995c), Nwana (1996), Moulin and Chaib-Draa (1996), Franklin and Graesser (1997), Huhns and Singh (1998a), and Jennings, Sycara and Wooldridge (1998b).

its environment through sensors and acting upon that environment through actuators". The central idea is that an agent persistently operates in an environment about which it has some partial information and which it can only partially control. Clearly, by this definition too many things (e.g. thermostats, automatic burglary alarms, word processors etc.) would qualify as agents. Using *situatedness* (or *embodiment*) in an environment and the *sensoric* and *effectoric capacities* as the sole criteria for agenthood is obviously not sufficient to express what agents are (let alone "intelligent" agents).

A much more powerful – though not less disputable – criterion is that of *autonomy*. As Wooldridge (1999, p. 27) writes, autonomous agents are agents who "decide for themselves", i.e. agents who "are able to act without the intervention of humans"(p. 29). Another view suggests that an agent is autonomous "in that its behaviour at least partially depends on its own experience" (Weiß 1999, p. 1). But how are we to judge whether these criteria are fulfilled by a piece of software?[2] A thermostat obviously acts without human intervention (after all, it is supposed to relieve humans from the task of adjusting a heating system), and the behaviour of a word processor obviously depends on the document the user has created (i.e. the keyboard input the program has experienced).

Faced with this problem, most authors resort to qualitative criteria for autonomy which are usually related to some notion of intelligence. "Intelligent" agency implies *flexible, robust, efficient* goal-directed behaviour in *complex, uncertain*, and *dynamic* environments. The properties that are usually considered desirable and important for such intelligent agents are (cf. (Wooldridge and Jennings 1995c)):

- *reactivity*: the ability to react to changes in the environment and to do so in a timely fashion;

- *pro-activeness*: the capability of goal-directed initiative – agents are expected to take action in order to fulfil their goals;

- *social ability*: agents are able to interact with other agents (and humans) to further their goals.

Although it is certainly true that we are only willing to ascribe real autonomy to artefacts if they are able to exhibit some kind of non-trivial, goal-driven behaviour, a crucial aspect of autonomy has been largely overlooked, which has to do with the *observation* of autonomous systems. By this perspective, which is very important for our frame-based methods, we would purport that

> A system S is *observer-autonomous* with respect to an external observer O, if S behaves differently under circumstances O considers identical for S.

This not only rules out the possibility of thermostats and burglary alarms being considered agents, but also of more complex artefacts such as word processors; even though they encapsulate complex functionalities, we expect them to react identically under comparable circumstances[3].

[2] The notions of agency and autonomy are also common in robotics research. However, in the remainder of this chapter, we will only refer to software agents.

[3] It is not our intention to comment on certain common word processors that actually seem to show very autonomous behaviour.

Note that this definition does not imply that autonomous systems are inherently non-deterministic. It merely states their such systems show a behaviour that varies between situations the observer considers equivalent for the system (situations that should "make no difference" to the system, in the observer's view). For example, if the thermostat behaved differently on two different days with the same temperature conditions because it takes forecasts into account which it receives through an online Internet connection, it would seem autonomous if we assumed that it only reacts to the current temperature measurement. Thus, what matters is whether the *causal coupling* between the inputs and the outputs of the system is non-trivial because the agent performs some additional internal processing that is hidden from the observer. Hence, in a certain sense, an agent is only autonomous to the degree that the observer is ignorant of its internal functioning.

2.1.2 Multiagent systems

Multiagent systems (MAS) are systems in which agents interact with other agents (and possibly humans) in a common environment. According to Jennings et al. (1998b), their most prominent characteristics are that

- agents have incomplete information and are restricted in their capabilities,
- data and control are decentralised, and
- computation is asynchronous.

As with the above definition of intelligent agents, this only represents a minimal list of mostly technical properties. In fact, MAS is a very general term for a variety of different types of systems used in different disciplines:

1. In a "strictly AI" view concerned with achieving intelligence in computational systems, MAS are collections of intelligent agents. Two lines of research can be distinguished within this field:
   (a) *Distributed problem-solving* (DPS) systems (Durfee 1999) decompose complex tasks and distribute them among strictly cooperative agents who communicate, plan and work together to achieve a shared goal.
   (b) *Multiagent societies*[4] allow for (potentially non-cooperative) forms of interaction other than those of DPS systems. In these systems, agents are typically self-interested and do not always share a common goal.
2. Research on *multiagent-based social simulation* (Sichman, Conte and Gilbert 1998, Moss and Davidsson 2000, Sichman, Bousquet and Davidsson 2003) utilises MAS for the computational study of social phenomena and to test hypotheses by means of simulation that have been previously formulated from a social science perspective.
3. Agents and MAS scenarios are being used in *user modelling* and *human-computer interaction* contexts where so-called *socially intelligent agents*[5] (Dautenhahn 2000) are

[4] This is not an established term. We use the term society here, because, as in human society, such MAS are characterised by the existence of *communication* rather than *cooperation*.

[5] Note that this usage of "social intelligence" is completely different from ours in that we do not require agents to fit into a *human* social context.

agents who exhibit anthropomorphic and believable behaviour which allows them to socially interact with humans.

4. *Agent-oriented software engineering* (AOSE) (Ciancarini and Wooldridge 2001, Lind 2001, Weiss 2001, Wooldridge, Weiss and Ciancarini 2002, Giunchiglia, Odell and Weiss 2003) seeks to apply agent technology to the area of software engineering, that is, to exploit the agent paradigm for the development of methods for building complex, distributed software systems.

In this thesis, we will almost exclusively deal with the "multiagent society" perspective, though we will occasionally refer to AOSE-related aspects. This means that, even though we borrow from sociological theories, we are neither interested in simulating social phenomena nor in whether our agents exhibit a behaviour that is human-like or understandable for humans. Our goal is to construct agents who can interact with each other effectively. Also, we do not assume a strict DPS position, because, as stated at the very beginning of the introductory chapter, we aim to contribute to research in *open* MAS.

2.1.3 Open systems

To highlight the aspects of MAS research that are relevant to ours, we should first clarify what we mean when we talk about *open* MAS. Initially, this term was introduced by Hewitt (1986), and further discussed in Hewitt's (1991) article on open information systems semantics and Gasser's (1991) response to this article. However, the properties associated with open systems today go beyond those originally laid out by Hewitt (asynchrony, local authority, late-arriving information, division and specialisation of labour and multiple authorities, arm's length relationships) as these are today assumed to hold in virtually *any* MAS.

More recently, Davidsson (2001) has presented a taxonomy of different types of artificial societies with more fine-grained distinctions. In his view, *fixed* and *closed* MAS are systems in which agents are all controlled by the same designer; additionally, in fixed systems, the set of agents in the system must be fixed at design time. *Open* societies allow agents with different owners to freely enter and leave the system. *Semi-open* systems are able to decide which agents may enter the society from outside, while *semi-closed* systems do not allow foreign agents to enter the system, but may allow external humans, organisations or agents to create an agent in the society (that is effectively controlled by the society "provider"). *Anarchic* systems, finally, are open systems in which not even a common communication language or a common set of roles for the participating agents has been agreed upon.

In fact, our own definition of open MAS comes quite close to that of anarchic systems. In our example Web linkage application (section 1.1), agents roam from site to site looking for interesting links and communicate with other agents to negotiate linkage structures that are profitable for them and for the visitors of their own site. Thus, the MAS must be necessarily open, and, although we will assume a common communication language with respect to *content* (i.e. the ontology of concepts and logical relationships used in messages are assumed to be common knowledge), there is no restriction on or *a priori* design of communicative conventions, interaction protocols or the like.

To make this point more precise, we summarise the main characteristics of the open MAS we deal with in the following list:

Openness of ownership. Agents may pertain to different human users, organisations or may be spawned by other artificial agents. Often, the owner cannot even be securely identified.

Openness of membership. Entry into and exit from the society is unpredictable. The agent population may be arbitrarily-sized and volatile. In general, there is no possibility to exclude agents from the society.

Openness of behaviour. Agents may exhibit arbitrary behaviour within the bounds of operations allowed by the environment. In particular, one agent cannot keep any other agent from performing an action or communicating with other agents.

Openness of conventions. Although a set of interaction protocols may be available, nothing can be said about whether agents will abide by certain rules of interaction, i.e. whether and how protocols will be used.

Openness of internal design. No assumptions can be made about the internal processing of an agent. This implies that the goals of the agent are unknown to the degree it wishes to conceal them, and that the agent may be untruthful, deceptive and malicious.

To summarise, open MAS are systems populated by varying number of agents of potentially unknown origin, with unknown motives and goals, whose behaviour is highly unpredictable and, in principle, unrestrained.

This definition of open systems is *dual* to our definition of (observer-)autonomy (p. 12) in that agents in such an open MAS appear to be fully autonomous for other agents. In section 2.2.3 we will see that this duality is very important for our theoretical model of communication.

2.1.4 Interaction and coordination

As Weiß (1999) writes, the problems of DAI research are "centered around the elementary question of *when and how to interact with whom*". Singh (1997) and Huhns (2000) even go as far as to plead for a new programming paradigm called *interaction-oriented programming*. With a flavour very similar to our depiction of future complex applications in open systems at the very beginning of the introductory chapter, they claim that viewing complex software systems as collections of loosely coupled, communicating components is useful *beyond* the realm of DAI.

But what is interaction, how does it relate to coordination, and what is the role of communication in all this? A famous definition of interaction and coordination (Malone and Crowston 1994) states that

> An interaction can be viewed as a formalisation of a concept of dependence between agents, no matter on whom or how they are dependent. Coordination is a special case of interaction in which agents are aware how they depend on other agents and attempt to adjust their actions appropriately.

It is important to note that interaction here does not imply real action ("inter-action" so to speak), i.e. some form of actual behaviour of an agent towards another agent that affects one of these agents. Likewise, coordination only denotes an adjustment of the decision-making processes of an agent that takes inter-agent dependencies into account. In fact, there are many situations in which coordination is achieved without (or prior to) direct action, e.g. by following fixed norms and rules that need not be communicated, by predicting the other's imminent actions from past experience, etc. Note also that coordination is a *neutral* concept that subsumes the special cases of cooperation, collaboration, competition and open conflict.

In this very general sense, all of MAS research is or course somehow concerned with interaction and coordination. However, a further refinement of these notions is necessary to focus on the research problems the present thesis deals with, which we might term *communicative* interaction and coordination.

Communicative interaction and coordination focuses on communication between artificial agents as the primary means of interaction. The central question of "when and how to interact with whom" can now be rephrased in more specific terms by asking (Hewitt 1991)

> What should be the communication mechanisms and conventions of civilised discourse for effective problem solving by a society of experts?

Formulating this question not only implies a shift in focus towards communication, it also marks a shift in the *level* of interaction one is dealing with, namely that the *micro* (agent cognition) and *meso* level (communicative "face-to-face" interaction, negotiation, joint planning and group formation) of multiagent systems are given more attention than the *macro* level (role and capability assignment, definition of global norms, resource distribution, distribution of profit, communication language definition, etc.).

2.1.5 Agent communication

Two research topics play a major role for this level of interaction: Agent communication languages (ACLs) and interaction protocols. As both are directly related to our research and will be more closely reviewed in section 2.3, we only provide some very general remarks at this point.

ACL research (see, e.g. Labrou et al. 1999, Kone et al. 2000, Dignum and Greaves 2000, Chaib-draa and Dignum 2002) focuses on the design, formalisation, implementation and verification of appropriate communication languages for agents. Languages such as as KQML (Finin, Labrou and Mayfield 1997, Labrou and Finin 1997) and FIPA-ACL (FIPA 1999a) describe what messages and message types are admissible in inter-agent communication, how their content is structured and what format, transmission protocols and physical communication channels agents have to use when exchanging messages. Furthermore, these languages have to be given a precise formal semantics if agents are to make sense of them.

Interaction protocols[6] (Burmeister, Haddadi and Sundermeyer 1995, Fallah-Seghrouchni, Haddad and Mazouzi 1999, Koning, Francois and Demazeau 1998, Kuwabara, Ishida and Osato 1995, McBurney, Parsons and Wooldridge 2002, Pitt and Mamdani 1999b, Quintero, Ucrós and Takhashi 1995, Schillo and Fischer 2001), on

[6] We only distinguish interaction protocols from ACLs here for conceptual reasons – many would consider protocol research as an integral part of ACL research.

the other hand, define what *sequences* of messages are admissible in particular agent interactions, and how the position of certain messages within a conversation affects their meaning.

The most famous example for such protocols is the *contract-net protocol* (Smith 1980, Smith and Davis 1981). This protocol describes how a manager who is to assign a task to a contractor after receiving bids (e.g. prices of executing the task) from different agents exchanges messages with those agents participating in the "contract-net".

From the *open systems* viewpoint that we have argued for above, a particularly interesting kind of interaction protocols is that of *negotiation* protocols (Raiffa 1982, Sandholm and Lesser 1995, Jennings et al. 2001, Koning et al. 1998, Panzarasa and Jennings 2001, Tamma, Wooldridge and Dickinson 2002). This is because, in the view of open MAS we have developed and which ascribes a very high level of autonomy to every member of the society, we have to espouse the "uncompromisingly individualistic" view Alonso (1998) alludes to, according to which "societies are defined as entities reducible to the conjunction of the commitments each agent has agreed on through negotiation" (ibid.).

Unlike other kinds of protocols, negotiation protocols are characterised by the distinct property that their execution is supposed to lead to coordinated action. This coordinated action is usually the result of a mutual agreement on the actions the interacting parties will pursue. In the above example of the contract-net, manager and contractor agree (i) that the contractor will carry out the task, (ii) that he will inform the manager upon completion (or, possibly, failure) of the task, and (iii) that the manager will pay the negotiated price in return. The essence of negotiation is that if it succeeds (i.e. if an agreement is reached *and* kept), the interacting parties will *commit* themselves to a *joint* course of future action.

The importance of negotiation as the sole possibility of achieving coordination among purely autonomous agents in open systems further narrows down the scope of our work – essentially, we are trying to build agents that can negotiate effectively. However, we take a path different from devising appropriate "infallible" protocols that are able to ensure fruitful cooperation, as attempted by the areas of automated negotiation (Rosenschein and Zlotkin 1994, Jennings et al. 2001) and mechanism design (Fischer, Ruß and Vierke 1998, Sandholm and Lesser 1995, Sandholm 1999).

Much of the motivation to do so stems from our study of social theory, and of the introduction of certain notions of interactionist sociology to the world of multiagent systems. The following section discusses the foundations of this interdisciplinary approach.

2.2 Interactionist Socionics

In this section, we attempt to provide an excursive introduction to interactionist sociology and to analyse how fundamental insights of this school of sociological thought carry over to the field of multiagent systems. Thereby, our central aim is to explain how a principled application of interactionist theories aids in understanding interaction in open systems and consequently in the construction of appropriate algorithms that agents can use to manage their interactions.

We will first describe, in brief, the emergent discipline of Socionics, which includes addressing the question "*why sociology?*", since the choice of sociology as a source for interdisciplinary work in conjunction with DAI – though not exactly far-fetched – deserves some justification and explanation.

The subsequent introduction to symbolic interactionism focuses particularly on the theories of George Herbert Mead and Erving Goffman, which are foundational for the methods we develop. Finally, we will formulate a list of assumptions for multiagent systems design that we have derived from interactionism.

2.2.1 Socionics

The term *Socionics*[7] was coined by Thomas Malsch (Malsch 1998, Malsch et al. 1998, Malsch 2001, Fischer and Florian 2003) for a new approach to combining sociology with DAI. After a similar initiative had been started by a group of American computer scientists and sociologists about ten years before the conception of this term (Bendifallah et al. 1988), the "new" attempt set out on a more prestigious, methodologically elaborate endeavour. According to Malsch (2001), three "references" provide the rules of the game for Socionics research:

1. The use of computer models in sociological research ("sociological reference"),
2. the development of new techniques and methods in DAI that make use of sociological theories ("computational reference"), and
3. the social impact of hybrid societies that consist of human and artificial agents ("praxis reference").

These references also define the research issues that constitute the quintessence of Socionics: *social simulation*, *sociologically informed MAS* and *hybrid MAS*.

The particular lure of such a scientific collaboration lies in the combination of two almost diametrically opposed disciplines such as sociology and computer science. This is because, in Malsch's view, in addition to a "simple" interdisciplinary exchange of methods, the two fields have the potential to challenge central assumptions the respective other discipline takes for granted.

We will explain below to which degree this is true of the research results presented here. As to the three references mentioned above, this thesis only deals with the second: *sociological theories are used to improve computational methods.* Thus, the alleged cross-fertilisation between the two disciplines is only visible in one direction in our work.

Why sociology?

Artificial intelligence has traditionally sought to exploit the theories, findings, and models of a variety of other disciplines[8], such as philosophy (logic, knowledge representation, automated reasoning), economics (decision and game theory), psychology (cognitive architectures, planning, machine learning), linguistics (natural language processing) and biology (neuroscience, genetic algorithms, ant algorithms).

However, none of these disciplines deals *principally* with *meaningful* communication among autonomous rational agents. While some communication-like phenomena are treated in all these fields (for instance, proof games in logic, observation of actions in game

[7] Following neologisms such as, e.g., *bionics*.

[8] Russell and Norvig (2003) discuss the relationships between AI and these areas at length in their introductory chapter.

theory, pheromone exchange in ant algorithms), none[9] of them assumes that agents with different motivations exchange a great variety of different, complex messages.

It is this aspect that makes sociology attractive for research in open multiagent systems: the concentration on high-level communication between individuals, and the conception of society as the sum of these communicative processes. More specifically, sociology assumes a stance characterised, *inter alia*, by the following aspects that make sense for a study of interaction in open systems:

Autonomy of individuals By the *Gedankenexperiment* of "double contingency" (Luhmann 1984), any two agents confront each other in full ignorance of what the other will do, since, in principle *any agent may do anything*. Moreover, "ego" cannot even know whether "alter" knows what "ego" will do! Although social expectations, norms and other rules of interaction may provide some guidance in predicting each other's actions, and even though agents may appear to strictly follow socially constructed "rituals" they are, *in principle*, free to act as they wish.

Communication and cognition People engage in communication to fulfil their needs and goals, and they have the ability to reflect on their actions and on the effects of these actions in a social context. Thus, "real" communication is not only an exchange of "signals" that trigger certain reactions, its usage is mediated by thought.

Meaning in retrospect As future (inter)actions cannot be fully predicted among humans, *meaning* can only be ascribed to actions retrospectively, after their effects have been observed. In other words, analysing the effects certain communicative actions have had in the past is the only clue we have when it comes to making sense of these actions.

The combination of these assumptions is not only in concordance with the assumptions made in section 2.1 regarding open societies of artificial agents. Moreover, it allows us to identify a focal point of research in open MAS, namely that of *communication systems*, i.e. systems of evolving communicative processes among rational agents.

As we have pointed out in (Nickles and Rovatsos 2004), this leads us to formulate a refined definition of Socionics. This definition is based on the insight that the core contribution of Socionics is to draw our attention to the empirical analysis of the communicative processes that take place in an open system under the assumption that these processes unfold among intelligent agents capable of goal-directed action and communication:

$$\textit{Socionics} = \textit{empirical communication analysis} + \textit{rational action}$$

Of course, there are further reasons for using sociology, such as

- the huge diversity of theoretical work in this discipline,
- the fact that humans are able to perform well in society, and that sociology has the ability to answer how this is achieved,

[9] Social psychology and linguistic pragmatics are probably those sub-areas that are most close to these assumptions. However, for our purposes, they are too much interested in either humans dealing with social relationships or in the functioning of human language.

- the fact that sociology is capable of dealing with the complexity of human society which is one of the most complex systems known.

However, these would speak for an application of sociology to any kind of MAS; the more specific arguments given above are particularly important for the open systems we deal with in this thesis.

2.2.2 Interactionist sociology

Symbolic interactionism (Blumer 1962) is one of the main schools of American sociological thought of the 20th century. While its roots lie in American *Pragmatism* (Nagl 1998), a philosophical movement that emerged in the late 19th century with Charles Sanders Peirce (1839–1914), William James (1842–1910), and John Dewey (1859–1952) as its most well-known representatives, George Herbert Mead's (1863–1931) *"Mind, Self and Society from the Standpoint of Social Behaviorism"* (1934) is commonly seen as the book that laid the foundations for symbolic interactionism in sociology.

Interactionism is mainly concerned with *interactions* between humans as the basic building blocks of society. Its outlook on the individual is that of an intelligent being that is capable of meaningful communication and that takes active part in social interaction by interpreting "symbols" employed by others and employing these symbols in interaction towards others in turn. Blumer (1986) summarises the core assumptions of symbolic interactionism:

> The first premise is that human beings act towards things on the basis of meanings that things have for them [...] The second premise is that the meaning of such things is derived from, or arises out of, the social interaction that one has with one's fellows. The third premise is that these meanings are handled in, and modified through, an interpretative process used by the person in dealing with the things he encounters.

These assumptions are made on the grounds that the use of language distinguishes man from animal, and endows them with the ability to reflect upon their reactions to a perceived situation. Mead, who was initially heavily influenced by behaviourism (at that time, the leading school in psychology), identified the capacity of humans to share the perception of their own actions with others (initially, by hearing their own voice) as the reason for this ability of reflexive thought. By understanding that different communicative symbols have different effects on the listener, man is capable of mediating the "reflex arc" between perceived stimulus and exhibited response with reflection, i.e. of manipulating his own reaction in accordance with his needs. This conclusion led Mead to abandon the strict behaviourist position in favour of a theory of *social behaviourism* that included a notion of human *mind* and how it is constructed through social interaction.

These views make clear why the resulting theoretical positions are often also referred to as the *interpretative* approach to social action, according to which (Paetow and Rovatsos 2003):

- Action is inherently creative (Joas 1997), but must also always be seen as a process of adjustment to the requirements of the current situation.

- Apart from the physical world, social actors live in a symbolically constructed world, a "symbolic universe" (Strauss 1993).

- Individuals act in a meaningful way, i.e. by producing and interpreting symbols.
- Society is conceived as an inherently dynamic process, and not as a fixed structure.
- Neither individuals nor society as a whole are at the focus of attention, but interaction.

One of the central questions that arises from these assumptions is: *How can collective social action be explained on the grounds of the interpretative processes the participating individuals engage in?* Obviously, it does not suffice for such “joint action” (Blumer 1986) to be a mere combination of individual actions – these actions must be “aligned” with each other to make sense as collective action:

> [A] joint action cannot be resolved into a common or same type of behavior on the part of the participants. Each [. . .] necessarily occupies a different position, acts from that position, and engages in a separate and distinctive act. It is the fitting together of those acts and not their commonality that constitutes joint action. [. . .] Their alignment does not occur through sheer mechanical juggling. [. . .] [T]he participants [must] fit their acts together (Blumer (1969, p. 70), quoted from Strauss (1993, p. 40)).

This is where Pragmatist philosophy comes into play: to achieve such alignment and to establish reasonable social procedures, interactionism assumes that the process of scientific discovery among a “community of investigators” called *social inquiry* (Dewey) carries over as a general scheme for achieving consensus in human society. This process consists of the following steps:

1. A felt difficulty, i.e. the shared perception of a problem.
2. Suggestion of a solution.
3. Development by reasoning of the consequences (examining the usefulness of a suggestion by projecting its potential effects).
4. Conclusion of belief or disbelief (that is, the suggestion is adopted or discarded).
5. (Possibly) further experiments.

What Pragmatist philosophers proposed as a generic model for establishing truth[10] is taken as a general model for the evolution of social structures and processes by interactionist sociology. Actors engage in communication and in the interpretation of communication until actions fit together in a “negotiated order” (Strauss 1978a, Maines 1982) which is only an interim state until the next negotiation phase. In a much more general interpretation than that of economics (where negotiation is usually associated with bargaining over goods and services), “negotiations” are viewed as a constant “struggle over signs” in

[10] In fact, Pragmatism denies any metaphysical or universal truth. Instead, it suggests that the ”truth” of a concept or theory is the sum of its consequences and effects in everyday life, and that, eventually, it will be established by its own success. A *pluralist* and *fallibilist* stance is assumed, which allows for a multitude of (even seemingly conflicting) theories to co-exist at any point in time. Their existence as “working hypotheses” is justified, even if they will be transformed, reconsidered or rejected in the future.

interactionism, in which each agent seeks to establish his own symbols and meanings in ongoing communication.

Conceptually, this seems to be an intuitively appealing and reasonably simple outlook on social discourse. In order to understand how all this maps to the cognitive processes of interacting individuals, we have to look into theories of how social action is organised in the human mind. Mead and Goffman provide two such theories.

Mead: Mind, Self and Society

The central concept in Mead's theory of social action is that of the *self*. Having a self means being able to interact with oneself, and the self is the central mechanism that is used in forming and guiding the conduct of a human actors (Blumer 1966). Blumer (1962, p. 181) summarises the concept of self as follows:

> In declaring that the human being has a self, Mead has in mind chiefly that the human being can be the object of his own actions. He can act toward himself as he might act toward others.

The rationale behind this claim put forward by Mead is that humans use *significant symbols*, which are, in the most general sense, actions that "stand for" other actions[11]. In order to anticipate the effects of uttering such a symbol, the individual must indicate to himself what the symbol means:

> What is essential to communication is that the symbol should arouse in one's self what it arouses in the other individual. (Mead 1934, p. 149)

Directing symbols to oneself in an introspective fashion requires (and precipitates, according to Mead) that one has a model of oneself, and this delivers the primary justification of the existence of a self. The self, however is not conceived of as a fixed structure, but as an ongoing process. In Mead's view, it is a process resulting from an (actor-internal) interaction that can be analysed in terms of a *triadic* constellation of three elements: "*I*", "*me*" and *generalised other*.

The "*I*" is the centre of agent motivation that spawns the impulse to perform an action. This impulse might be ignited by the physiological needs of the individual, by an ongoing interaction, or by a pattern of cooperative behaviour (Turner 1988) that has been activated. The "I" acts spontaneously (without any further reflection) and does not manifest itself at any moments in time other than those in which it induces such an action impulse.

Hence, a second component called "*me*" is needed to observe the behaviour of the "I" in retrospect from a certain distance and to contemplate on experienced motivations that led to past actions. It serves different purposes (Paetow and Rovatsos 2003):

- In its capacity of mirroring the "I" as a self-conception, it can be used to reflect upon the "I". The "I" constantly "converses" with the "me" to identify how the current impulse can be reconciled with past conduct.
- It contains the experience of past actions of the "I". In fact, this implies that there are several "me"s for the different situations in which the individual has experienced different states of the "I" in the past.

[11] Under consequentialist Pragmatist assumptions, "meaning of behaviours" subsumes "meaning of objects", as consensual conceptions of objects are only defined in terms of *how these objects are acted towards*.

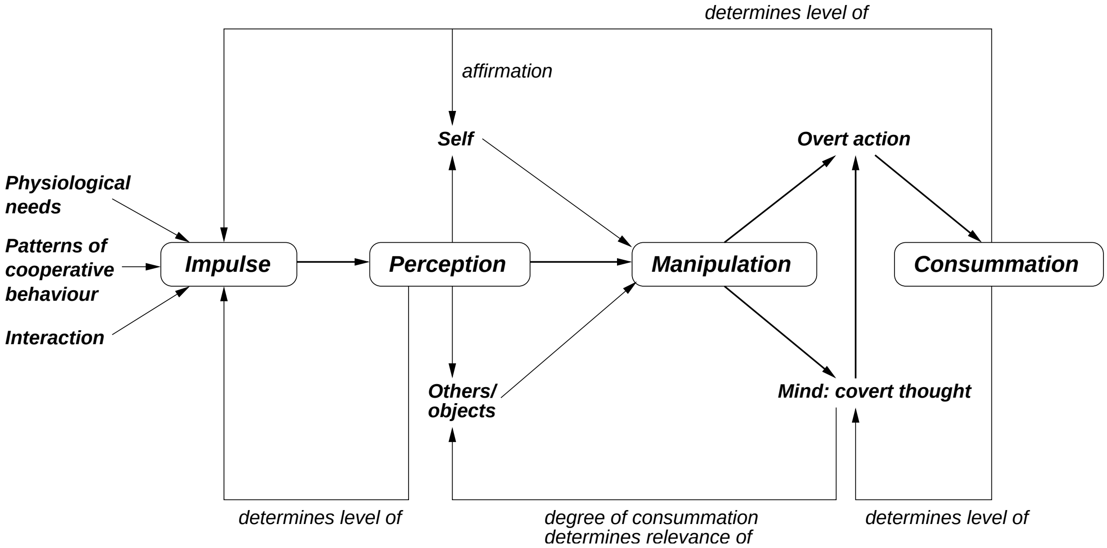


Fig. 2.1: Mead's four-stage model of the social act, adapted from Turner (1988).

- It incorporates own aspirations, needs and motivations as well as social expectations, norms, beliefs, values and conventions and is thus able to explain how these two sides were combined to yield the actions performed in the past.

To bring in the attitudes of a social context the individual is situated in, however, a third component is necessary: that of the *generalised other*. This hypothetical, abstract other belongs to the same social group or is otherwise characterised by sharing a common set of social expectations, and plays a central role in the construction of the self. Only through the generalised other can social control be exerted on the individual, can actions be censored, can their sanctioning be anticipated, and can general values influence the individual. Thereby, *role-taking* (Miller 1981), i.e. predicting how another person would (overtly) react to one's intended actions by (covertly) examining their effects on oneself, is of paramount importance. As figure 2.1 illustrates, "I", "me", and generalised other interact in the four stages of the social act until action is produced:

1. *Impulse:* The initiation of an act, caused by "[a] lack of adjustment and adaptation to one's surroundings" (Turner 1988, p. 32). This initiation is spontaneous and not reflected upon; it occurs out of the needs of the agent or automatic reactions that are generated in habitualised interaction.

2. *Perception:* Stimuli are perceived in the form of information about the objects and the others that are relevant to the elimination of an impulse. Perception makes the situation accessible for cognitive processing, and it also allows the individual to observe his own behaviour.

3. *Manipulation:* Through both overt action and covert thought (or "imaginative rehearsal", as Turner (1988, p. 33) has it), the actor attempts to manipulate the current flow of (inter)action which is blocked by the disequilibrating impulse. This may entail several cycles of adjustment with increasing cognitive complexity, if the blockage is hard to overcome. Of course, manipulation has to take the actor's self-perception and other perceived stimuli into account to achieve an appropriate adjustment.

4. *Consummation:* The inhibition of action is overcome by identifying an appropriate behaviour that achieves elimination of the disequilibrium. The expected reaction no longer differs from the actual reaction of the other(s), a coordinated joint action has been achieved, alignment is attained. Also, the "me" self-image(s) and generalised other(s) are enriched with information from the newly experienced act. The degree of consummation determines retrospectively (i) the relevance of objects and others in the act, (ii) the level of conscious thought that is still necessary, (iii) the degree to which the actor's self-image is affirmed or modified through the new act and, of course, (iv) the degree to which the initial impulse has been eliminated.

In *habitual* or *routine* interaction, the stage of consummation is reached without any substantial manipulation, while *conflict* requires "intelligent social reconstruction" (Campbell 1981), i.e. complex manipulation processes to achieve fruitful coordination.[12]

Mead's model of social action is very powerful and very useful for understanding the principles of social cognition from the standpoint of symbolic interactionism. However, it fails to make the relationships between actions in more complex forms of interaction explicit, since it always remains at the action-to-action level of analysis. This gap can be bridged by looking, for example, at Erving Goffman's concepts of frames and framing.

Goffman: Frames and Framing

Erving Goffman (1922–1982), an American sociologist who primarily engaged in analyses of everyday interaction situations (carried out in extensive field studies), developed the concept of *frames* as a metaphor for what defines a social context. In Goffman's (1974, p. 8) own words,

> when individuals attend to any current situation, they face the question *"What is it that's going on here?"* Whether asked explicitly, as in times of confusion or doubt, or tacitly, during occasions of usual certitude, the question is put and the answer to it is presumed by the way the individuals then proceed to get on with the affairs at hand.

Thus, frames can be used as schemes of interpretation for the current interaction situation that allow the individual to act "appropriately", i.e. in a competent, routine fashion. They are applied as "patterns of meaning" that reflect complex expectation structures and contain representations of interaction knowledge derived from experience that feed into and inform future interaction processes (Paetow and Rovatsos 2003).

Framing, on the other hand, is the individual or collective process of activating and applying frames. It occurs whenever the actors participating in an encounter interpret the situation according to the available frame "arsenal", and exploit the knowledge contained therein to guide and assess their own behaviour and that of others. Of course *misframing* is a common phenomenon, and is seen as a primary source of interaction problems. Although some divergences in the frames activated by interactants may go unnoticed[13], others may lead to an inhibition of the flow of interaction or even to open conflict.

[12] See (Paetow and Rovatsos 2003) for a detailed discussion of the interplay between routine and conflict interaction.

[13] Just think of a person who is bluntly criticising another, whereupon the person criticised laughs out loud (wrongly) thinking it was meant to be a joke.

In his *opus magnum*, Goffman (1974) describes and analyses a great variety of different kinds of frames. Among more standard kinds of interaction, he discusses such phenomena as "stunts", "muffings", "daydreaming", "contests", "ceremonials", "practicings", "demonstrations", "role-playing" (in psychotherapy), and "experiments", that are usually considered rather marginal or even extravagant forms of interaction. These examples, however, serve as illustrations for the vulnerability of frames, i.e. to express how much framing can be affected by manipulations (so-called *keyings*, *fabrications*, *designs*, *re-groundings*, etc.) they become subject to.

The anthropological stance that is assumed by Goffman adds an additional, very distinctive flavour to his outlook on interaction. Here, in keeping with the interactionist tradition, the individual is a *homo significans*[14] whose understanding of the social world is guided by the meanings of significant symbols. Moreover, Goffman emphasises the *dramaturgical* aspect of human behaviour (Goffman 1959, Hitzler 1992). In his opinion, people "play" their role in "interaction rituals" strategically in order to cope with the problems they have to confront in everyday interaction.

For our purposes, Goffman's concepts of frames and framing are important as they supplement the Meadian notions of self and social action and vice versa: Mead's model of the different stages of social action and how they evolve through the interaction between "I", "me" and generalised other provides a very concrete *process* model for social reasoning. In principle, it can readily be operationalised for computational purposes step by step. Frames, on the other hand, are the *data structures*, so to speak, that can be used to represent different types of interactions. They allow for a combination of individual actions to form complex *trajectories*, and, unlike the restricted micro-models of individual actions in the Meadian view, frames can be used to include context information in order to obtain descriptions of complex interaction settings. Also, frames allow for *generalisation* over different interaction situations; they can be used to abstract from the details of concrete actions and reactions to extract more widely applicable patterns of interaction.

In the following section we depart from the realm of sociology and discuss the implications of using interactionist theories (and more specifically, the concepts of Mead and Goffman just laid out) for building agents that can successfully operate in open MAS.

2.2.3 Implications for our research

Quite a number of researchers have advocated the use of symbolic interactionism in multi-agent systems in the past. Before introducing our own list of assumptions and the research agenda that results from them, we shall briefly review existing work in DAI that utilises interactionist principles so that we can make explicit reference to them where necessary (in particular, to distinguish our own approach from existing work).

[14] We use the typology proposed by Reckwitz (2000), who distinguishes between this culture-theoretic notion of *homo significans* and two other types of explanation frameworks for social action: the *homo oeconomicus* model, according to which actions are chosen because they appear to be most profitable for the individual, and the *homo sociologicus*, by which social behaviour is explained on the grounds of sanctioned expectations or internalised norms.

Interactionism and DAI

Kornfeld and Hewitt (1981) were the first ones to make use of the *scientific community metaphor* in the construction of DAI systems that bears strong resemblance to the "community of investigators" idea in Pragmatism. Their system consisted of experts who interacted to find the best solution to a given problem by making suggestions, criticising them, etc.

Another early advocator of interactionism in the DAI community is Les Gasser. The MACE system (Gasser, Braganza and Hermann 1987a, Gasser, Braganza and Hermann 1987b) included both notions of (i) "social worlds" (Strauss 1978b) operationalised as knowledge-level agent boundaries (Carley and Gasser 1999), and (ii) explicit models of other agents called "acquaintances" that allowed agents to model the skills, roles, and knowledge of other agents. Gasser has also argued elsewhere that this is a crucial step to achieve "taking the role of the other" (Gasser 1991, p. 133). A more recent advocator of role-taking, but also of other Meadian concepts was Craig (1994), who elaborated a full-fledged model of the *self* to be used in MAS. Unfortunately, to our knowledge, there exists no account of systems that were actually built using this model.

Strübing (1998a, 1998b) investigates the potential of symbolic interactionism for MAS research and provides an in-depth analysis. Among other issues, he ponders on the possibility of "real" symbolic interaction among machines, discusses (and confirms) the suitability of interactionism for open systems, the adequacy of the "negotiated order approach", etc. Burke (1995) makes the case for a "social AI" as a new approach to general AI problem-solving and draws largely upon interactionist theories. He conjectures that reflective interaction may be the way to solve many problems that remain unsolved in present AI.

To our knowledge, Goffmanian theories have only been used by (Chicoisne and Pesty 1999) in MAS to date. However, as these authors only use Goffman's "theory of faces", their work is not directly related to our discussion of frames and framing.

A list of assumptions

We will now present the assumptions that result from the discussions of open MAS, Socionics and interactionist sociology in sections 2.1.3, 2.2.1 and 2.2.2. Although many of them may appear to follow trivially from the above considerations, the formulation of a comprehensive list of the principles underlying our research is not only a matter of methodological clarity; it also serves as a starting point for the research agenda pursued in this thesis.

Assumption 2.1: Agents and their environment
Agents are situated in an uncertain, dynamically changing environment. They obtain information about the state of the environment through perception and they can manipulate it through physical action. Perception and action provide only *partial/incomplete* information about and control of the environment. Perception may be *incorrect*, action execution may *fail*. Interaction with the environment is *persistent* i.e. we assume that agents will continue to operate in their environment for a very long time in the future.

Assumption 2.2: Agent deliberation and rational action
Agents have preferences regarding different states of the environment, and they strive to

achieve those states that are most desirable to them. To this end, they *deliberate*, i.e. they take action to achieve their goals, and their decisions are *rational* in this sense (they are not directly influenced by other external factors). They may have different (conflicting) goals at a time, and they may pursue several goals in parallel. They revise the status of goal achievement upon incoming new information.

Assumption 2.3: Causal models of the world
As the world changes and is not entirely predictable, it is useful for agents to organise their knowledge of the world in some kind of *causal model* in order to achieve effective goal-directed action. Such cause-and-effect models may be encoded in logical rules, statistical distributions or any other appropriate representation, and they may be hard-coded offline or learned from observation. In a dynamically changing environment, the capability to *learn* such models is certainly an advantage.

Assumption 2.4: Interaction with other agents
The environment is co-inhabited by a population of other agents who may enter or leave it at any point in time. The agents need not be designed identically, and their internal design is *opaque* to other agents – all agents perceive of other agents are their *overt* actions. However, agents' actions have effects on each other's goal attainment, agents are *inter-dependent*.

Assumption 2.5: Communication
Agents may *communicate* with each other through a direct exchange of messages. *Messages* are different from other (physical) actions executed to manipulate the environment in three ways: firstly, the *autonomy* of the recipient of the message stands in contrast to the (rather mechanical) rules that govern physical environments. Agents who receive messages are free to fulfil or disappoint the *expectations* associated with them. Secondly, communication *postpones* "real" physical action[15]: it allows for the establishment of causal relationships between symbols and subsequent symbols or physical actions, so that symbols may "stand for" other symbols or actions (and have no other significance in physical terms). Thirdly, we can assume that agents have an infinite repertoire of different messages at their disposal, as these can be distinguished by different symbolic content. At any point in time, they can come up with new symbols.

Assumption 2.6: Communicative expectations
To predict system behaviour, it is not only useful to build causal models of the physical environment, but also of other agents. Whenever such models refer to messages or publicly perceived physical actions as causes or effects of the behaviour of other agents and the environment, we refer to them as *(communicative) expectations.*

Assumption 2.7: Generalisation of expectations
To manage cognitive complexity, the cognitive representations of expectations must generalise from individual observations of interactions by grouping together similar experiences in descriptions of categories of similar interactions. In particular, on the grounds of a *homogeneity* assumption among all agents (which assumes that they are all rational,

[15] Of course, messages *are* physical actions in real terms. Usually, though, the exchange of messages does not have a strong, immediate impact on the physical environment with respect to goal achievement of the agents involved.

knowledgeable and capable of meaningful symbolic communication), descriptions of expectations abstract away the individual actor and replace him by a whole set of agents.

Assumption 2.8: Individualistic, strategic view of communication
From an "agent rationality" standpoint, the foremost function of communication lies in providing such a causal model for the behaviour of other *agents* that an agent can use in a similar way as rules that he discovers regarding the physical environment. Information about patterns of interaction and knowledge about the rules that govern the communicative behaviour of agents can then be used *strategically* by the agent to achieve his private goals.

Assumption 2.9: Empirical, constructivist and consequentialist semantics
In accordance with interactionist theory, the *meaning* of symbols used in communication is defined through their effects, i.e. we have to adopt a *consequentialist* outlook on communication semantics. We can distinguish between *first-order* effects (the immediate reaction of others to an utterance or action) and *second-order* effects, i.e. the way the expectation structures themselves are modified by the agent's current action. Also, since meaning is reflected by the expectations formed by an agent using information about communication observed in the past, it is always subject to the way expectation structures are construed by the agent. In particular, expectation structures do not only depend on his observations, but also on his own goals and motives. This leads to an *empirical* and *constructivist* view of meaning.

Assumption 2.10: Autonomy vs. predictability.
To perform well, agents should attempt to reduce the uncertainty associated with communicative expectations in the long run. At the same time, the agent himself seeks to maximise his own *autonomy*, he wants to be free to take any decision at any time to achieve his own goals. As expectations are generalised, this means that the agent's autonomy is a threat to his *own* predictability for others. A dilemma arises: ideally, an agent's peers would react to a message in a mechanised, fully predictable way so that any contingency about their behaviour can be ruled out.[16] The essential question thus becomes *"How can others be made to respond in a predictable way while the agent himself might change his mind and break existing expectations?"*

Taken together, these assumptions enable us to formulate a research agenda for the remaining chapters that explains which methods we use and what their scope is:

- We restrict our analysis to the micro-level of "face-to-face" agent interactions, in which agents respond to each other's utterances in a timely fashion. For agents to be able to correctly perceive and process an interactive encounter, this implies that interactions only take place among a limited number of agents, and that they have a fairly limited duration.

[16] This is a strongly simplified view. Weick (1979), for example, argues fervently that unpredictable behaviour of individuals largely *contributes* to the viability of organisations. Sociological systems theory (Luhmann 1995) suggests that deviation of behaviour from existing expectations is the prime source of innovation in society. We will assume that the agents under analysis in this work do not have the farsightedness to trade current contingencies in the other's behaviour for potential future innovation this other might provide through his deviance.

With respect to assumptions 2.4, 2.5, 2.6 and 2.7, this means that expectations must be formed out of the message sequences perceived in such encounters so that they can be strategically employed in future communication (assumption 2.8).

- The formation and generalisation of expectations (assumptions 2.6 and 2.7) is accomplished through a computational model of frames that represent communication patterns. Strategic construction and usage of frames (assumptions 2.8, 2.10) will be realised by a computational operationalisation of framing.

  Since Goffman does not provide an explicit process model for framing, Meads model of impulse-perception-manipulation-consummation will be used in the framing process to the end of balancing private goals with social expectations as represented by existing frames (see assumptions 2.2 and 2.10). As the use of communication symbols affects future semantics, meta-strategies such as "using symbols consistently" must be used that help ensure the emergence of a suitable "grammar of interaction" (assumption 2.9).

This links the material discussed in the previous sections to the contributions summarised in section 1.2 and explains how interaction frames and framing fit into our overall goal of developing novel methods for effective interaction management in open MAS. It also completes our account of background knowledge from the areas of multiagent systems and interactionist Socionics that was necessary to understand the work presented in the remaining chapters. Next, we shall take a closer look at related research themes.

2.3 Related Research Themes

As the contributions listed in section 1.2 suggest, the relationships between our research and other areas are manifold. Although we will refer to individual research results of others that are relevant to ours where appropriate in later chapters, we would like to mention the most prominent research *themes* in the following pages to set the scene for the methods and results presented in the remainder of this thesis.

To guide this discussion, we will use the illustration in figure 2.2. It groups different themes round a schematic view of the aspects we are concerned with. Each research theme is cast in the context of interaction frames and framing by relating it to different parts of the system view assumed for our study of these notions. We will now discuss these relationships in detail.

2.3.1 Agent architectures, rationality and learning

Interaction frames and framing are employed to achieve *intelligent social reasoning* and *learning*. This implies devising an agent architecture that is endowed with these capabilities. If we view frames as knowledge-level structures that contain information about past communication processes, framing can be seen as the activity of using observed patterns and regularities to achieve one's goals. This view connects our work to different themes of research in agents and agent architectures.

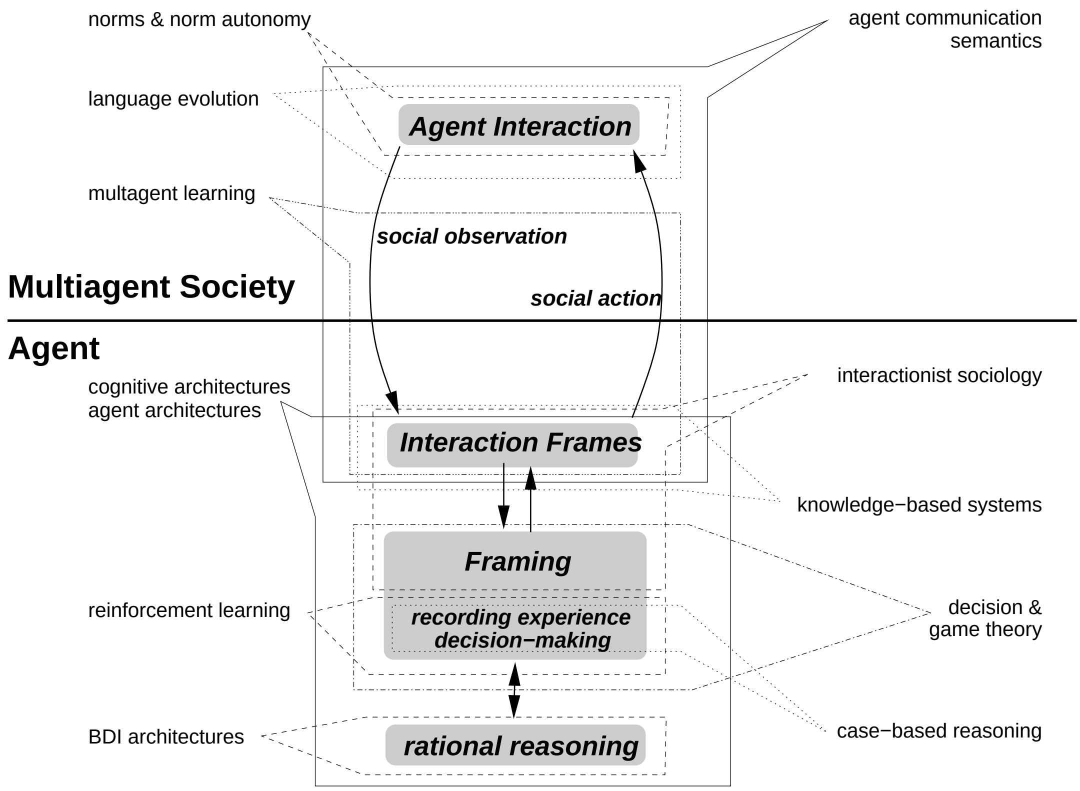


Fig. 2.2: Relationship between society-level (top) and agent-level (bottom) aspects of interaction frames and framing (middle) to various research themes (left and right).

Multiagent learning

The relationship to multiagent learning (Sen and Weiß 1999) is the most obvious: In our frame-based approach, agents collect information about how they and their peers have interacted in the past, and apply this information to achieve a better standing. As is shown in figure 2.2, this means that social observation is organised into interaction frames , which are then used for social action.

Among the different approaches to multiagent learning, the frame-based approach resembles most work done in the area of "opponent modelling". This area concentrates on the study of how agents can learn their opponents' preferences (Bui, Kieronska and Venkatesh 1996), their strategies (Carmel and Markovitch 1996, Freund et al. 1995, Vidal and Durfee 1997), the outcomes of joint actions, or all of these (Rovatsos and Lind 2000). Most of the time, these models are also used to derive an optimal strategy towards the opponent.

The majority of these multiagent learning approaches adopt a heavily *cognition-biased* view of learning, which aims at extracting as much information from observation as possible about an *individual*. However, in large-scale, open MAS, in which agents have only occasional encounters with peers they are acquainted with, learning models of individual peer agents may not be feasible. This is because the cost of acquiring and maintaining an adequate model of the other normally outweighs its potential benefits if the probability of interacting with that same agent in the future is not very high.

This problem has been one of the main motivations to develop methods that are concerned with a more *social* view of opponent modelling. Rather than focusing on the specific properties of particular agents, frame-based learning is concerned with learning the behaviour of agents in certain *classes of interactions*. It is this aspect that makes our approach distinct from other work in multiagent learning.

Balancing individual with social rationality

The fact that the learning data used by agents to extract regularities in interaction stems from non-local processes while decision-making is a local activity raises the question of how individual rationality can be balanced with social coherence. In fact, this is one of the main and most profound questions of MAS research: how can globally coherent behaviour be achieved among egotistical agents, who are only concerned with the pursuit of their own goals?

Trying to solve this problem by devising a new agent architecture brings us to the realm of *layered agent architectures* such as InteRRaP (Müller 1997, Jung and Fischer 1998) and Touring Machines (Ferguson 1992, Ferguson 1995), These architectures combine components for reactive behaviour and individual means-ends reasoning with components that cater for social coordination (joint planning, interaction protocols, etc.). Architectures specifically designed for *layered learning* (Stone and Veloso 1996, Stone 2000, Rovatsos and Lind 2000) also bear some relation to the architecture proposed here, although we do not consider learning at any level other than that of social interaction.

In the InFFrA architecture (cf. chapter 3) developed in this thesis, the existence of a sub-social rational reasoning component is assumed that is loosely coupled to the social reasoning layer. In the implementation of the InFFrA-based LIESON system (see section 6.1), we have used a Belief-Desire-Intention (BDI) reasoner for this sub-social reasoning level, but this choice is not mandatory.

The BDI model (Bratman et al. 1988, Georgeff and Lansky 1987, Rao and Georgeff 1992) is probably the model of rational agency that is most widely accepted in the MAS community today. It originates in the theory of human practical reasoning proposed by Michael Bratman (1987). According to this model, agents maintain *beliefs* about the state of the world and have *desires* regarding which world states to achieve. Once they commit themselves to fulfilling certain desires, they generate *intentions* to work towards these desires (Cohen and Levesque 1990a). Additional constraints ensure that BDI agents only deliberate in a reasonable way, for example by halting the execution of plans that have failed, by abandoning goals that cannot be achieved anymore, and by ignoring intentions that would lead to world states already achieved.

InFFrA extends the generic BDI model by showing how interaction patterns in the form of interaction frames can be integrated in it. Basically, this involves employing interaction frames as "social plans" in agents' local planning activities. The need for integrating social capabilities with BDI reasoning has been long identified (originally in the works on joint intentions by Cohen and Levesque (1991), but also in more recent contributions (Panzarasa, Norman and Jennings 1999, Panzarasa, Jennings and Norman 2002, Dignum and van Linder 2002, Dignum et al. 2000)). However, to our knowledge, the work of Dastani and colleagues (Dastani, van der Ham and Dignum 2002) is the only approach to date in which agents engage in interaction procedures depending on how the effects of these procedures affect their goals.

Cognitive architectures & knowledge representation

Cognitive architectures like ACT-R (Anderson and Lebiere 1998), SOAR (Laird, Newell and Rosenbloom 1987, Newell 1990), and PRODIGY (Carbonell, Knoblock and Minton 1989) are concerned with "the creation and understanding of synthetic agents that support the same capabilities as humans" (Langley and Laird 2002). Thus, they are at the core of the AI endeavour, especially because, in contrast to other, specialised methods, they aim at covering a broad range of these capabilities.

As mentioned in the remark regarding our "mixture of methods" (section 1.2, p. 6), our work certainly is in this same spirit, albeit we do not claim that the notion of interaction frames is suitable for any other general-purpose cognitive functions than those pertaining to social cognition. Also, we are not interested in "psychological adequacy", i.e. in simulating human cognitive functions.

For its use of case-based reasoning (Kolodner 1993, Aamodt and Plaza 1994, Watson and Marir 1994) methods in planning, the PRODIGY (Carbonell et al. 1989) architecture is probably the cognitive architecture that is most similar to ours. In PRODIGY, new planning operators can be derived from observation, and they are also assessed with respect to their usefulness. The same holds for InFFrA , except that instead of "operators" in the planning sense, "interaction practices" are observed and employed to achieve one's goals. However, planning in the sense of an intelligent combination of of agents' actions as it is common in standard AI (cf. chapter 11 in (Russell and Norvig 2003)) is not the focus of our approach. The reason for this is that communicative action, as it hardly depends on environmental conditions (other than the existence of a sender-receiver situation) allows for huge degrees of freedom, since what is "said" has little direct influence on what will be "done". Therefore, we are less concerned with the problem of finding good joint plans and agreeing on them[17] and more with recognising the intentions of other agents given their utterances.

This also explains the use of interactionist theories which are necessary to understand the "pragmatic" meaning of messages that enables us to instrumentalise communicative processes. Since the relationship to sociology has been discussed in depth in section 2.2.2, we omit further details at this point.

Finally, the use of the term "frame" requires some clarification when speaking of cognitive architectures, since it has a long-lived tradition in classical AI and cognitive science. The notion of frames proposed by Minsky (1975), which was very influential for the field of *knowledge representation*, is a much stronger concept than our version of "interaction frame". In his understanding, a frame is "a representation of an object or category, with attributes and relations to other objects or categories" (Russell and Norvig 2003, p. 366). This is only true of interaction frames in a very restricted sense, namely that they do represent such objects, yet that these objects exclusively contain information about "scripts" (Schank and Abelson 1977) of interaction processes. In other words, we do not claim that interaction frames are a concept for general-purpose knowledge representation like Minsky's frames.

[17] Work on distributed and multiagent planning abounds. Durfee (1999) provides a recent overview. Other common approaches are those of Decker (1987), Durfee, Lesser and Corkill (1992), Yokoo et al. (1992), and Durfee and Lesser (1991).

Decision theory and game theory

Quite naturally, the attempt to derive computational models of optimal behaviour and of how it can be achieved in uncertain, complex domains brings us to the realm of decision- and game-theoretic approaches, which is the last of the research themes we would like to mention in the "social reasoning" context. Decision theory and game theory (Fudenberg and Tirole 1991) address problems of rational decision-making in the context of available (often imperfect) information, given (i) action options, (ii) their potential (often uncertain) effects, and (iii) actors' preferences regarding these effects. The success of decision and game theory in MAS research (see (Fischer et al. 1998, Sandholm 1999) for overviews) is mostly due to the fact that they allow for a modelling of interaction situations with great mathematical rigour. Also, they provide simple, verifiable and very general principles of rationality and optimality along whose lines very generic models of decision-theoretic agents can be built, for example (Russell and Wefald 1991, Russell and Subramanian 1995).

We apply methods that originate in this area to model interaction situations as multiagent games, in which agents attempt to learn optimal strategies from experience. For this, they consider their own past actions and those of their peers, and, of course, rewards obtained in the past. While this is the standard setting for *multiagent reinforcement learning* (MARL), as studied, among others, by Claus and Boutilier (1998), Boutilier (1999), Hu and Wellman (1998), Littman (1994), Tan (1993), Crites and Barto (1996), and Weiß (1995), we have identified a need for modifying the classical MARL model for our purposes. This need springs from our concentration on communicative processes (rather than models of general action in state-based domains) which induce certain special properties on the decision models that have to be used to adequately capture rationality in agent architectures based on interaction frames. Our efforts to adapt the MARL model to suit our purposes have resulted in the aforementioned notions of *social abstraction* and *transient social optimality* (cf. page 7). These notions imply, in a sense, a revision of certain principles of rationality that are widely accepted in the fields of AI and machine learning.

2.3.2 Norms & communication semantics

Apart from the topics discussed above, which are mostly related to agent-level research problems, there is a number of themes the connection between which and our work only becomes obvious when adopting a "bird's eye", global view of frame-based systems.

Two categories can be distinguished among these: the more general issue of *norms* and their evolution and, with a focus on a more specific kind of norms, that of *communication semantics.* A separate section is devoted to *interaction protocols*, which can be seen as system-wide norms at a more practical level, because they are more closely related to interaction frames.

Norms, norm autonomy and norm evolution

The relationship between interaction frames and work on social norms in MAS (Conte and Castelfranchi 1996, Castelfranchi 2000, Conte and Dellarocas 2001) springs from the fact that interaction frames are used as normative knowledge to guide future social behaviour, and that they are the result of observing recurring patterns of interaction. However, the term *expectations* (Luhmann 1995, Brauer et al. 2001, Lorentzen and Nickles 2001, Nickles and Rovatsos 2004) is actually more suitable than that of norms, because interaction frames allow to capture the whole spectrum from cognitive to normative expectations.[18]

Among the literature on norms, we can further identify a more specific sub-theme, which is that of *norm autonomy* (Verhagen 2000), i.e. the autonomous adoption and rejection of certain norms from the standpoint of rational agents.

Rather than investigating the global effects of norms (a path often pursued in the area of multiagent-based social simulation (e.g. Saam and Harrer 1999, Conte and Castelfranchi 1996, Castelfranchi 2000) (see also section 2.1.2)), there is a growing literature on when, why and how agents should adopt existing norms (Verhagen 2000, Sartor 2001, Dellarocas and Klein 2001, Castelfranchi et al. 1999). The work presented here is along these lines – it considers expectations about social behaviour as given (regardless whether these stem from observation or are pre-structured) but not as unquestionable. Although agents are not likely to modify global expectations (viz. interaction frames) in the rather restricted scope of their decisions and actions, they may choose to deliberately break them in order to achieve their goals, if this seems appropriate. From a norms perspective, we therefore adopt an agent-centric view (Alonso 1998, Alonso 1999) that focuses on how agents can take knowledge about the social context into account when devising an individual agenda in a strategic way (rather than developing top-down models of how agents should balance existing social norms with their own goals (Hogg and Jennings 1997)).

This "autonomy in the face of social expectations" is one of the distinctive features of our approach. It is the result of developing methods for *local* reasoning about expectations resulting from *social* processes. Since agents may deviate from existing expectations, there is a chance that existing norms may change, i.e. our view implies an *evolutionary* outlook on norms. This norm evolution is closely related to the evolution of the meaning of communication, which brings us to the second norm-related theme, namely communication semantics.

Communication semantics

In social theory, all norms eventually boil down to communicative norms, i.e. any social expectation is reflected in some communicative behaviour of the members of a society. Conversely, we might ask "what is the meaning of communication in the face of social expectations?" Adopting this perspective, frames can be turned into knowledge about the meaning of communicative actions (rather than social plans or interaction patterns). This is because they capture information about and empirical evidence for the consequences of communication, and from an agent point of view, the effects of utterances are what matters about their semantics (cf. our assumptions in section 2.2.3).

It is very important to understand that adopting this communication view on frames does not merely constitute a purely theoretical re-interpretation of their function. Quite the opposite is the case: this view is of vital importance for a full understanding of interaction frames, because learning and applying frames means constantly (re-)constructing the meaning of agent communication. This aspect is captured by the notion of *empirical semantics* that we propose for agent communication (see section 4.1). This notion is based

[18] *Cognitive* expectations are expectations that are adapted according to observed, whereas *normative* expectations are (rather) immutable.

on constructing statistical models of meaning which are derived through an application of principles stemming from different sociological schools of thought, all of which share a *constructivist* foundation (the theories of Mead and Goffman, but also sociological systems theory (Luhmann 1995)).

The empirical semantics approach is most easily explained as "defining the meaning of communication through its consequences". These consequences are the effects of utterances as experienced by those participating in and observing communicative processes in a given social context. As mentioned in section 1.2 (p. 7, this implies an *evolutionary* view of semantics, since agents constantly change communicative expectations through their actions.

Most of the existing ACLs (such as the languages KQML and FIPA already mentioned in section 2.1.5) and hence also most frameworks for ACL semantics use speech act theory (Austin 1962, Searle 1969) as their theoretical foundation.

Speech act theory is based on the principle of viewing messages as actions. Thereby, it distinguishes between *locution* (physical utterance), *illocution* (desired effect) and *perlocution* (achieved effect) when analysing the meaning of a message. Describing the semantics of an ACL used in a MAS successfully mainly depends on whether the link between illocution and perlocution can be explained, i.e. whether we can describe the *effects* of utterances (those desired by the sender and those brought about by the recipient of the message) solely in terms of the speech acts used. Various proposed semantics suggest, however, that it is necessary to either resort to the mental states of agents (Cohen and Perrault 1979, Sadek 1991, Cohen and Levesque 1990b, Singh 1993, Cohen and Levesque 1995, Kumar et al. 2002) or to publicly visible commitments (Pitt and Mamdani 1999a, Pitt and Mamdani 1999b, Rimassa and Viroli 2002, Fornara and Colombetti 2002, Guerin and Pitt 2001, Singh 2000) in order to capture the semantics of speech acts, i.e. to aspects of the system that are *external* to the language itself.

Unfortunately, in *open* MAS, which are the target domain for the methods we propose, it is not clear how specifications of mental attitudes or social commitments can be linked to the observed interactions. As mentioned before (section 2.1.3), these systems are characterised by dynamically changing populations of self-interested agents whose internal design is not (completely) accessible from the other agents' or their designers' point of view. How can we make predictions about agents' future actions if the semantics of their communication is defined in terms of mental states or commitments not related to the design of these agents? The only hypothesis we can form is that the effects messages have had in the past are representative for their future effects, and this forces us to employ an empirical approach.

In fact, our theory of ACL semantics is therefore less closely related to ACL research than it is to the *language evolution* theme. From a linguistics perspective, what we attempt to use as a semantic model for communication is an "empirical pragmatics", even though our motivation is not, as that of linguists', to understand how (human) language works.

This is reminiscent of the works on robotic language evolution by Luc Steels (1998) and his group[19], in which robots were made to learn common vocabularies of objects they perceive by exchanging imaginary strings when talking about them. In contrast to these works, we are neither interested in robotic perception of objects and concept formation nor in

[19] The interested reader may consult (Steels and Vogt 1997) and (Steels 2003) for excursive summaries of this research.

negotiating words for these. Rather, we are interested in agents who negotiate the semantics of messages in the context of interaction flows to achieve fruitful coordination. As we are only concerned in the function of messages as "markers" for different paths of interactions, we also ignore issues such as syntax (but see Gmytrasiewicz 2002, Gmytrasiewicz, Summers and Gopal 2002). Ultimately, we hope to prove that agents can develop emergent ontologies (Maedche and Staab 2001, Behrens and Kashyap 2002) of interaction by using the empirical semantics approach.

Interaction protocols

In the previous paragraphs, we have argued that interaction frames are abstractions of certain classes of interactions and that they should guide agents' behaviour in a given social context. Pragmatically speaking, this perspective seems very similar to that of *interaction protocols*, which are the primary means for managing communication processes in MAS in the traditional view (see section 2.1.5).

Interaction protocols have two primary functions: Firstly, they define admissible message sequences (by means of state-chart diagrams, finite automata or other representations). The fact that they are common knowledge among agents renders agents capable of understanding the meaning of a specific message in a conversation. Secondly, they describe the control flow model of (a restricted kind of) conversations, so that agents can plan their communicative actions in a same way as they plan physical actions.

In the face of these characteristics, one is tempted to ask *"Is 'interaction frame' no more than a different name for 'interaction protocol'?"* This is not true for several reasons:

- Frames are not globally pre-defined, definite patterns of conversations. Although there might be some *a priori* information about existing frames, they are cognitive constructs, adopted and adapted according to the choices of the individual.
- Frames need not be as strict as protocols. Instead of defining precise message sequences, they might contain only rough constraints regarding the properties of a class of interactions (which makes them much more similar to *conversation policies*). In particular, frames may be refined while they are being executed, as new information comes in.
- Frames contain information concerning the "environment" of the communication process, i.e. information that not only explains which messages are being uttered. This may be information about participating actors and their mental states and about physical conditions that are necessary for the frame to be executed (cf. chapter 3).
- Frames can be broken. If an agent deviates from the admissible sequences a protocol defines, we consider the protocol execution to have failed (the conversation ends, a message is thrown, etc.). In the framing view, agents must cope with deviance on the side of others, they must somehow react to it, and, ideally, they should incorporate the unexpected behaviour of the other in their long-term strategies.
- Frames evolve. Unlike protocols, frames may change over time: new frames may be constructed, obsolete ones may be deleted, the categorisation of different situations into a set of frames may be revised, etc. This requires that the experience with certain frames is stored with them, which clearly distinguishes them from protocols.

That said, MAS in which agents employ protocols instrumentally according to their private goals (see, for example, Dastani et al. 2002) and that offer a fixed set of immutable protocols might count as very simple frame-based systems in a very lenient view. But even such systems would not count as "almost frame-based" if agents in them do not store their experiences with the protocols and exploit this experience (and they would still lack the important emergent character of truly frame-based MAS).

2.4 Summary

This chapter laid out the background knowledge necessary to understand our approach and how it relates to different disciplines. We first introduced the fields of Distributed Artificial Intelligence and Socionics and then gave an overview of interactionist socionics, the branch of Socionics that uses computational models informed by the sociological theories of symbolic interactionism. Since the theories of Mead and Goffman serve as starting points for the social reasoning architecture we propose, these theories were given particular attention. This discussion resulted in a list of assumptions that underlies the material presented in subsequent chapters.

To understand how our research fits into the state of the art in multiagent systems research, an extensive survey of related themes was included, our primary intention being to show analogies and differences to other work that deals with similar issues, such as social learning, strategic communication and rational agency.

A core insight from this analysis is that interaction frames and framing are novel concepts for DAI that can be valuable in the process of devising socially intelligent agents capable of successful operation in open MAS, because they rest on assumptions that are particularly well-suited for the problems associated with this kind of systems. The following chapters will show how this can be done in practice.

3. The InFFrA Abstract Agent Architecture

The *In*teraction *F*rames and *Fr*aming *A*rchitecture InFFrA is an abstract agent architecture based on the concepts of *frames* and *framing*. It provides a schema for building social reasoning components for agents who store and organise their interaction experience in the form of frames and employ these frames strategically in interaction situations. InFFrA is abstract in the sense that it does not include specifications of concrete, implementable data structures and algorithms (like, for example, the m^2InFFrA model proposed in chapters 4 and 5). It rather defines a generic framework based upon which concrete methods can be developed. The architecture consists of two main elements:

1. A model of computational interaction frames that describes the information necessary to adequately capture the nature of different patterns of interaction among agents.

2. A control flow model for framing that addresses the computational processes needed to process and apply interaction frames.

To give a feel for the kind of knowledge captured by frames, we will first present an intuitive example. From this example we will derive desiderata for a computational operationalisation of frames and framing. After this, this operationalisation will be presented together with examples. A final section is devoted to a critical discussion of the architecture.

3.1 An Example

To understand what frames and framing mean, a wedding may serve as a very good example from everyday life. For this purpose, we first have to conceptualise "procedure of getting married" as a communicative process, which, for the sake of our example, starts with a man proposing to his female partner and is completed upon departure to honeymoon after the wedding festivities.

Obviously, this process involves a huge amount of communication. Thus, although the entire *trajectory* (Goffman 1974) of communicative actions that can be traced in this procedure could be conceived as a single frame, it makes sense to split it into different subframes (especially because it probably spans a period of several months in its entirety).[1] One possibility for such decomposition would be to have a frame for each interaction phase in the wedding procedure, i.e. proposition, declaration of the intention to marry

[1] In fact, the very use of the term "frames" alludes to frames on a celluloid roll that capture scenes in a film and suggests this decomposition.

towards other parties (family, friends, etc.), organisation of the wedding and the honeymoon, the wedding ceremony itself, and the wedding party. Several arguments speak for such a decomposition:

- It obviously exceeds the cognitive capacities of participating actors to project this entire undertaking in planning their activities. Therefore, it seems natural to split it into fairly self-contained units that can be practically used to cope with the different situations that will occur until completion of the "task" as a whole.

- The different sub-procedures are often loosely coupled. In particular, the individuals who participate in some of these interactions need not be aware of others (the priest need not be aware of the precise arrangements for the party, the travel agent who assists in planning the honeymoon does not know the details of the wedding ceremony, etc.). This helps reduce the overall complexity of the whole system of interleaved interactions.

- Singular interaction processes are highly context-dependent, so that it is not necessary to dispose of all details of the entire procedure in a particular interaction that belongs to a particular phase. For example, it is rather unlikely that the woman will demand of the man to be able to present all the details about the ceremony at the time of proposing. In fact, frames can even provide a context for other frames themselves in the sense that certain interactions are pre- or post-conditions of others (no honeymoon without the formal act of getting married, (almost) no possibility of calling off the wedding party once guests have been invited, etc.).

That said, it may of course be reasonable to maintain a hierarchy in which the "wedding frame" is a rather abstract, very general description of the entire process that is connected to several sub-frames and manages the dependencies between them. What is important in any case is that the frame should be *manageable* in size (either detailed and describing a relatively short process or more abstract for a more complex activity) so that it can be used in resource-bounded reasoning processes, and that it *makes sense* as a self-contained unit of interaction.

A second important aspect of the interaction frame view is that the frames employed in such a complex activity are primarily concerned with the *interaction* that is going on (hence the name). Steps of the procedure that are private actions of the parties involved do not count as parts of frames, unless they are brought to the attention of communicating actors and contribute to the correct "alignment" of the joint action. For example, it does not matter how the cook prepares the wedding dinner, as long as it is served on time and satisfies the expectations of the guests (even bad food will not inhibit the flow of interaction unless someone protests openly and breaks the joyful, festive ambience).

Thirdly, although some of the interactions in this process may be characterised by physical distance and delayed reaction (e.g. mailing invitations and getting a response by telephone, submitting documents at the local registry office, etc.), frames are primarily used to manage *face-to-face interaction between co-present actors*, such as the proposal, the communication between couple and priest at the church, etc. The idea is that frames are compact information structures that can be processed on-line in a timely fashion during interaction encounters to achieve flexible, strategic behaviour. Their purpose is not to capture complex plans that are constructed off-line and executed later.

So far, we have argued for a communicative orientation of frames, but it is misleading to think that information about communicative actions suffices to describe a class of interactions completely. This is because the perceived communicative actions may not always provide sufficient information to unambiguously determine *"what is going on here?"* or some of the communication that is necessary to correctly interpret the situation may not be available to (or deliberately hidden from) some of the participating actors. The alleged future husband may, for instance, be playing a (very mean) joke on his partner while proposing. Noticing that his mates are secretly watching from behind might force her to *re-frame* later on and understand that the proposition was only staged for this "audience", which in turn would form an integral part of the actor set participating in the "joke proposition" frame. In fact, virtually *all* forms of untruthfulness and deception are based on using communicative actions that do not mean what they are supposed to mean. This true meaning can only be determined by using extra-communicative information.

The context-sensitive character of communication becomes obvious if we look at the many other ways in which the wedding may be framed. For example, the wedding ceremony itself might not be a "real" ceremony, but, say,

- a staged ceremony in a theatrical play or film,
- a rehearsal that takes place prior to the actual wedding,
- an "experiment" that the priest conducts to test the acoustics of the church,
- the playful staging of a wedding by two children in an empty church,
- a "mock" simulation of a marriage between a couple who do not actually intend to get married, but want the audience to assume they are, or
- the implicit forging of an alliance between different states the children of whose monarchs are getting married.

The differences between these variations of the "normal" wedding ceremonial can be very subtle in communicative terms, or they may not even exist at all. For instance, if the ceremony is a scene in a film, "suspension of disbelief" actually requires of the actors to give no indication of the staged character of the situation. The spectator can only understand its meaning by reflecting upon the fact that he is watching a film and using the background knowledge that such films are not "real".

A final notable property of interaction frames is that they must generalise from particular interactions. This is not only because generalisation makes them more powerful as tools for representing interaction knowledge. Also, it is necessary to make agents re-use a learned frame. Without this re-use, the frame would actually be useless (there is no point in trying to behave appropriately at a nuptial ceremony if every such ceremony is completely different). Generalising from individual interactions also helps to speed up the learning process, as information from different instances can be combined to obtain a more adequate picture of the respective class of interactions. For instance, an experience of a wedding feast with many formal speeches may be added to an existing wedding feast frame which was originally constructed after a wedding without any speeches. The result of this combination might yield a more accurate description of wedding feast as a feast "which may or may not involve formal speeches".

This example illustrates what the essential properties of frames are, and thus aids in understanding the computational models of frames and framing we will propose below. Next, we shall briefly discuss desiderata for these.

3.2 Desiderata

Computational models of frames and framing should fulfil the following requirements:

Communicative focus The description of a class of communication processes must be the core element of a frame. Any additional information may be used to support the description of the interaction process, but communication is what the frame ultimately is *about*.

Structuring & categorisation The different types of interactions an agent is involved in should be categorised into different frames. A set of frames should structure the "social world" for the agent and it should ideally provide guidance for any encounter the agent may find himself in.

Strategic application A frame must contain information about the consequences of a certain class of interactions, so that the agent can use it to make strategic decisions with respect to his private goals. In particular, the agent should initiate interaction processes if frames suggest that certain goals can only be achieved through joint action.

Conventional character The reason for using frames is to reduce complexity in interaction and to obtain guidance about how to act. Therefore, framing should lead to selection of an appropriate frame and to compliance with the actions it prescribes, so that a frame can be used as a simple action convention once it has been selected. (This does not mean that the agent should not be able to revise an earlier framing decision, if necessary.)

Persistence Frames should be cognitively represented by agents and it should be possible to maintain them regardless of whether they are being used or not. Unlike plans, they are not created "on demand" and erased after the plan was carried out. Instead, they are either the result of long-term observation, imitation or experimentation with different forms of interaction. Frames have their own history, they relate to other frames, and may be re-combined or modified to obtain new frames.

Plasticity As agents store frames as local representations of interaction knowledge, they are open to manipulation by their "owner". This means that they can be adapted to the needs of the agent (e.g. by deleting obsolete frames, adding information to a frame, etc.), but it also implies that the privately maintained frames can be far from the reality of the interaction that takes place in a system.

Context dependence Application of a frame is subject to certain conditions and it brings about certain conditions. To provide hints about the contexts under which it is relevant, a frame should therefore include information about these conditions.

Abstraction To capture knowledge about an entire class of interactions, frames must abstract from individual interactions. So, instead of describing concrete agents that participate in the frames, they should employ models of roles and relationships to describe these, and instead of defining precise messages that are to be communicated, they should employ more abstract patterns of interactions that allow for certain (varying) degrees of freedom.

To summarise, frames should be abstract knowledge structures that describe classes of recurring, relevant interactions together with the context in which these may occur. Agents should be able to shape them according to their needs and to employ them in their goal-directed reasoning.

3.3 Interaction Frames in InFFrA

Interaction frames are the data structures on which the reasoning model of InFFrA operates. A frame is a structure that describes a class of interactions. At a *descriptive* level, it specifies

- the roles held by the participants of the interactions,
- the courses the interaction may take,
- the context in which the frame may occur, and
- the beliefs held by the interacting parties.

Additionally, a frame includes *meta-level* information, which captures

- the status of the frame during execution,
- relationships between the frame and other frames,
- experiences with the frame in past interactions, and
- assumptions regarding which of the interacting parties has knowledge of the frame.

Table 3.1 summarises these attributes (or "slots"), which will now be described in more detail.

3.3.1 Descriptive attributes

Roles and relationships

To define a set of interaction processes, it is necessary to specify which actors play a part in them. In the example of section 3.1, certain interactions were only possible between specific actors, and some of these actors were actually defined *by* the respective type of interaction (e.g. groom and bride have to participate in the ceremony, and they only become groom and bride by virtue of the ceremony, in turn).

As we have argued, these actor definitions must be as generic as possible, if we want the frame to be applicable to a large number of situations. For this reason, they should not describe individuals or groups but whole classes of these. This is where roles and relationships come into play.

descriptive attributes	
roles & relationships	define the participants in the concerned interaction
trajectories	describe courses of interaction that are possible under this frame
context	situations in which the frame is relevant and states of the world it brings about
beliefs	domain knowledge that is necessary to carry out the interaction correctly and knowledge that results from frame execution
meta-level attributes	
status	instantiations of the descriptive attributes with values for the current encounter
links	a set of relationships of the frame to other frames
history	experience with the frame in past interactions
extension	distribution of knowledge about the frame

Tab. 3.1: Frame attributes, categorised into descriptive and meta-level types. Attribute names are given on the left-hand side with corresponding descriptions on the right-hand side.

Without intending to introduce a fundamentally new, full-fledged model of roles that is adequate for MASs (see, for example, (Kendall 1998, Gutknecht and Ferber 1998, Weiß et al. 2003)), we can use the equation

$$\textit{role} = \textit{social expectation} + \textit{social position}$$

as an underlying theoretical model. It states that a role is given by a set of expectations associated with anyone who is filling this role and by a social position, i.e. the relative position of the role to others in a given social (institutional, organisational) setting.

Social expectations can refer to

- *behavioural expectations*, i.e. descriptions of the behaviour the role may exhibit and the capabilities it has;
- *deliberative expectations*, which describe the beliefs, goals and preferences the holder of a role has, the tasks it is seeking to complete, but also the way it forms its intentions and commitments;
- *normative expectations*, specifying the rights and duties, obligations and permissions associated with a role.

The social position of a role, on the other hand, is obtained by analysing relationships towards other roles. This is the reason why relationships are added to the roles slot of a frame: Without the context of other roles, it is not possible to define what is distinct about a role.

Social relationships that specify the position of a role can be of the following kinds:

- *acquaintance*: one role is aware of the existence of the other (this relationship can but need not be symmetric);

- *dependence*: one role requires certain activities from other roles to achieve its goals;
- *similarity*: roles share certain attributes, such as expectations associated with them or relationships towards third parties;
- *aggregation*: a role makes part of another role (in a group, or in a conceptual distinction between sub-roles of one actor (e.g. father and grandfather, who can be the same person));
- *representation*: a role acts "on behalf" of someone else; this is particularly important in collective roles (groups), in which the group needs certain individuals to take concrete action (the group *delegates* these actions to its representative(s));
- *segmentation*: relates a number of roles to each other that partition a more general role (e.g. "faculty" and "staff" segment "university employee");
- *inheritance*: a role specialises some other role by introducing additional expectations and relationships.

Returning to the link exchange scenario of section 1.1, a specification of roles and relationships for a linkage negotiation, in which a "linkage brokering agency" mediates in the attempt of a low-traffic commercial site to buy a link from a high-traffic commercial site might look as shown in table 3.2.[2] In this model, there is a link client role (who needs a high traffic site to link with it so that the client can improve the traffic of his own site), a link seller role (who has high Web traffic and can profit from selling links to others), and a link broker role (constantly searching the Web for suitable linkage "partners", negotiating linkage contracts with them and earning a commission from both parties).

For this example, we use the following predicates to describe roles and relationships:

- $can(R, A, C)$ denotes that R can perform action A if condition C holds, and $does(R, A, C)$ denotes that A is the typical behaviour of R whenever C holds.
- $goal(R, S)$ is true, iff R has S as a goal (S can be an action or a state of the world).
- $belief(R, B)$ holds, iff R thinks that B is the case.
- $permission(R, X)$ and $obligation(R, X)$ mean that R may/must do X.
- $needs(R, S, G, A)$ states that R needs role S to perform action A for goal G.
- *acquainted* is a symmetric relation between two roles, $inherits(S, T)$ indicates standard inheritance (in the sense that S has all the attributes of T and more), *group* defines a collective role out of a set of existing ones.

The other predicates used in the example are domain-specific and should be fairly self-explanatory.

According to the role and relationship specification of table 3.2, that both seller and client agent can add a link to another site (the site of an agent A is given by the function

[2] To avoid introduction of unnecessarily proprietary notation, we will use ordinary first-order predicate logic in the examples presented in this section (with the convention of capitalising variables and assuming universal quantification whenever quantifiers are omitted. For more specific purposes, other notational means can be used in concrete instances of InFFrA, such as RNS (Weiß et al. 2003) for roles.

Roles: *lc* – a link client, *ls* – a link seller, *lba* – a link broker agent.
Expectations for *lc*:

behavioural:	$can(lc, payTo(\cdot, P), P < 20,000\$)$, $can(lc, addLink(siteOf(lc), X), site(X))$
deliberative:	$goal(lc, increaseTraffic(lc))$, $belief(lc, highTraffic(X) \wedge addLink(X, siteOf(lc)) \Rightarrow increaseTraffic(lc))$
normative:	$contract(lc, owner(X), C) \Rightarrow obligation(lc, pay(owner(X), price(C)))\wedge$ $permission(lc, claim(lc, owner(X), addLink(X, siteOf(lc)))$

Expectations for *ls*:

behavioural:	$can(ls, addLink(siteOf(ls), X), site(X))$ $can(ls, sell(ls, owner(X), link(C)), site(X) \wedge contract(C))$
deliberative:	$goal(ls, increaseTraffic(ls)), goal(ls, makeMoney(ls))$ $belief(ls, highTraffic(ls))$, $belief(ls, sell(ls, \cdot, link(C)) \wedge price(C) > 10,000\$ \Rightarrow$ $makeMoney(ls))$
normative:	$sell(ls, A, link(C)) \Rightarrow$ $permission(ls, claim(ls, A, payTo(ls, price(C)))\wedge$ $obligation(ls, addLink(siteOf(ls), siteOf(X)))$

Expectations for *lba*:

behavioural:	$can(lba, spotLinkagePartners, \text{true})$, $does(lba, offerServices, clientsFound)$
deliberative:	$goal(lba, makeMoney(lba))$, $belief(lba, contract(X, Y, C) \wedge proposes(lba, C) \Rightarrow$ $claim(C, clients(C), payTo(lba, commission(C)))$ $payTo(lba, P) \wedge P > 10,000\$ \Rightarrow makeMoney(lba)$
normative:	$contract(A, B, C) \Rightarrow permission(lba,$ $claim(lba, A, payTo(lba, price(C)))\wedge$ $claim(lba, B, payTo(lba, price(C))))$

Relationships:

acquaintance:	$acquainted(lba, ls), acquainted(lba, lc), \neg acquainted(lc, ls)$
similarity:	$shareCapability(lc, ls, addLink), shareGoal(lc, ls, increaseTraffic)$
dependency:	$needs(A, B, addLink(siteOf(B), siteOf(A)), increaseTraffic(A))$, $needs(A, \cdot, payTo(A, P) \wedge P > 10,000\$, makeMoney(A))$, $needs(lba, clients(\cdot), makeMoney(lba))$
aggregation:	none
segmentation:	none
inheritance:	$inherits(lba, brokerAgent)$

Tab. 3.2: An example specification of roles and relationships. Role expectations are given for each of the three roles while relationships are summarised for all of them.

$X = \mathit{siteOf}(A)$, and $A = \mathit{owner}(X)$ holds, respectively). The client is additionally characterised by having some money while the seller's "capital" is his popularity. Both want to increase their popularity, but the seller has the additional goal of making some profit from the linkage deal (at least \$10,000). In this link brokerage scenario, seller and buyer need not be acquainted, since there is a broker who constantly looks for suitable linkage partners and offers his services to them. If he manages to fix a contract that he has proposed, he will get a commission (which is his business objective, after all). The terms of a contract C are its price $\mathit{price}(C)$, the link that will be laid due to the contract $\mathit{link}(C)$, and the commission $\mathit{commission}(C)$ the broker will obtain.

Fixing a contract is associated with several normative expectations on all sides. The buyer (client) is supposed to pay ($\mathit{payTo}(A, P)$ denotes that P units of money are paid to A), and he is allowed to request that the link be added by the seller. The seller's permissions and obligations are "inverse" to those of the buyer. The broker is, of course, allowed to claim his brokerage and is not committing himself to any further action. We use $\mathit{claim}(A, B, X)$ to denote the event that A claims action X from B.

As for relationships, buyer and seller share a capability and a goal in this setting. All agents are inter-dependent in the sense that they need "traffic" and "money", which are resources currently owned by others. The broker needs some clients to make some money, of course (as further above, we use "$\cdot$" as a wildcard symbol). Finally, to give an example of inheritance, the linkage broker agent might specialise a more generic *brokerAgent* class defined elsewhere.

The formalisation used in this example is not prescriptive for InFFrA-based agent designs, and many other specification languages are conceivable whose suitability will depend on the application domain and the computational means used for implementing the respective architecture. Strictly speaking, the specification language used is not part of the core InFFrA architecture, and this should be kept in mind throughout the descriptions of all frame slots, where we will use similar example formalisations. Note, however, that the predicate logic used in these examples is in keeping with the requirement that frame representations be computationally tractable, so that they can be used by agents themselves (as formulated in section 1.2, p. 6).

To "detach" the abstract modelling of certain frame attributes from concrete formalisation, we will introduce simple diagrams for them. Figure 3.1 shows such a diagram for the role and relationship model.

Trajectories

Specifying *who* is taking part in a class of interaction processes that is to be represented by a frame is a prerequisite to describing *what* these actors will actually do in the interaction. This is done by defining a model of the possible *trajectories* that are captured by the frame, i.e. providing a description of courses of joint action relevant to the frame.

Such a trajectory model is the most essential constituent of a frame definition, since it allows agents to predict the actions that may occur in an interaction and thus to reduce the uncertainty about potential behaviours of interacting parties. Additionally, it specifies the control flow of the message exchange during an unfolding interaction so that it can also be used for monitoring communication and regulating the contributions of the agent who is using it.

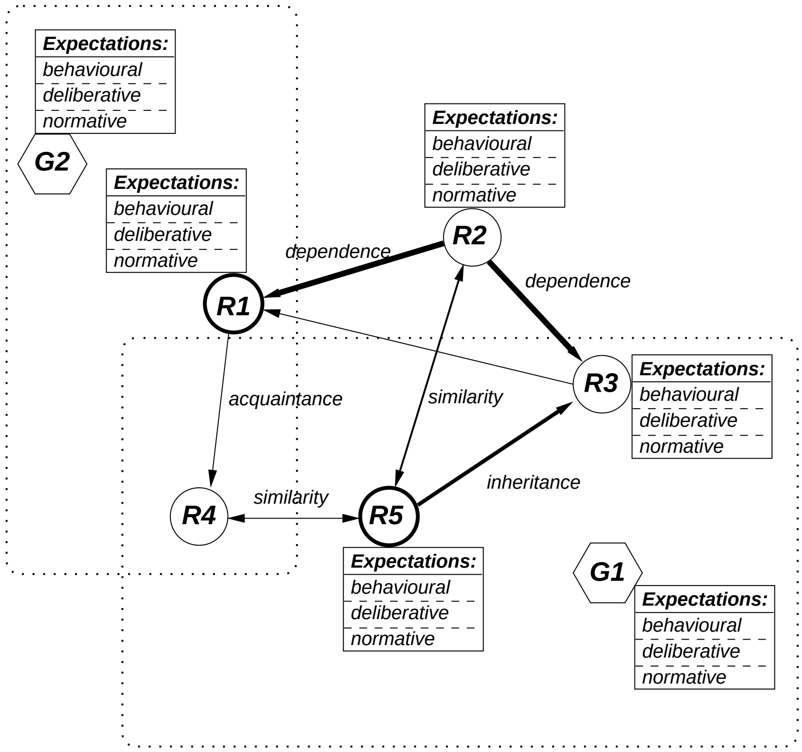


Fig. 3.1: Role and relationship model. Shows roles as round nodes with expectations in boxes attached to the nodes and edges between them to denote relationships (strength of relationships is indicated by edge thickness). Roles are aggregated to (overlapping) groups (labelled with hexagons) by means of rounded, dotted boxes. Each role has expectations attached to it.

As with other frame attributes, InFFrA does not impose a particular formalism for trajectory models. This is because different application domains require different levels of abstraction and precision. Sometimes, simple constraints may be sufficient. In other situations, a full-fledged, detailed model of the surface structure of admissible message sequences may be necessary. In that, two dimensions are relevant for the description of trajectories:

1. *Rigidity:* The prescriptions made by the "rules of conduct" contained in a trajectory model can range from weak recommendations (e.g. default rules that may be overridden) to strong, mandatory rules. For example, the wedding ceremony frame requires that the priest declares the couple as "husband and wife" at the end of the ceremony, but it does only give rough raiment guidelines.

2. *Specificity:* Trajectory models can describe precise actions or they may just provide deontic specifications of behaviour, such as responsibilities, rights and duties. For instance, the couple is supposed to thank the guests for the wedding presents at some point, but there need not exist a precise habitual scheme that explains how this should be done.

For the purposes of our examples, we will use partially ordered sequences of message patterns between the roles defined in the role and relationship slot. These patterns have the format

$$performative(sender, receiver, content)$$

where *performative* is a message performative in the sense of speech act theory (Austin 1962, Searle 1969), *sender* and *receiver* are names of defined roles and *content* is the content pattern of the message, i.e. a logical expression which may contain variables for (sub)-formulae and objects.[3] As the relationship of InFFrA to speech act theory will be covered at length in section 4.1, it suffices for the moment to think of the performatives as labels for different types of messages such as requests, suggestions, queries, etc.

Figure 3.2 shows a graphical notation that can be used to depict such message pattern sequences similar to the protocol diagrams of Agent UML (Bauer, Müller and Odell 2000, Odell, Parunak and Bauer 2000b, Odell, Parunak and Bauer 2000a). This trajectory model for the "linkage brokering" example of table 3.2 describes how a link broker agent (*lba*) advertises the fact that it has spotted a good candidate for a linkage contract to the link client (*lc*). If the client is interested, he quotes the maximal price *MaxPrice* he is willing to pay, and if the potential link seller (*ls*) is also interested, he will announce the minimum price *MinPrice* he demands for the link. Note that *lba* is not forced to forward *MaxPrice* to *ls*, but that he can name any price *FlexPrice* during negotiations with the seller. Then, *lba* fixes the final price *ContrPrice* for the contract. The client may now accept or reject the contract. If he accepts, all parties have to fulfil their commitments: Seller and buyer have to pay commissions to the broker, and the seller has to lay the physical link, whereafter the link purchasing party has to pay the price fixed for the contract (the *do* performative is used here to denote physical actions, such as money transfer or Web site modifications).

What has been omitted in the example are logical constraints for the content of messages, which could have been added to the trajectory model. One useful constraint would be

$$MaxPrice \geq ContrPrice \geq FlexPrice \geq MinPrice,$$

for instance, to express that all parties will try to minimise/maximise their profit. However, as logical conditions are covered by the "context" attribute of the frame, they are deliberately left out in the trajectory description.

To increase the expressiveness of trajectory definitions, we can use iteration of trajectory sub-sequences as shown in figure 3.3. The diagram shown in this figure re-defines the middle part of the brokering example to include multiple negotiation iterations. Now, the seller can make counter-offers to the initial offer of the broker, accept the offer or simply quit at any point. If his offer at some point exceeds the maximum price the buyer is willing to pay, the broker terminates the negotiation. As such loops introduce a novel temporal element, it is now necessary to specify how variables will be bound to concrete values. To express, for example, that *FlexPrice* can be set to a new value by both parties in each iteration, we could either (i) use an explicit notion of time and label the variables with time-stamps ($FlexPrice(t)$, $FlexPrice(t+1)$, etc. where t is a global system variable that is always instantiated with a global time stamp) or (ii) use a statement $new(FlexPrice)$ to express that the variable is set to a new value in a particular trajectory step.

Many other formalisms exist for describing such interaction trajectories, such as finite-state machines, Petri nets, etc. Regardless of the choice of formalism, it should be ensured

[3] We omit further technical details at this point, because a pattern language that is very similar to the one used here will be formally defined in chapter 4.

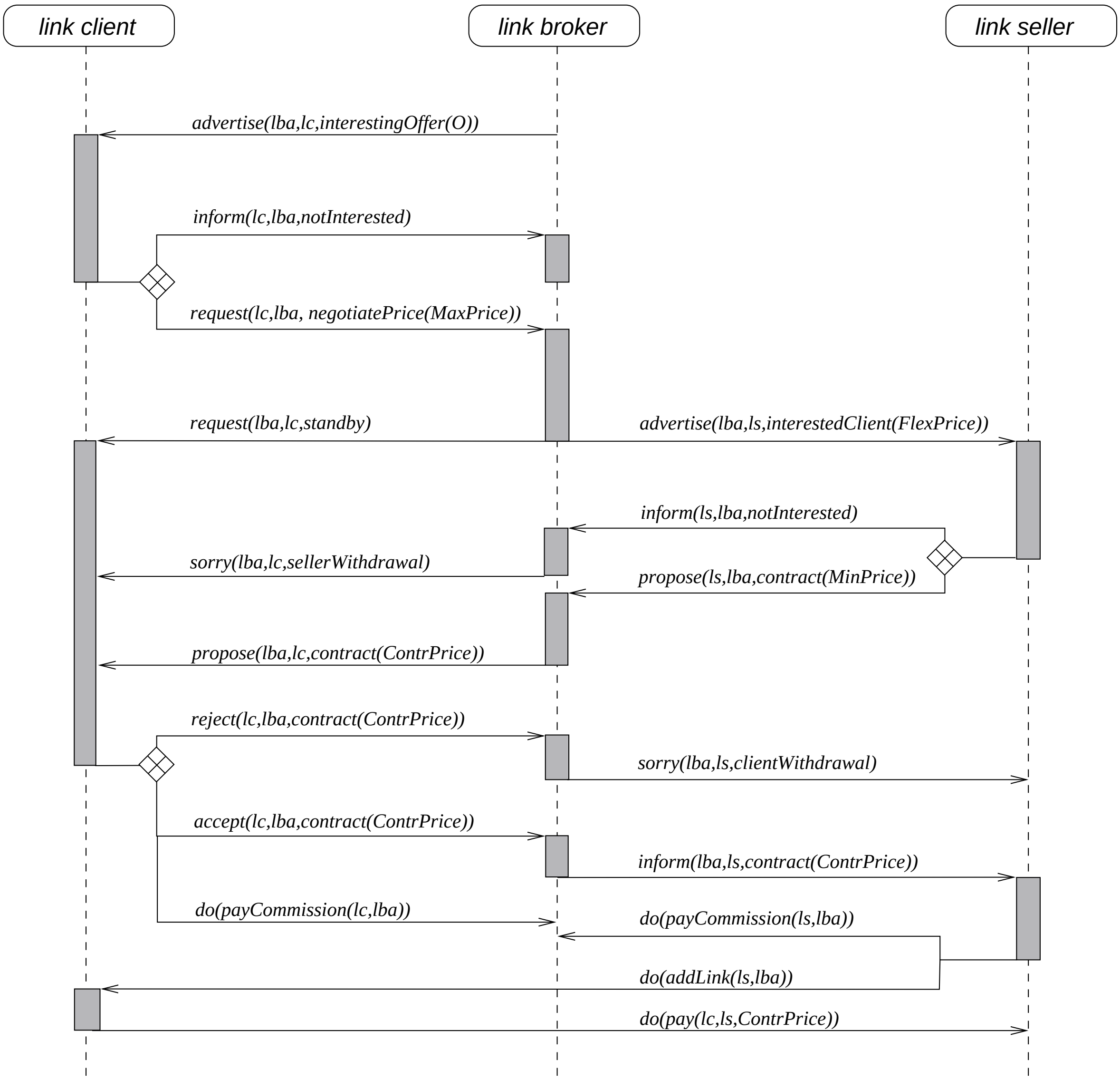


Fig. 3.2: Example trajectory model. Roles are shown as vertical "swim-lanes" with grey boxes for the internal reasoning processes of role fillers. The arrows that connect roles are labelled with the messages they denote. Branching indicates combination of several messaging actions, diamonds denote choice.

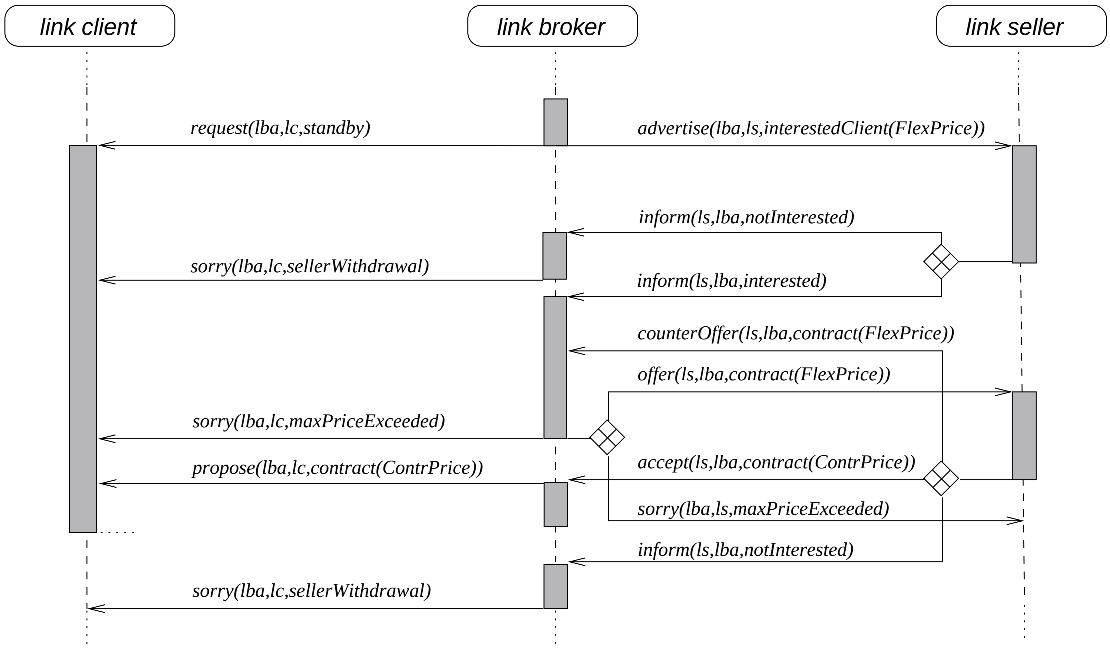


Fig. 3.3: Trajectory example with iteration. The diagram shows only the middle part of the trajectory between the *advertise* and the *propose* statement.

that (i) it is possible to capture the patterns of interaction that one wants to express through a frame, and (ii) that tractable methods are available for agents to validate whether the observed messages and actions comply with a certain frame or not, since this is one of the basic activities that are necessary to perform effective framing, as will be explained in section 3.4.

It is important to note that trajectories are the core element of a frame. They represent the actual observations that will be made during the interaction and are hence the primary means of selecting a frame in an interactive encounter. This implies that, although all other slots may contain additional background information which can be used to determine

1. when a frame is relevant and

2. how the observed behaviour can be interpreted,

this additional knowledge will *not* be used for predicting actions or selecting own actions. For these purposes, only the trajectory model matters. For example, although the behavioural expectations associated with the three roles in the brokering example may contain descriptions of what they "usually do", this need not be by any means connected to the brokering conversation. These expectations can, however, be used to determine whether a concrete agent qualifies as a broker, or to explain why a very powerful seller is willing to lay a link for a small amount of money by suggesting that he must desperately need the money.

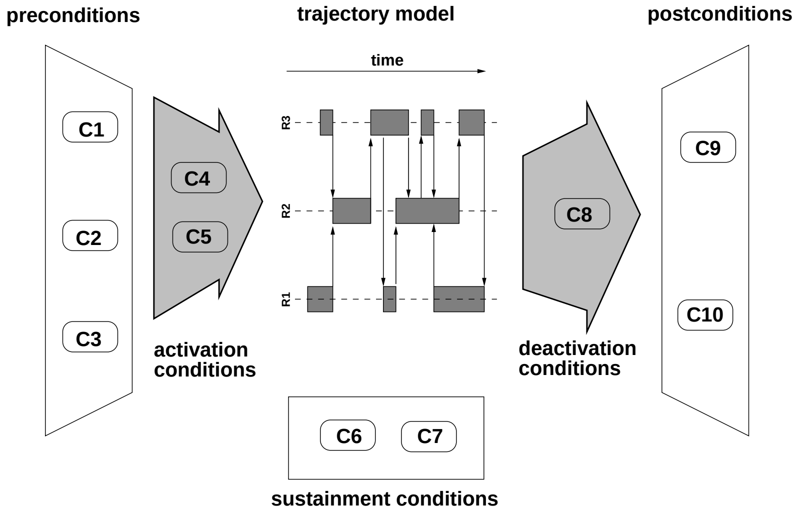


Fig. 3.4: Frame context model. Conditions $C1, \ldots, C10$ are grouped together in transparent polygons (enactment conditions) or shaded arrows (activation conditions). A flipped miniature version of the trajectory model is embedded between these different types of conditions to express that they must hold before, during or after its execution.

Context

The *context* of a frame defines *when* the interactions described by the trajectory model can occur by defining pre- and postconditions for its activation, and also conditions that are necessary for the interaction to be sustained.

Two categories of such context conditions can be distinguished:

1. *Enactment conditions:* These are necessary for the frame to be enacted properly in a physical sense and resemble very much operator conditions in classical AI planning. We can further distinguish between
   - *pre-conditions* that have to be fulfilled prior to the enactment of the frame,
   - *sustainment conditions* that concern the constraints that need to be maintained in intermediate steps of frame execution, and
   - *post-conditions* that hold after frame completion.
2. *Activation conditions:* These spawn activation and de-activation of the frame. Although they are not conditions in the physical sense, they can be used to provide explicit rules that reflect circumstances under which the frame is relevant.

Essentially, the difference between these two kinds of conditions is that failure to prove enactment conditions means that it is impossible to execute the steps of the frame trajectories under present circumstances. Activation conditions, on the other hand, are constraints that the constructor of the frame imposes on its use *himself*, conditions under which he would "voluntarily" choose or discard this frame. Figure 3.4 shows a graphical notation for the context of a frame.

A simple context model for the linkage brokering frame (from the point of view of the buyer) could be given by logical conditions for each of the condition types listed above and might look as follows:

1. *Enactment conditions:*
   - *pre-conditions:* $\neg existsLink(siteOf(ls), siteOf(lb))$
   - *sustainment conditions*: $contract(lb, ls, C) \Rightarrow money(lb, t) \geq price(C)$
   - *post-conditions:*

$$contract(lb, ls, C) \;\Rightarrow \Big(money(lb, t') \leq money(lb, t) - price(C) \\ \wedge\, existsLink(siteOf(ls), siteOf(lb))\Big)$$

2. *Activation conditions:*
   - *activation: $interestingOffer(O)$*
   - *de-activation: $contract(lb, ls, C) \wedge \neg addLink(siteOf(ls), siteOf(lb))$*

The enactment conditions in this example are fairly straightforward: The frame only makes sense if no link exists yet, otherwise there exists no offer O (see figure 3.2) that can be formulated by the broker. For the frame to be sustained, the buyer needs to dispose of at least $price(C)$ units of money if a contract C is to be fixed ($money(A, t)$ denotes the amount of money that A has at time t). Also, after the frame is over (at time t'), the buyer agent will have lost at least $price(C)$ units of money (maybe more, if he spent money on other things in the meantime).

It is interesting to note that different from the pre-conditions, the post-condition and the sustainment condition need only hold in case a contract is fixed. This is because the trajectory model allows for several paths of execution in which these conditions need not hold, for example if no contract is fixed. This is very important for the semantics of the frame: *if an enactment condition is required, this means that none of the trajectories that are possible under the trajectory model can be executed if this condition does not hold.*

As for the activation conditions of the frame, the buyer is willing to activate the frame if O is interesting. Note that $interestingOffer(O)$ here means that the agent who maintains the frame can prove this statement using his own knowledge base, which is very different from the same statement occurring in the *advertise* message of the trajectory model, where it was purported as being true by the broker agent. Sharing variables between trajectory model and conditions enables the framing agent to parametrise logical conditions with message contents that occur during the observed encounter.

Also note that this activation condition implies that the buyer agent will not even reply (at least not using the messages specified in the trajectory model of this frame) if O is not interesting, which illustrates that activation conditions have a strong impact on communicative behaviour. According to the de-activation condition, the buyer agent will cancel execution of this frame if the seller does not lay a link after a deal has been fixed. The rationale behind this might be that the agent will appeal to a legal institution if the seller does not comply with the agreement, or that he will simply not pay to minimise his own risk.

The semantics of such activation conditions are somewhat different from those of enactment conditions: *if an activation (de-activation) condition becomes true, the agent will* *activate (de-activate) the frame in the sense that some (none of the) trajectory sequence(s) defined by the trajectory model will be carried out.*

Beliefs

In the spirit of interactionism, the mental states of interacting individuals are not of any importance as long as interaction functions normally. This might lead to the assumption that the *beliefs* slot of frames, which contains information about the knowledge necessary to execute a frame correctly, is only of subordinate significance when modelling an interaction frame.

However, as with roles and relationships and contexts (cf. page 51), knowledge about the epistemic properties of the interacting agents may help interpret the ongoing interaction and assess the relevance of a frame. Moreover, information about agents' beliefs can be very valuable if the frame is *not* enacted smoothly, as it can help to "repair" an interaction that has gone awry. If an agent is aware of the knowledge his peer must have to play his part in the interaction, he can identify epistemic reasons such as ignorance, false assumptions or misalignment of distributed frame conceptions and try to eliminate these problems (for example by informing the other, asking for clarification, etc.).

The possibilities for specifying beliefs are manifold, both at the level of choice of formalism and at the level of the content of descriptions of epistemic properties. As concerns formalisation, such methods as epistemic logic (Fagin et al. 1995), graphical methods for probabilistic reasoning (such as Bayesian networks) (Pearl 1988), semantic networks and ontologies can be used, to name but a few examples. Content of belief descriptions may refer to such general things as a shared ontology or communication language, or to very specific knowledge such as the deliberative expectations associated with the roles in the example of table 3.2 (if we want to require that agents know all these facts for the frame to be enacted).

As before, we use a simple exemplary diagram (figure 3.5) to describe belief models and leave more specific definitions to concrete instances of InFFrA. Despite its exemplary character, it fulfils the minimal requirement for a belief model in InFFrA which is that beliefs must be ascribed to the roles defined in the role and relationship model of the same frame.

Summary

Taken together, the descriptive frame attributes describe what the class of interactions represented by a frame consist of, and this is nicely captured by the frame diagram shown in figure 3.6. A question that arises naturally is why the three non-trajectorial attributes are not merged into a single "conditions" slot, given that the role and relationship model, the context model and the belief model must be verified to determine whether an interaction encounter will match the patterns represented by the trajectory model. After all, they are all, in a sense, conditions for using or not using the frame.

Still, the following arguments can be put forward for upholding the distinction between these three attributes:

1. Conceptual clarity: Roles and relationships define the *who*, contexts the *when*, beliefs the *why* and trajectories the *how* of a class of interaction processes.

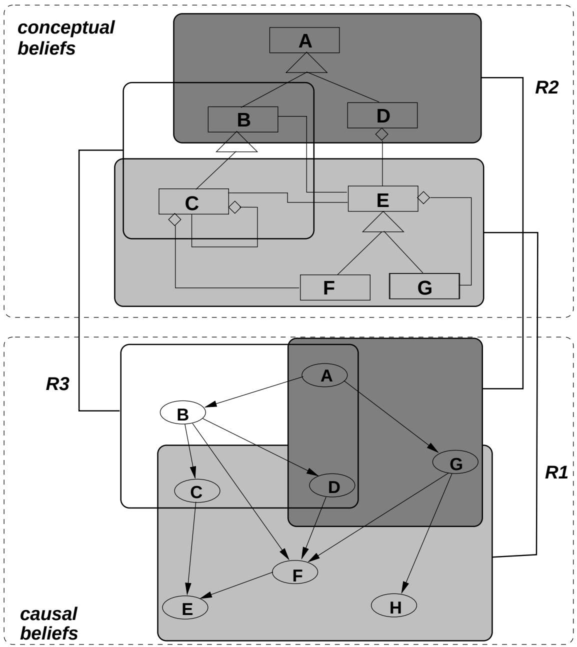


Fig. 3.5: Frame belief model with a taxonomic ontology and a Bayesian network as two examples of conceptual/causal knowledge structures necessary to implement the frame. The interconnected (shaded) sub-areas of the network are labelled with role identifiers from the role and relationship model to indicate the knowledge each role has.

2. Although there is some overlapping between role, context and belief models, there are subtle semantic differences:

   - The deliberative expectations in a role model do not express who *knows* about them (in extreme cases, a role filler may not even be aware of his own goals) and serve only as information which aids in "matching" a role against a concrete agent. Repeating these deliberative expectations in the belief model, though, requires specifying which roles are aware of them. It expresses the fact that this knowledge is a prerequisite for executing the frame.
   - At first glance, it may seem that behavioural expectations in role descriptions might have been included in the trajectory model instead. However, this model does not describe what an agent who plays a role will do in the context of the current frame, but rather what his general behavioural patterns are. For example, there may be behavioural cross-references between behaviours of a role that occur in several frames (e.g. refering to a link buyer in (i) the brokering frame and (ii) a civil court trial which is held because he does not pay for the link). These can be valuable in recognising someone as the holder of a role in different interaction encounters.
   - Beliefs and roles could have been integrated with enactment conditions, but a clear separation between conditions about the environment, the interacting parties, and their beliefs is helpful during framing decision making. This is because the fulfilment of environmental context conditions can be checked in a much stricter fashion than that of roles and beliefs, where the frame holder may be more lenient. For

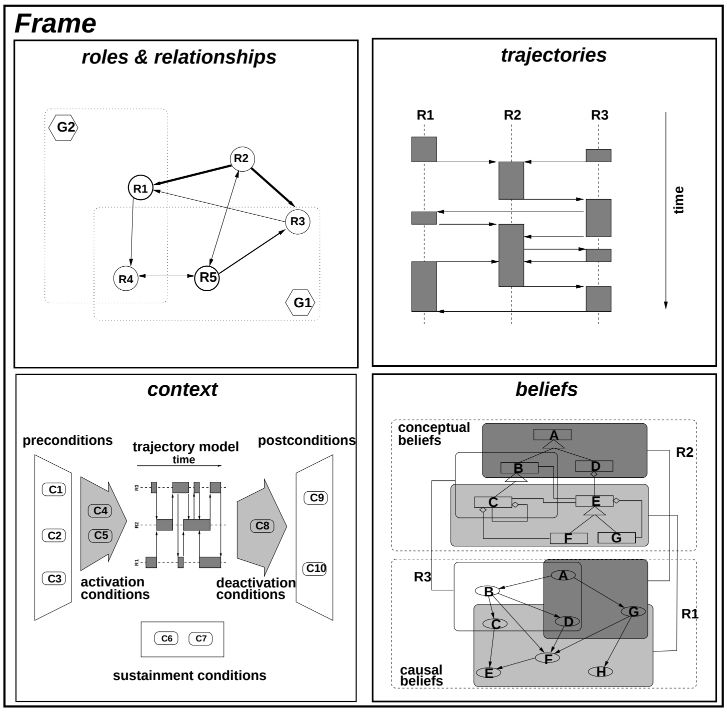


Fig. 3.6: Schematic view of an interaction frame containing the four descriptive attribute slots

example, if the link buyer has no money (context model), there is no point trying to execute the frame, but if he does not know the seller's goal (role and relationship model) this need not necessarily impede the correct enactment of the frame.

That said, it may often be useful to combine roles, context and beliefs to a single class of "conditions" for implementation purposes, i.e. to treat them all as logical constraints that have been verified using the same inference procedure. In fact, the concrete architecture presented in chapters 4 and 5 utilises precisely this method.

3.3.2 Meta-level attributes

While descriptive frame attributes specify the properties of a class of interactions in terms of actors, conditions and courses the interaction may take, *meta-level attributes* contain information about the frame itself. This information is useful to manage the frame in the context of a social reasoning process based on the concept of framing (see 3.4).

InFFrA interaction frames maintain four kinds of meta-level attributes called *status, links, history* and *extension*.

Status

The status slot is used to store information about the role of the present frame in the currently ongoing interaction.

If the frame is not relevant at all in the current encounter, the status attribute is (trivially) devoid of any information or may at most contain some indication of the fact that the frame is not being used at the moment. Otherwise, the status attribute contains a *status field* for each descriptive attribute. These status fields relate the general statements about roles, trajectories, activation and beliefs given in the descriptive attributes section to the specific situation the agent is in. In other words, status provides the "reverse generalisation" for a frame that maps the abstract frame to a particular, concrete interaction process.

Essentially, this means that to determine the status of a frame one has to

- map role specifications to individual agents or sets of agents and keep track of the degree to which they "match" the respective roles;
- keep track of trajectory completion with respect to (i) which paths have been pursued so far, (ii) which portion of them has already been observed, and (iii) what values the variables in the trajectory model have been bound to;
- store which of the context conditions are currently satisfied and to which degree the frame is activated;
- infer which of the beliefs described in the belief model are held by the interacting parties.

attribute	*description*
role status	An assignment of the roles in the role and relationship model to actual agents, groups, organisations etc. May involve tentative assignments or statements about uncertainty regarding "role matching". Additionally, assumptions in support of the current assignment may be supplied.
trajectory status	Encapsulates information about the current status of frame execution. This involves recording observed action and communication relating it to the trajectory model. This information may be enriched by statements about deviance from and adherence to trajectory expectations, and it can also give possible reasons for such deviance or adherence.
context status	Keeps track of beliefs about the applicability of enactment and activation conditions. May also include statements of doubt or uncertainty concerning the relevance of this frame with respect to the ongoing interaction or qualitative statements about "degree of activation" and explanations for these.
belief status	Information regarding the epistemic properties of the current enactment of the frame. May express, for example, uncertainty about the degree to which allegedly common knowledge is actually shared by all interactants or include assessments of the degree to which a lack of knowledge impedes frame execution or not.

Tab. 3.3: Contents of the status attribute of a frame

As the more detailed view of status data (table 3.3) suggests, frame status can capture much more than a simple mapping from a frame to the current situation, e.g.

- information about the uncertainty associated with this mapping,
- reasons that speak for or against the applicability of the frame, and
- an explicit, detailed representation of how actors deviate from the expectations induced by the frame.

Ultimately, the decision about which of these attributes to include and how to generate them depends on whether and how the framing agent will make use of them. This is also true of the other meta-level attributes, as they are all used for frame management.

It should be remarked that different status attributes are not independent from each other: the correctness of role assignment must be constantly underpinned by observing the actions of others and oneself; conversely, those actions must be interpreted with respect to the existing expectations associated with a role to classify as compliant or deviant actions, etc. This raises the interesting question of whether unexpected behaviour should be assigned to one's own faulty understanding of roles, deliberate deviance, erroneous interpretation, inconsistency in frame activation among the parties, etc. In fact, the identification of these reasons is one of the most important activities in framing, as will be discussed in section 3.4.

As an example, consider an agent who holds the link client role in the linkage brokering frame with trajectory as in figure 3.2 and who modifies the value of *ContrPrice* in his reply to the broker's $propose(lba, lc, contract(ContrPrice))$ message. This means that, strictly speaking, the observed message sequence does not match the trajectory model. However, a framing agent (e.g., the broker) should not overthrow this frame too easily, since the client's deviant behaviour may have been simply caused by ignorance, and the broker could make an attempt for "recovery" by informing him of the constraint. Yet this would only be reasonable if the broker has some reason to believe that the client is not aware of the constraint. For this fact, it is important to combine the status monitoring activities for the different frame slots.

Links

The *links* section contains meta-frame information that places the current frame in the context of other frames by specifying relationships between the frame in question and other frames. Two kinds of such inter-frame relationships can be distinguished:

1. *Structural relationships*are defined purely in terms of relationships between the attributes of the respective frames, and can be verified by means of syntactic comparison. They include (but are not limited to):
   - *Inheritance:* A frame F inherits a frame G if its roles and relationships, context and beliefs extend those of G in a conjunctive manner, and if the trajectory model of F admits a subset of the trajectories allowed by G.
   - *Aggregation:* A frame F contains a frame G if the roles, relationships, contexts and beliefs and admissible trajectories of F are a super-set of those of G in a disjunctive fashion.
   - *Coupling:* A frame F is sequentially/parallel coupled to G if their trajectories are executed in sequence/in parallel. Special cases of such relationships are "F resembles a sub-sequence/prefix/postfix of G" (in a sequential sense) or "F spawns G as

a sub-process". The relationships between the remaining (non-trajectory) descriptive attributes of the two frames are the same as in aggregation.

- *Similarity:* This is a whole class of relationships which denote that two frames have certain descriptive elements in common, e.g. that they share a (number of) role(s), a condition, some set of beliefs, a trajectory prefix, an identical number of trajectory loops, etc.

More specific structural links can be conceived of, such as frames that only inherit certain types of information from other frames, special kinds of coupling (e.g. the trajectory of F branches into G and H after the ith step), etc.

2. *Framing relationships,* i.e. "meaningful" relationships between frames that are relevant for the framing process of the agent who uses them. Unlike structural links, they cannot be verified by looking at the definitions of the frames in question, but rather stem from (frame-external) background assumptions the agent makes. Again, there are different kinds of such relationships, from which we list the most common ones:

   - *Alternative:* Expresses that a frame F is (not) applicable when another frame G is applicable, because it has roughly the same context and result, e.g. two frames for different types of auctions.
   - *Continuation:* An encounter in which F was enacted is likely to be continued with one that matches G. An example for this is a "first contact" frame between two business partners that is later followed by negotiations and formal agreement described by other frames.
   - *Resolution:* F can be used to resolve a problem that has occurred during the execution of G. Good examples are frames for argumentation, mediation through third parties, appeals to authorities or legal institutions, etc.
   - *Interleaving:* This is useful to link frames that are interleaved but not directly interconnected in terms of trajectories, for instance negotiations with different parties led concurrently but not directly aligned with each other because only a subset of the negotiating partners are aware of the concurrent execution.
   - *Modulation:* F is a modulation[4] of G if it is not possible to discriminate between them on the grounds of trajectory analysis, i.e. if one frame casts a different meaning on the trajectory of the other. For example, a negotiation frame could be enacted simply for the purpose of determining the other's valuations, without actually intending to make a deal.

Numerous extensions and variations are possible to the relationships sketched here. They can be given probabilistic semantics, they can be subject to additional logical constraints, they can be defined in terms of whole sets of frames rather than individual frames, etc.

The significance of such frame links is that they enable the agent to compare frames structurally and semantically to reason about them at frame (rather than single action) level – it is only through links that frames become subject to reification in reasoning. *Inter alia*, this allows for devising *framing rules* as constraints that deal with frames as first-class citizens. These would be included in the context of a frame to express, for example, that one frame must have been completed before another one is started.

[4] We use modulation as a super-term for the manipulations discussed at length by Goffman (1974), as mentioned in section 2.2.2 (see p. 25).

History

The next frame attribute we have to discuss is *history*. Histories are, in fact, nothing else but a specific kind of links between a frame and earlier versions of it that use past status attributes as arguments.

A frame history describes a sequence of past enactments of the present frames. This involves recording the experiences made with a frame, or, more precisely

- the status assigned to frame attributes in each encounter instance,
- the transformations to a frame that were induced by each new experience, and
- the operations on the frame through which these modifications were carried out.

So the idea is that the history of a frame is a sequence of concrete encounters in which it has been enacted together with a record of the modifications that were induced by these experiences.

A convenient format for storing such a history is a *history matrix*

$$\begin{pmatrix} F_0 & S_0 & Op_0 \\ F_1 & S_1 & Op_1 \\ F_2 & S_2 & Op_2 \\ F_3 & S_3 & Op_3 \\ \vdots & \vdots & \vdots \\ F_m & S_m & Op_m \end{pmatrix}$$

with operations Op_i, frame versions F_i (F_0 is the initially generated frame) and status variables S_i. Such a matrix has the following semantics: Out n encounters which have occurred under different variants of frame F_0, we store m (where $m \leq n$ because not every experience is considered equally important to be memorised). Thereby, Op_i was the operation that transformed F_i to F_{i+1} after the ith encounter and the status assigned to the attributes of F_i was S_i.

An issue that remains to be discussed concerns the nature of the operations Op_i that transform frames. Although the construction and modification of frames is one of the central topics of section 3.4, some general remarks can already be made at this point.

The most simple operation that can be conceived of after completion of an interaction that corresponds to a frame is to extend the history slot of that frame by the status of the frame during that interaction. This allows to reconstruct the past case and to take it into account in future reasoning. Another fairly simple operation is to memorise the utility obtained from an interaction or the goals it achieved by adding appropriate post-conditions to the frame context. For example, adding a post-condition

$$utilityGain(U)$$

to express that the difference between total utility before and after the interaction was U necessitates determining the numerical value of U in the status variables S_i and thus to explicitly store the private value of a frame in the form of different utility values $U_1, \ldots, U_m$.

More complex operations involve transforming the frame to increase its generalisation capabilities (if an interaction has been perceived that does not match the current frame definition but creation of a new frame is to be avoided), merging it with another frame, deleting certain specific attributes, etc.

Extension

As frames are maintained and modified by individuals, it is only natural that the information they contain falls into two categories: (i) *common attributes*, i.e. knowledge about the interaction that is assumed to be common knowledge among the interacting parties and (ii) *private attributes* which supplement the frame data structure with information that is private to the agent who is applying the frame. The *extension* attribute captures precisely this distinction by describing which of the frame attributes are common knowledge and which are only known to the agent who maintains the frame.

The reason why such an additional attribute is needed despite the fact that the knowledge states of agents could also be described in the beliefs slot is connected to a problem of infinite regression. If, for example, we wanted to express that an entire frame is common knowledge, i.e. that all interacting parties have access to all the information contained in the attributes of this frame, we would have to reify the frame as some object of the world. If, then, the statement about the frame being common knowledge *itself* is part of the beliefs slot of that frame, this would cause a serious problem when trying to list all the things the agents know.

Practically speaking, it suffices for the extension attribute to tag each element of the other (descriptive and meta-level) attributes with the keyword *common* to distinguish whether they are known by other agents or not. Trivially, all information that is present in the frame is known to the agent who maintains it. In addition to this, we may want to indicate that the view held by some agents differs from the private view of the frame maintainer.

In our link brokering scenario, for example, we might want to express that the link client *lc* does not know

1. how the *propose* message of the link seller *ls* comes about, i.e. that the broker agent *lba* advertises interest of a client, etc.

2. that *ls* may make a counter-offer to *lba* (by introducing a new price *MinPrice*) in the negotiation process, i.e. the client thinks that *MinPrice* an arbitrary price claimed by *ls*.

This can be done by adding the following *extension tags* (referring to the non-iterative trajectory of table 3.2):

$$common(\{lba, self\}, advertise(lba, ls, interestedClient(FlexPrice)), ?)$$
$$common(\{lba, self\}, inform(ls, lba, notInterested), ?)$$
$$common(\{lba, self\}, MinPrice \leq FlexPrice, \mathrm{true})$$

Statements of the form $common(G, B, B')$ express that all roles in group G commonly know B while those who are not in G believe that B' is the case instead.[5] To define what it means for a fact to be commonly known among a group of agents, we use the definition of Fagin et al. (1995), which states the following: For a group G, φ is common knowledge if everyone in G knows φ, everyone in G knows that everyone in G knows φ, and so on *ad infinitum*.

[5] Note the use of a special identifier *self* for the frame owner, who may not play any of the roles in the frame (yet it can be commonly known that he has knowledge of this frame), and the use of the symbol "?" to express that the frame owner does not know what non-members of the group believe "instead".

The first two items in the tag list above state that *lba* and *self* know that *lc* may not know anything about the first two messages of the sub-negotiation between *lba* and *ls* (alternatively, *lc* may believe that *ls* took the initiative for the whole deal, etc.) The third tag signifies that *lc* has knowledge of a trivial constraint (true) in the place of the actual counter-offer price constraint $MinPrice \leq FlexPrice$ in his beliefs. These tags may not only be applied to trajectories, but also to all other attributes.

The purpose of the extension attribute lies in the fact that it is essential for the manipulation of frames. This is because, if an agent wants to use a frame in a way that is different from the common interpretation (or at least the interpretation of its current adversary), he has to exploit the assumptions commonly held and modify his private knowledge. This amounts to *deceiving* others about the true meaning of the ongoing interaction, and this kind of deception which may play an important role in intelligent framing.

3.4 Framing in InFFrA

Framing is the process of employing frames in interaction situations. Although it is intuitively clear from the discussion of Goffman's theory for human actors (section 2.2.2), many questions arise when we attempt to develop a computational model of framing:

- How are frames selected in concrete situations? How does this process relate to the agent's private goals and motivations?
- How are new frames learned from observation and how are existing ones adapted to changing patterns of interactions? How can agents "invent" new frames to introduce new patterns of joint action?
- What are the concrete implications of frame selection for the agent's behaviour? When should he adhere to the expectations associated with a frame and when should he break them?

These questions suggest that framing is a very complex activity that involves (i) tracking the enactment of activated frames, (ii) choosing whether to retain the current frame or to change to another frame when appropriate, (iii) modifying frame knowledge with experience and (iv) relating these three activities to one's private goals in order to make them part of individual rational decision-making.

In the following paragraphs, we will first present a simplified overview of the computational model of framing employed in InFFrA. Then, we will lay out this model in full detail by describing the data structures it uses and the operations that are performed on them.

3.4.1 Overview

The top-level model of framing in InFFrA is a processing loop between perception and action which is actually pretty similar to the general agent processing cycle proposed in (Russell and Norvig 2003). However, it is specifically designed for agent perception, reasoning and action in *interaction* processes (as opposed to general agent operation). For this reason, InFFrA should be rather thought of as a *social reasoning architecture* that has to be combined with other, sub-social reasoning components to obtain a comprehensive agent architecture.

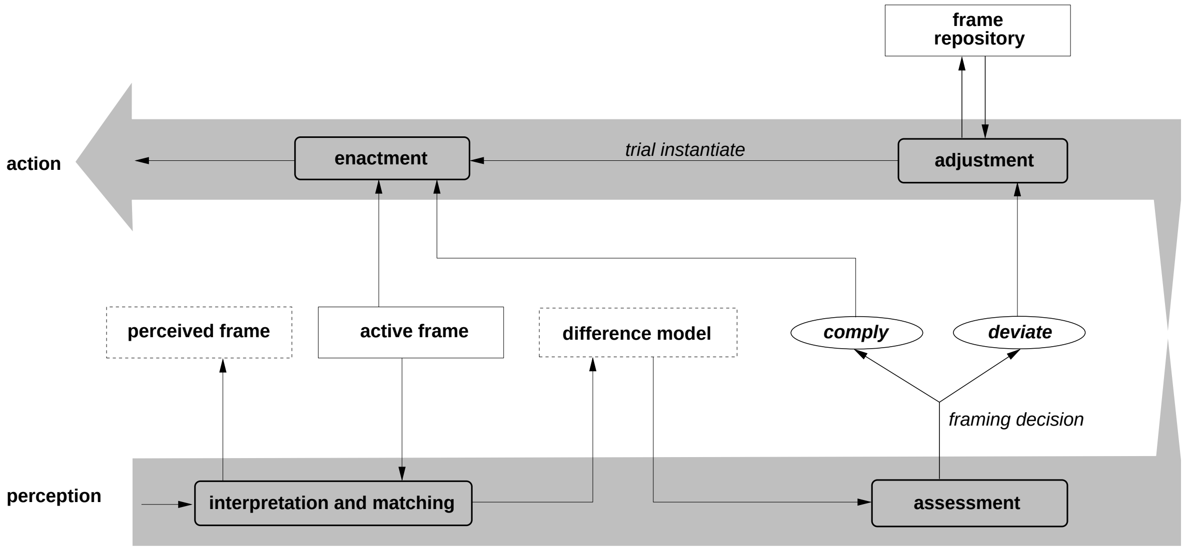


Fig. 3.7: Simplified overview of the InFFrA framing process

As can be seen from the overview of framing depicted in figure 3.7, the perceive-reason-act loop for frame-based social reasoning consists of four major stages:

1. Interpretation and matching: The agent processes incoming percepts, extracts up-to-date information about the interaction, and generates a description of the encounter, the so-called *perceived frame*. He matches this frame against the *active frame* which is supposed to provide the information required to appropriately participate in the encounter.

2. Assessment: The result of the matching procedure is a *difference model* that expresses which parts of the perceived are in accordance with the active frame, and which aspects of the encounter differ from the expectations associated with the active frame. This model is now used to assess whether the currently active frame is still appropriate, and a decision is made as to whether it should be maintained ("comply") or not ("deviate").

3. Adjustment: If the agent decides to *deviate* from the active frame, he obviously has to activate some other frame, i.e. to *re-frame*. Frame adjustment means either modifying the activated frame to better suit the interaction situation or retrieving some other frame from the *frame repository*. Candidate frames are *trial instantiated* as active frames iteratively until the resulting difference model is sufficiently weak so that assessment yields "comply".

4. Enactment: Whenever the agent decides to comply with the active frame, he can use that frame to derive prescriptive constraints for his own actions. These influence his action decisions (that may also be determined to a certain degree by sub-social reasoning) so that he can (continue to) play his part in the remainder of the encounter.

At this point, it becomes clear how this four-step process is an extension of Mead's four-stage model of the social act discussed in section 2.2.2. Impulse, perception, manipulation and consummation (cf. figure 2.1, page 23) now become interpretation and matching, assessment, adjustment and enactment, though not in a strict 1:1-correspondence. This is because employing frames as "chunks" of expectations rather than individual acts as in the Meadian model and combining this model with the agent perceive-reason-act loop necessitates some modifications.

A closer look at the relationship between framing and the Meadian model reveals the following correspondences and differences:

Impulse/Perception vs. Interpretation and Matching As in Mead's model, a perceived "disequilibrium" spawns a framing reasoning cycle. This can be caused by the emergence of a private goal that has to be achieved, an ongoing encounter that is being continued, etc. The InFFrA model has two advantages over the Meadian view: Firstly, the use of frames enables us to perform all relevant perception (in the sense of Mead) in parallel with impulse. This yields a much simpler model than generating the impulse and then having to make a choice regarding which percepts to pay attention to. In other words, the frame-based approach makes it possible to use generic frame templates as "perception filters", rather than having to search the space of all possible attention foci after perceiving an impulse. Secondly, generating an explicit difference model allows for predicting the consequences of mismatches at an early stage of the encounter. The agent does not have to wait for the next impulse to occur – he can plan ahead by considering the whole interaction process as a meaningful trajectory of action.

Manipulation vs. Assessment/Adjustment The manipulation stage in Mead's model seeks to eliminate the impulse by finding an appropriate reaction that is in keeping with the agent's own self-image and with his own and others' expectations. In terms of framing, this translates to finding an appropriate continuation of the perceived interaction sequence if the current active frame is no longer considered adequate. And this is exactly what the assessment and adjustment phases of the InFFrA framing process achieve. On the grounds of the difference model, which expresses to which degree the perceived frame diverges from the projected active frame, known frames are adapted or new ones created until some alternative frame is in accordance with the perceived situation. In that, the trial instantiation process maps to the "imaginative rehearsal" alluded to in Mead's model (see page 23). The agent "mock activates" different candidate frames internally until concordance between expected reaction and experienced reaction is achieved.

Consummation vs. Enactment This is the aspect of the computational model of framing that is probably most similar to the corresponding Meadian notion. At this stage, the divergence between perceived and projected behaviour has been eliminated by activating a new, suitable frame and leads to open action that is determined by the trajectory of the active frame. The inhibition of interaction is now overcome, and the activation of a new frame influences the self-image of the agent (by observing one's own behaviour in the next framing cycle), the image he has of others, the relevance of other actors and objects and the degree to which the initial impulse has been eliminated (for example, whether the alleged goal of the interaction has been attained).

This comparison nicely illustrates how InFFrA manages to overcome the deficiencies of the different sociological concepts used. On the one hand, it supplements the somewhat static, data-oriented concept of interaction frames with a dynamic, process-oriented reasoning model that largely draws upon Mead's model of the social act (see the remarks in section 2.2.3, p. 29). On the other hand, InFFrA frames have the capacity of integrating complex interaction knowledge so that the "action-to-action" level of the Meadian perspective can be transcended.

To see how precisely this combination is achieved, we will now discuss the data structures used in the framing process. After this, the framing stages sketched above will be further decomposed and laid out.

3.4.2 Data structures

As mentioned above, the core data structures used in InFFrA framing are the *perceived frame* (that records information about the currently ongoing interaction encounter), the *active frame* (the projected view of how the unfolding interaction will turn out), the *difference model* (that is used to assess the current framing decision), and the *frame repository* (a database of frames used in the overall social reasoning process).

Perceived frame

The perceived frame can be conceived of as a frame which, starting from virtually no information at the beginning of an interaction, is extended in each framing cycle with incoming observations about the interaction. It provides, above all, a *descriptive model* of the interaction "as is" and a facility to store relevant information about it so as to obtain a focused picture of the situative context of the encounter. Additionally, it influences the situation interpretation module by affecting the way perception is processed. For example, if the perceived frame already states that a particular set of actors are the exclusive participants of the current interaction, the behaviour of other agents is ignored until this interaction is over.

In comparison to other frames, the perceived frame exhibits some special properties that result from its special purpose:

- Initialisation: The perceived frame is initialised with a single message as the root of its trajectory model and with two concrete agents as roles, namely the sender and receiver of this message. No relationships, contexts or beliefs (let alone meta-level attributes) should be inferred before the first message is uttered or perceived for the received frame to remain unbiased and able to adjust to any type of interaction that may occur.
- Trajectory branching: The trajectory model of a perceived frame may contain "lanes" of interaction that are observed in parallel, but trajectory paths do not branch out as only one sequence (per concurrent path) can be observed per encounter.[6]
- Descriptive mentalism: A perceived frame is supposed to capture observations rather than assumptions of the modelling agent. Hence, the mentalistic aspects of a frame (deliberative role expectations, beliefs, activation conditions, extension tags) should only contain knowledge that can *directly* be inferred from what is uttered in communication (e.g. if an agent says "I know X" this may be included in the beliefs slot). In

[6] An exceptional case are agents who use branching to express uncertainty about what has been observed.

particular, knowledge that is in one's own (sub-social) knowledge base should not be used to infer mental states of others. This has the advantage that we do not assume other agents to operate on our own private knowledge which they may not share.

- No meta level: The perceived frame has no history, no links, no status and no extension. This is because (i) it is only perceived once and not maintained any longer after the encounter is finished, (ii) it does not stem from the repository (so it cannot have any relationships to other frames), (iii) it is not an abstraction of several concrete cases (it contains no abstract information for which we have to store an instantiation in the form of a status attribute), and (iv) it has no extension for the reasons of avoiding mentalistic assumptions just mentioned.

All in all, the perceived frame is a very simple frame that does not use the full modelling power of the InFFrA frame model. In many cases, it will simply consist of a sequence of perceived messages and a list of agent names that reflect which agents utter and observe the messages. Still, it is important to interpret the current interaction encounter in terms of a frame, as this allows for the application of a generic *frame comparison* procedure that can also be used in other framing steps.

Active frame

While the perceived frame provides a descriptive model of the interaction, the active frame delivers the *normative* model of the interaction as it should be according to the experience (or designer knowledge) stored in this frame.

As the source of information regarding the actions that have to be taken by the agent, the active frame is *the* central data structure of the framing process: No matter how the agent has reached a decision about which frame to activate (and how many modifications, trial instantiations etc. it has performed until that point), it will blindly "obey" the active frame once activated, at least for the imminent action. In accordance with the Meadian model of the social act, the active frame is extended with information about the current experience once applied, i.e. the agent records this recent experience as a new instance of the activated frame. Apart from that, the active frame is also used to obtain an assessment of the current framing situation through comparison to the perceived frame.

As with the perceived frame, its purpose imposes certain constraints on the properties of the active frame:

- Unless the framing agent is only observing an interaction he will not participate in, the status attribute of the frame must provide sufficient information for the agent to be able to derive the next action he should take. This means that (i) the enactment conditions required for (at least) the imminent action must currently be satisfied so that it is physically possible, (ii) the agent has assigned one of the roles of the frame to himself, (iii) choices for the agent which result from branching points in the trajectory model must be reduced to a single option.

- After an encounter is finished, the history slot of the active frame should be extended by a new item. Even in the most simple implementation, this involves recording the status of variables of the trajectory and role models, so that at least the concrete message sequence of the encounter can be re-constructed. If this is not possible for reasons of limited space, some kind of simplification has to be performed, to avoid storing all instances (e.g. generalisation of individual status instances or simply ignoring certain cases).

Henceforth, we will also assume that the active frame always stems from the frame repository, i.e. that it has already been stored there, even if it just an *ad hoc* modification of an existing frame or an entirely new frame. This helps keep the reasoning cycle simple: If all frame adaptations and constructions are performed in the adjustment phase and the results are directly stored in the repository, there is no need to store the frame after an encounter. Also, this implies that no provisions need to be made regarding unexpected termination of the encounter (where no additional framing cycle may be entered because no new messages are observed).

Difference model

Defining a representation for the difference model is one of the aspects of InFFrA that leaves a lot of freedom to the designer. The reason for this is that the importance of differences between perceived frame and active frame may vary between applications, and this depends on the way framing assessment (see section 3.4.3 below) is performed.

In an organisational setting, for example, only the owners of certain roles that are formally defined (e.g. project manager, line manager, CEO) can participate in certain interactions (e.g. an executive board meeting). This supplies framing agents with strict criteria as to whether the current situation matches one of these heavily role-based frames. As a consequence, the difference model definition should emphasise the importance of role compliance or deviance in such a setting. Quite contrarily, the subtleties of trajectory path selection may be much more important than roles in a more informal setting such as a personal argument between two agents.

In any case, the difference model should

- provide information concerning those aspects of the perceived frame that *conform* with the expectations of the active frame and those that *deviate* from them,
- distinguish between differences in all descriptive slots of the two frames so that the representation of the difference model itself is "semantically close" to that of a frame (ideally, the whole framing procedure would only need to deal with a single kind of data structures, i.e. frames),
- ignore meta-level attributes other than history (as the perceived frame does not dispose of any such meta-level information).

Frame repository

The frame repository is an up-to-date collection of frames that the framing agent has at his disposal. It is used to retrieve candidates for activation and to store frames that have been activated in the past. In addition to the frames the agent has created and adapted himself, the repository may be initialised with a set of pre-designed frames (in very simple implementations, the repository may even be entirely hard-coded and immutable).

Apart from being organised as a database that allows for efficient update and retrieval, the repository is characterised by the following features:

- It contains past active frames, where a distinction should be made as to whether these frames were only used temporarily during an encounter and frames that were successfully completed. Some implementations may even require that only successfully completed frames are stored in the long run.[7]
- The size of the repository should be bounded to prevent overly complex or time-consuming retrieval and update operations. As the experience of the agent constantly increases with new interactions, appropriate methods need to be applied to generalise from individual instances of identical frames to (i) avoid redundancies that increase the computational burden of frame selection and (ii) ensure that the agent's social reasoning component responds in a timely fashion.
- Extensive use should be made of the structural relationships and framing relationships contained in the link attributes of stored frames when accessing frames during re-framing. The more expressive these links are, the more will the search space be reduced in finding suitable alternative frames when the currently active is no longer considered appropriate.

It should be remarked that since we are mainly interested in "face-to-face", micro-level interactions, the responsiveness of an InFFrA reasoner is of particular importance, because such interactions have fairly strict temporal restrictions. Most agents that engage in a conversation will not wait forever for a response from other parties but simply end the encounter after some time. Therefore is is essential that the repository is well-organised and suitably structured to allow for efficient frame selection (especially because additional time is needed to trial instantiate candidate frames).

3.4.3 Functional components

As described in section 3.4.1, the InFFrA framing cycle consists of four stages: situation interpretation and matching, assessment, adjustment and enactment. Figure 3.8 shows a further functional decomposition of these steps that extends the process overview of figure 3.7. It shows in more detail how the different functional modules operate on the data structures introduced above, and how their ensemble caters for an integrated framing functionality.

Situation interpretation

As shown in table 3.4, situation interpretation obtains percepts directly from the agent's environment, but also information about the agent's private goals, valuations and preferences. The reason for this is that, as far as the InFFrA reasoning layer is concerned, these things also form part of its environment.

Percept data does not come unfiltered, except when no interaction encounter is running; as soon as an encounter has been initiated and a perceived frame has been generated, this perceived frame determines which aspects of perception are relevant and instructs the situation interpretation module to ignore all other aspects.

[7] Of course, during an encounter, the agent should still be allowed to "experiment" with temporary frames whose appropriateness has not been verified by previous successful application.

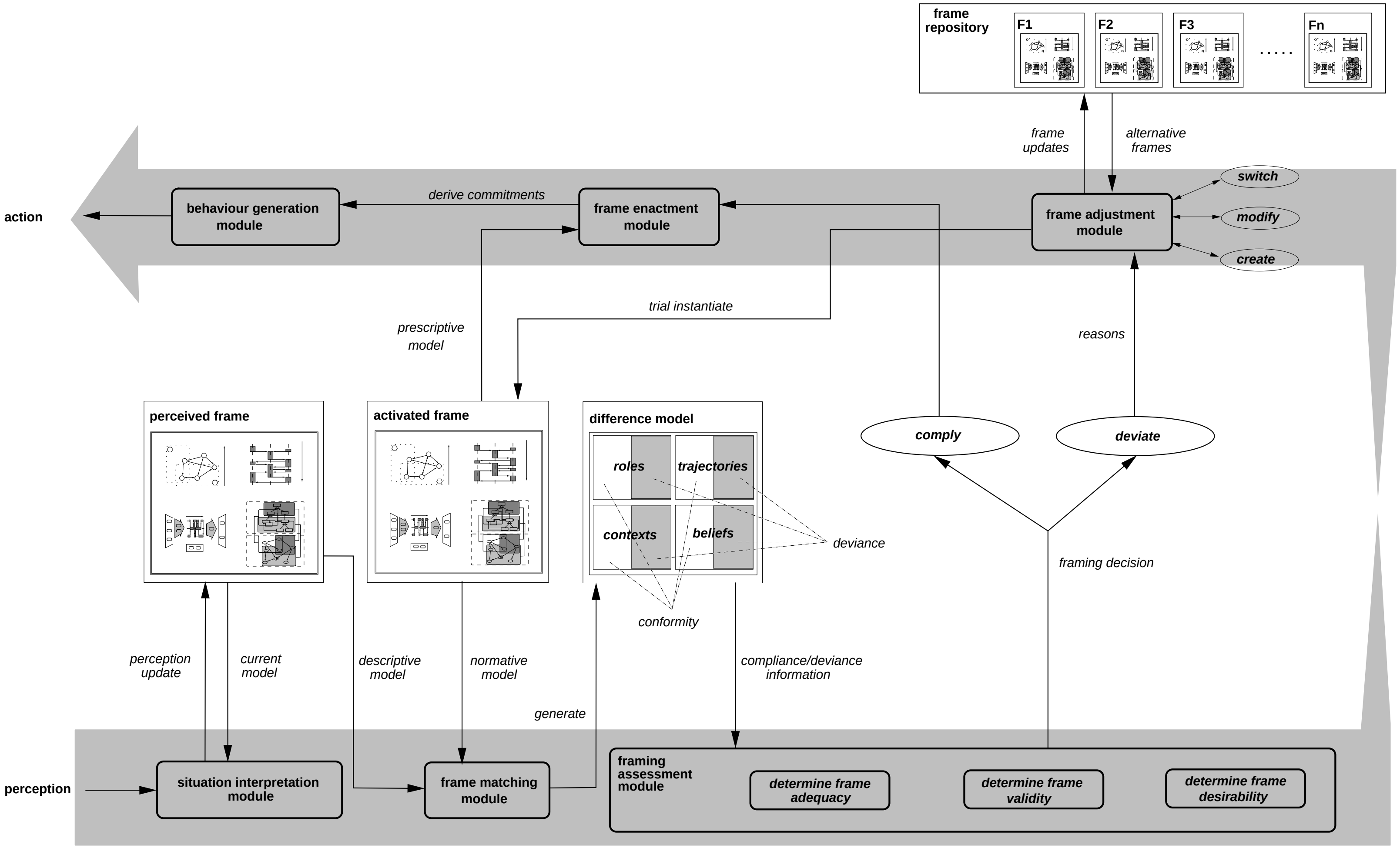


Fig. 3.8: Detailed view of framing in InFFrA

inputs	*processing*	*outputs*
percepts, private beliefs, goals and preferences, situation focus from perceived frame	selects relevant percepts for perceived frame, initiates and terminates encounters	perceived frame generation and update, perceived frame forwarded to matching module

Tab. 3.4: Situation interpretation module overview

This does not mean, though, that the perceived frame has control over the situation interpretation module. Quite the opposite is the case: Since it is the task of this module to generate and update the perceived frame, it can commit itself indirectly to its own future focus of attention by identifying which aspects of the perceived situation go into the perceived frame. That way, the computational load of processing percepts is reduced in future framing iterations (ideally, the range of relevant percepts should decrease in each consecutive step of an interaction).

More concretely, the situation interpretation module handles the following operations:

- If (sub-social) goals are perceived that require joint action (or at least some participation of other agents), a new perceived frame is generated, indicating that an encounter has started. Forwarding the perceived frame to the frame matching module triggers operation of the other framing components.
- Likewise, a new perceived frame is created if an (unrequested) message from some other agent is perceived that spawns a new encounter.[8]
- According to the underlying (descriptive) model of the current interaction situation, the situation interpretation module identifies which messages and belief items are relevant to the ongoing interaction and updates the perceived frame accordingly. In particular, it appends new messages to the trajectory model of the perceived frame.
- If no further messages or actions are perceived within the maximal time-span estimated for an encounter, the module terminates operation of the InFFrA layer. The timeout value may be a global InFFrA parameter or be determined by the module itself during operation (e.g. according to the estimated time it will take a peer to reply depending on the complexity of a query).

Frame matching

The frame matching module is responsible for computing the difference model that expresses in which respects the perceived frame adheres to the normative expectations that result from the active frame and where the current interaction process deviates from these expectations.

During mock activation, the same comparison is performed for trial frames that are only activated temporarily until one of them is chosen for "real" activation. Using the

[8] Note that this does not imply that the agent will react to this message by own utterances or actions. It only means that the social reasoning layer of the agent is activated.

same procedure for a frame that has already been activated and for trial frames that are experimented with not only simplifies the specification of the entire framing model. It also reflects the view that all frames should be treated the same since they are all potential candidates when searching for the best frame. As suggested by table 3.5, operation of the

inputs	***processing***	***outputs***
perceived frame, active/trial frame, activation from situation interpretation model	if activated, compares active/trial frame and perceived frame; generates or updates difference model	difference model, information about type (active/trial) of matched frame

Tab. 3.5: Frame matching module overview

frame matching module is triggered by obtaining the perceived model from the situation interpretation module. If activated, the matching module performs the following steps:

1. Inspect the currently active frame.
2. Perform the actual matching procedure to obtain a difference model.
   (a) If the active frame was trial instantiated, record that it has already been checked.
   (b) Else (if it was the actually activated frame, that is), this information should be included in the difference model.
3. Forward the resulting difference model to the framing assessment module.

Information about whether the active frame is a “real” frame or just a trial frame is required by the assessment module: If it is a truly activated frame, the frame assessment module needs to know that a re-framing process will be initiated if it decides that the active frame is not appropriate. Else, the assessment module should know that the trial frame need not be considered more than once during the same trial instantiation (as it has already been tried out) and also forward this information to the frame adjustment module.

If the matching module obtains no new perceived frame, the active frame has already been stored in the repository before, and nothing needs to be done.

Framing assessment

Together with the frame adjustment module, the framing assessment module embodies the core of the *learning* aspects of framing. In machine learning terminology (Mitchell 1997), the active frame represents the current *learning hypothesis* that is supposed to “solve” the interaction problem. The perceived frame, on the other hand, is the *training sample* that is being processed in a particular situation. The matching module provides difference model information, so that the assessment module can function as a *critic* that evaluates the current hypothesis and decides whether it should be retained or overthrown.

The assessment process is summarised in table 3.6: The module obtains the difference model from the matching module together with information as to whether the frame

inputs	*processing*	*outputs*
difference model, type of active frame	evaluates difference model and determines adequacy, validity and desirability of current active frame	framing decision, reasons for decision, type of active frame

Tab. 3.6: Framing assessment module overview

under analysis has been activated or whether just mock-activated. Then, it analyses the difference model in three respects:

1. *Frame adequacy:* The active frame has to be executable in a physical sense. The enactment conditions that have to hold for the actions contained in the remaining trajectory (for all parties) have to be either (i) already satisfied according to the agent's beliefs (relevant beliefs have been stored in the perceived frame and included in the difference model if they concern enactment) or (ii) precipitated by trajectory actions that will occur before the respective conditions are required to hold. Also, the actions need to be executable with respect to specificity, i.e. they need to be concrete actions and not just abstract templates or variables that cannot be executed as such.

2. *Frame validity:* The active frame should adequately capture the interaction perceived so far. Most importantly, as the behaviour of participating actors is supposed to be indicative of the type of interaction that is unfolding, the actions performed so far should match the normative trajectory model of the active frame. Roles and relationships, (de-)activation conditions and beliefs can be used to obtain additional information about the validity of the active frame.

3. *Frame desirability:* Even if the active frame is executable and representative of the perceived encounter, it may not be desirable for the agent. For example, the cost incurred by the remaining trajectory actions may be too high, or they may not achieve any goal that is relevant for the agent. In that case, even though frame activation is "correct", it is unreasonable for the agent to stick to the current frame. However, the social cost of deviating from the current frame may be higher than the private loss, so that the agent should comply with the frame in some situations.

Clearly, this description of the framing assessment modules leaves a lot of issues unresolved. This is quite deliberate as different applications may require different assessment methods.

In particular, different variations are possible regarding the ordering and the strictness of the three assessment sub-procedures, and these largely determine the framing behaviour of the InFFrA agent. A very cautious agent will first check for complete physical adequacy right at the beginning of an encounter, while a more risk-seeking agent might only be concerned with the executability of imminent actions and "hope" that conditions that are only relevant later will somehow be brought about before they are needed. A "conforming" agent will be more concerned with adhering to existing practice than with satisfying its own needs in every situation. This might lead to laying more weight on validity rather than desirability. Also, the combination of these aspects may be adapted flexibly during trial instantiation or during consecutive framing cycles, depending on how many choices are still available. For example, a frame that is risky utility-wise may be more easily accepted during early phases of an interaction, because it is hoped that additional information will help to better estimate this risk. Likewise, a risky frame will be accepted when the agent is left with less choices after repeated trial instantiation, because there is no(t much) alternative.

In any case, the result of evaluating the difference model is a decision as to whether the agent should *comply* with the active frame or *deviate* from it. Compliance means that the agent may directly proceed with frame enactment, i.e. with deriving its own behaviour from the current frame.

Deviance, on the other hand, implies a change of frame called *re-framing*, and spawns a (series of) trial instantiation(s) in which frames from the repository are adapted and mock-activated by the frame adjustment module until a suitable frame is found. For this purpose, the assessment module should supply the adjustment module with as much information about the *reasons* for the framing decision as is available. This involves indicating whether re-framing is performed in the context of trial instantiation or whether it means dismissing a frame that has already been used to generate agent behaviour in previous cycles, but also providing information that will guide the search process of the adjustment stage (a simple yes/no decision hardly gives any hints as to which other frames are suitable candidates).

For example, if the reason for deviance is a validity failure, frames with different trajectories are good candidates. If, on the other hand, low desirability is the reason, this suggests that a frame is needed which has a similar trajectory model but also provides a higher utility for the agent.

Frame adjustment

The frame adjustment module proceeds as shown in table 3.7. It processes information

inputs	***processing***	***outputs***
reasons for re-framing, activation/trial instantiation status	retrieves frames from the repository, modifies existing frames and generates new frames; determines candidate frames	trial frame

Tab. 3.7: Frame adjustment module overview

about the reasons for re-framing generated by the assessment module, searches the frame repository (which serves as a *hypothesis space*) for suitable alternatives and outputs a trial frame, with which the active frame is temporarily instantiated for further evaluation. If existing frames do not provide a model for the perceived interaction that seems likely to pass the assessment phase, the adjustment model may

- modify either the cancelled frame or a frame from the repository,
- generate a new frame (from scratch or by combining existing repository frames),

- output a frame that was previously activated with an indication that it *has* to be continued because there is no alternative, or
- truncate the dismissed frame trajectory so that the interaction will simply end after the current step, if no other solution seems viable.

As mentioned before, the status of the trial instantiation process is quite important for the adjustment phase. This is because infinite or extremely time-consuming trials have to be avoided, but also because different criteria apply in different re-framing situations. As an example, consider an interaction that is almost finished; there, it may not be advisable to search for new frames, because the expected gain of using a better frame is rather low. Or, if very general candidate frames have already been tried out and failed, it is very unlikely that more specific ones will be suitable.

Again, various approaches to implementing an adjustment module are possible, and the learning algorithms and heuristics employed for frame adjustment also bear implications on how the frame repository should be organised. If, for example, case-based methods are used which involve little frame adaptation and heavily rely on retrieval and combination of earlier cases, the efficiency of retrieval and the definition of operators for combining "nearest neighbours" is very important. Quite contrarily, elaborate rule-based methods for creating new frames by using domain-specific information and existing frame conceptions will necessitate stronger semantic links between frames in the repository.

Finally, the importance of meta-level frame attributes should not be under-estimated in the context of frame adjustment. History information can be exploited to re-construct previous versions of a frame, so that wrong modifications can be undone (if, for instance, a frame modification was attempted that did not work out), and links may directly suggest alternatives by narrowing down the search space *prior to* further costly trial instantiation. For example, explicit framing rules can be implemented that use frame links as pre-conditions that govern re-framing behaviour (cf. the remarks on p. 59).

Frame enactment

When a frame has been activated that is judged adequate, valid and desirable by the assessment module, frame enactment comes into play, i.e. deriving constraints for the framing agent's behaviour from the current frame (see table 3.8).

inputs	***processing***	***outputs***
active frame, comply decision from assessment module	derives social constraints for agent behaviour from the frame trajectory	commitments for behaviour generation

Tab. 3.8: Frame enactment module overview

The process of enactment involves:

- Identifying the existence of (de-)activation conditions to start or end an encounter. Note that "ending" an encounter here only means that no further action is taken; if, however, further messages are perceived that suggest a continuation, the encounter may be carried on, and therefore it is the situation interpretation module which actually decides on encounter termination.

- Inspecting the trajectory model of the active frame that is now used as a *prescriptive model* to determine the next action(s) that have to be taken from a social reasoning perspective. This may require choosing a specific option if the trajectory model allows for several responses or if it contains non-ground action/message templates that require instantiation to be executed.

It is important to understand that frame enactment does not imply that the suggested actions will actually be performed. The module merely outputs social commitments (in the sense of "commitments that are derived from applying conceptions of interaction patterns") to the behaviour generation module as the result of the agent's interaction-level reasoning process.

Behaviour generation

As shown in table 3.9, the behaviour generation module obtains social commitments at the level of concrete actions from the frame enactment module and spawns their execution if they can be reconciled with the decisions made by sub-social components (e.g. a BDI reasoner). Its design may range from very simple, where it merely forwards the action

inputs	***processing***	***outputs***
social commitments	reconcile frame-governed decisions with sub-social action decisions or goals	resulting actions (if any)

Tab. 3.9: Behaviour generation module overview

prescribed by the InFFrA layer to the local reasoning process of the agent and leaves it up to other reasoning layers to decide whether the action will actually be executed (or simply overrides all sub-social decision-making) to very complex, for instance if it seeks to actively balance social commitments with local goals through explicit reasoning.

The latter solution, however, requires that all necessary information about local goals is forwarded from the active frame via the frame enactment module (since the behaviour generation module is not linked to any other module). This is rather counter-intuitive, as it would force the framing process to mix knowledge about agent preferences and goals with (frame-based) social expectations.

Yet, it may be the case that the designer of the social reasoning component cannot modify the sub-social reasoning components of the agent and see to the balancing of social and individual rationality there. In this case, there is no alternative to implementing a more complex behaviour generation process endowed with this functionality.

3.5 Discussion

As InFFrA is an abstract social reasoning architecture, it raises quite some questions all of which are related to aspects of potential implementations that have been left unspecified.

Since it is supposed to provide a general schema for devising such implementations, these questions implicitly define the steps the agent architect has to take to derive a concrete design for socially intelligent, InFFrA-based agents. We shall briefly discuss some of these issues in the following paragraphs.

Frames and framing From the above description of InFFrA, it is clear that the framing process must be adapted to the frames used. In that, the following principles should be followed:

- *Frames should only include information that is used during framing.*
  Unless frame data is used for situation interpretation, matching, assessment, adjustment or enactment, it need not be captured in frames. Conversely, the design of the framing modules should make use of all information that is made available by the frames.

- *The complexity of framing and frames should be adapted to the "social flexibility" of agents.*
  If the application domain suggests strict adherence to fairly simple interaction procedures, there is no need to make things difficult by overloading the agents with unnecessary reasoning complexity. Likewise, if the modalities of desirable social interaction are largely under-specified in the overall system, intelligent framing may help develop novel, emergent forms of interaction that may lead to improved coordination.

- *The design of* InFFrA *data structures largely determines the effectiveness of framing.*
  The control flow of InFFrA framing is fairly strict, and it requires that individual components interact only with few data structures. Thus, the definition of interfaces between the components and of the information that is made available between components plays a decisive role for the performance of the InFFrA layer.

Interaction with agent-level action Because InFFrA is not a self-contained agent architecture, the designer must specify in which ways it interacts with other agent-level processes such as local planning and reasoning, but also with perception and action. This involves:

- Specifying when agents should interact, in particular, when they should initiate interaction processes and how they should combine these processes with non-interactive action

- Defining the "social stance" an agent assumes and what effects this is supposed to have at a global scale. Should agents rely on established procedures and conform with them, or should they try out new modes of interaction? How much should private utility matter to them compared to social welfare?

- Clarifying how InFFrA can be integrated in the global agent-level control flow. When and how is perception propagated to the social reasoning level? How are concurrent interactions with different interaction partners managed? How is InFFrA processing interleaved with other reasoning processes, such as planning, belief revision, deliberation, etc.?

- Explaining how frame learning relates to other learning activities. If new concepts are learned, does this enrich the modelling possibilities for frames, will the agent redesign his frames? Does the agent seek to improve the framing process itself over time, i.e. is it "learning to frame" by modifying his own InFFrA components? If so, what quality measures can the agent apply to an existing framing procedure to make learning decisions?

Control flow issues Even if InFFrA lays out a basic framing control flow, many issues require further clarification in concrete implementations. Some of these are:

- What is an adequate size limit for the repository? When should frames be "forgotten", and how should these frames be identified? Should the repository continually attempt to find new combinations of existing frames (when idle?) and what are the operators that should be used for this purpose?

- At which level of abstraction should frames be stored? On the one hand, abstract frames help keep the repository small and they reduce the number of potentially necessary trial instantiations. On the other hand, they will become computationally heavy if they include long histories, and will be difficult to process in the enactment phase.

- How many trial instantiations should be allowed per framing cycle? How much information about these should be stored in the candidate frames, and how much importance should be given, for example, to failed instantiations when assessing the overall usefulness of a frame?

These questions illustrate that InFFrA leaves plenty of room for more specific decisions. While this is a general problem with abstract computational architectures, it also shows in how many different ways the frame-based approach can be used.

3.6 Summary

In this chapter, we showed how an abstract social reasoning architecture can be developed that builds on the principles of interaction frames and framing. After some intuitive examples and a list of desiderata, we presented an overview of the InFFrA architecture with a particular focus on the way it integrates Goffman's frames with Mead's model of social action. Then, the elements of computational interaction frames and framing in InFFrA were laid out one by one and in great detail.

Thereby, three distinctive properties of InFFrA became clear that deserve being emphasised once more:

- The fact that the different frame attributes are utilised by different functional components of the framing process model,

- the "imaginative rehearsal" that is performed through trial instantiation during framing, and its relationship to machine learning, and

- the integration of socially intelligent behaviour with local, goal-oriented reasoning.

The major contribution of InFFrA is that it is entirely based on interaction frames as the building blocks for social behaviour, and that, unlike other architectures, it focuses on the learning and application of interaction processes that are abstracted into classes of interactions represented by frames.

As mentioned in the introductory chapter, InFFrA explores the whole range of possibilities for modelling frames and framing activities, thus providing the “big picture” of what we obtain when applying the respective sociological concepts to artificial agents. This implies that any concrete architecture that is based on the methods just presented will have to make specific choices regarding the aspects that are to be used in a particular instance of InFFrA, and on how these should be formally modelled. In the following chapter, we present one such concrete, formalised model that turns the theoretical principles of InFFrA into practice.

4. A Formal Model of InFFrA

An abstract architecture like InFFrA is useful for the conceptual design of socially intelligent agents that employ interaction frames and framing. For purposes of devising and implementing concrete computational agents that embody this functionality, however, such a conceptual framework is not sufficient – a concrete computational model is required.

To fill this gap, we will now introduce a formal model of InFFrA called m^2InFFrA. It is a simple but powerful instance of the family of agent designs represented by InFFrA with the distinctive feature of a probabilistic, empirical semantics. Its theoretical foundation is a novel model of agent communication that is in accordance with the assumptions put forward in section 2.2.3.

This chapter is structured as follows: We start by introducing the new view of communication that we propose. After that, the largest part of the chapter is devoted to the formal definition of m^2InFFrA and its discussion. We round up with a short summary.

4.1 A Theory of Communication

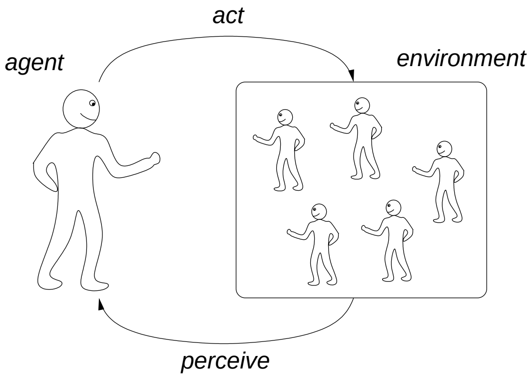
To understand how m^2InFFrA agents use frames for strategic communication, a formal semantics has to be developed that expresses the *meaning* that frames have for agents. Looking back at the assumptions made in section 2.2.3, where we stated that social reasoning should be based on *communicative expectations* and argued for the necessity of communication semantics to be *empirical*, *constructivist* and *consequentialist*, it is clear that a theoretical model of communication is must be used as a foundation for frame semantics that complies with these requirements.

To explain our view of inter-agent communication, let us step back and rethink what its underlying principles are in the context of interaction between *autonomous* agents in *open* multiagent systems (see section 2.1.3) if we assume an interactionist stance (cf. section 2.2.2) and if we consider the relationship between the frame-based approach and traditional views of agent communication discussed in section 2.3.2.

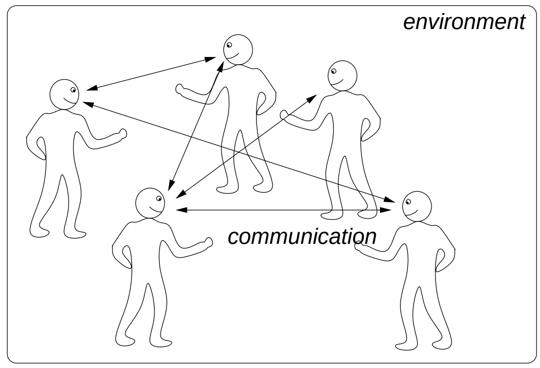
4.1.1 Communication systems

Deliberative agency implies that agents maintain a model of the world and manipulate symbolic representations of their knowledge to make appropriate decisions so as to achieve rational, goal-directed behaviour. According to the traditional view of agent-based and multiagent systems, this is usually done by modelling an environment populated by other agents.

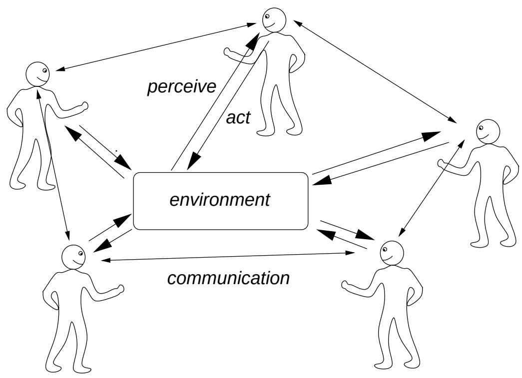
Quite contrary to this view, we can also look at multiagent systems as *communication systems* (CSs) (Nickles and Rovatsos 2004) that are mainly characterised by communicative



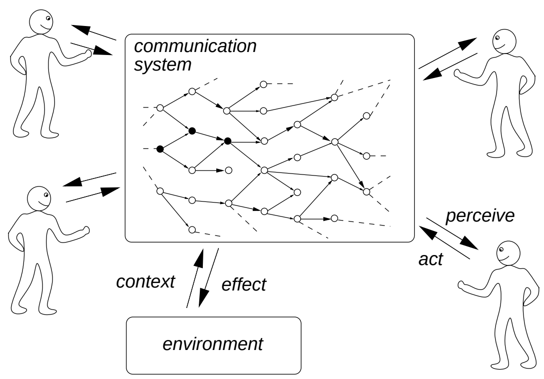
(a) *Single agent view:* the agent interacts with an environment that contains other agents.



(b) *Situated multiagent view*: the agent interacts with other agents inside a common environment.



(c) *Open multiagent view:* the agent interacts with an environment and with other agents.



(d) *Communication systems view:* agents interact with communication structures. The physical environment provides the context and is affected by physical actions.

Fig. 4.1: Different views of a multiagent system. In contrast to the traditional views (a), (b) and (c), the CS view in (d) focuses on communication processes and regards agents and physical environment as factors that contribute to communication.

events and the relationships between them. As shown in figure 4.1, agents contribute to the evolution of communication structures in a system through their own communicative actions and can inspect these structures to reason about the CS. According to this view, the physical environment is external to the communicative process, as are the agents. Both become part of the "environment" of the CS, which is now seen as the core component of the MAS. Of course, agents' cognitive states and the state of the physical environment affect which utterances will be performed by agents and hence the evolution of the global communicative process.

At this point, we shall not go into the details of the theory of CSs based on sociological systems theory (Luhmann 1995) for which we have developed a formal framework in (Nickles and Rovatsos 2004)[1]. What is important for our purposes is that when it comes to

[1] Section 7.2 contains a more detailed discussion of the relationship between InFFrA and the communica-

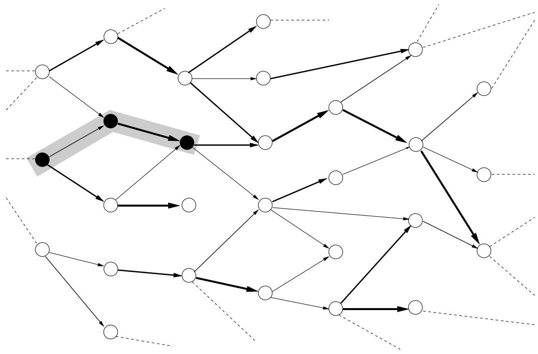


Fig. 4.2: Expectation network: Nodes represent communicative actions, edges correlations between them (variable line width is used to indicate different degrees of correlation). The shaded node sequence is used to describe the recently observed portion of the network, e.g. an ongoing conversation.

rational reasoning and decision making, deliberative agents can use models of *interaction* structures in pretty much the same way as they would use representations of the physical world. In other words, CSs can serve as a means of constructing and strategically using *communicative expectations* in the sense of our assumptions (section 2.2.3).

4.1.2 Expectation networks

For CSs to be constructed from observation and used for prediction, the communicative processes they describe have to be represented in some way, and we have to explain how this representation is processed by the observing entity.

To highlight the central aspects of our communication theory, it is convenient to imagine that the communication process being modelled as an *expectation network* (EN) (Lorentzen and Nickles 2001, Nickles and Lorentzen 2003), i.e. a graph in which nodes represent communicative actions and edges represent correlations between occurrences of different such actions (possibly weighted with probabilities). Figure 4.2 shows an example of such an EN.

If a network reflects regularities in observed interaction experience, the statistical correlations between subsequent actions of interacting parties can be seen as an approximation of the causal relationships in communicative behaviour (cf. p. 27). According to the consequentialist view of communication, we can use an existing network to “calculate” the meaning of utterances in terms of their predicted consequences under the assumption that past regularities will be repeated in the future. Informally speaking, this can be done by tracing the current sequence of messages in the network and computing the most probable actions that are expected to occur within a certain temporal scope starting from the present situation. As shown in the example of figure 4.3, the further predicted events lie ahead, the less accurate (and less relevant) will the prediction be. After the next communicative action has been observed, weights are updated, and the prediction starts anew from this point. Networks of this kind are fairly generic representations of expectations, yet they have only very limited expressiveness, since the expectation structure is reduced to a

tion systems approach.

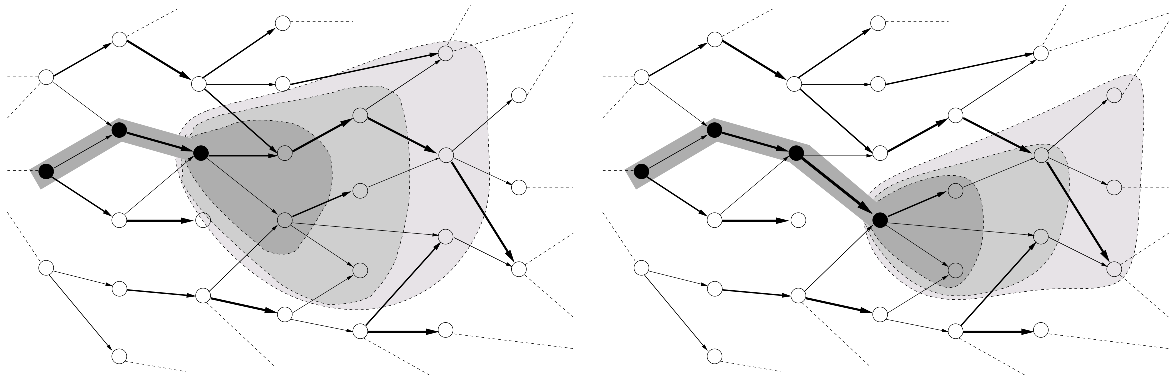


Fig. 4.3: Semantics and their evolution in an EN: the decreasingly dark shaded regions of predicted future actions denote that predictions regarding "distant" events are increasingly vague. The transition from the situation shown on the left to that on the right occurs upon observation of a new action that is appended to the currently relevant path. With this new observation, the correlation between the message previously observed and the current message increases as compared to alternatives that did not occur.

system of discrete events. Communicative actions are treated as simple "signalling" events. However, in the absence of any prior knowledge, they can still be used for describing the semantics of agent communication, at least in principle – after countless observations, the resulting expectations would probably adequately reflect the meaning of communication in a society.

4.1.3 Building expectation networks

We stated above that CSs depend both on the formalism used to describe regularities in communication and on the ways these models of communication are processed. Even the very simple model of ENs sketched above raises a number of issues, and in fact the answers to these questions can be used to characterise particular CS definition:

1. *Which events count as communicative actions?*
2. *How do we decide which communicative actions are interrelated?*
3. *What makes two communicative actions distinct for the CS?*
4. *How can degrees of expectation be derived?*
5. *What is the scope of prediction regarding future communication?*

As graph-based (probabilistic) models can be used to describe arbitrary discrete stochastic processes, question 1 needs to be answered to restrict the scope of observations added to the network to what counts as "communication". In accordance with interactionist theories (see section 2.2.2) and the assumptions made in section 2.2.3, we adopt the following definition:

> Any action executed by an agent that can be observed by others is considered *communication* if the agent can be expected (i) to know that other agents are observing the action and (ii) to consider others' potential reactions to this action before executing it.

Thus, physical actions class as communication in the same way as messages in the stricter sense if their execution can be perceived by others and if the agent who is performing them knows they are being observed by others.[2] The reason for this interpretation is that such physical actions are relevant to the flow of interaction if they are consciously executed in full awareness of the expectations of oneself and those of others. For simplicity, we will employ the term "messages" for all communicative actions in the following whenever there is no need to distinguish between physical and non-physical communicative actions.

Question 2 is essential for determining where to insert a new observation in the EN, i.e. when to regard a message as a *continuation* of another message. As we are only concerned with dialogues between "virtually co-present" actors, we will apply a temporal and pseudo-spatial criterion to determine continuation events. In other words, we will assume that agents meet in so-called *encounters* whose beginning and termination can be safely determined, and that they take turns in uttering messages. Also, we will assume that agents can unambiguously decide which agents are involved (actively or passively) in the encounter, and agents will assume an encounter to terminate whenever they do not receive any further messages within a certain amount of time after the last message.

Note, however, that this is not the only possible way to look at continuation. Often, temporal and spatial aspects do not matter at all, and other criteria are used to detect continuation. Subsequent letters exchanged between persons across different countries or continents are rarely not considered responses to the respective previous message, unless a very long time lies between them – in this case, continuation is determined by sender and recipient rather than by spatial or temporal proximity. A televised feature can be the continuation of some other broadcast produced by a different channel by virtue of referring to the earlier broadcast, even if they are directed to different audiences and the people involved in producing them do not even know of each other's existence (in terms of real persons). Here, reference, common subject and shared medium are much more relevant than time, space and people involved.

As concerns making distinctions between different messages (question 3), this is mainly a matter of applying a suitable generalisation mechanism in the construction of ENs. The above examples seem to suggest that any new symbol used in communication is inserted into the network as a new node. However, this is only the most trivial method that can be thought of and has two major disadvantages:

1. The set of particular symbols (network nodes) becomes huge and cannot be efficiently stored and re-used by agents, and

2. a manageable global system of communication symbols cannot be established because symbols are not repeated among different agents for their meaning to spread.

Therefore, it is useful to generalise over different symbols and to introduce new nodes only for messages that "make a difference" with respect to the way in which the CS is going to be

[2] Conversely, messages (in the sense of symbol exchange) have physical effects (generation of sound-waves, placing ink on paper etc.), but these can mostly be considered irrelevant to the achievement of agents' goals.

used. For example, the same message uttered by different agents might not require making that distinction because its effects are more or less the same across the entire society. Also, messages with slightly different content may be generalised to the same EN node if no difference can be discerned in the reactions of others to these messages.

The importance of what kinds of abstractions from individual messages are used in a CS cannot be over-estimated, as they determine how compactly the EN that will be construed over time will encode the interaction patterns of a social context (and thus provide useful information to the agents using it). In addition to this, the generalisation strategy determines whether the EN will be a (multi-)tree, a directed acyclic graph (in which some messages have several predecessors) or even a general graph with cycles and/or loops; this may have a strong impact on the algorithms that have to be devised to make predictions based on the EN.

Question 4 refers to the problem of determining appropriate degrees of expectation strength or "expectability" for certain continuations. In the semantic models we present here, we will always use probability estimates derived directly from relative continuation frequencies and update them in the "naive" way upon incoming observations. Also, we will assume a prior uniform distribution over all possible continuations for any message. Again, this is not the only (even if the most obvious) way of updating weights. The speed of increasing or decreasing such a probability might depend on certain criteria, e.g. social power (if someone very powerful chooses a particular response, it will have a higher normative impact than if this is done by less powerful individuals). Or, weights may be initialised with a specific distribution that reflects some initial assumptions the agents make prior to any observation.

Finally, question 5 raises a very pragmatic issue: How should we use the EN to make predictions (or, in other words, to "compute" the meaning of the currently relevant symbols)? The view we adopt for our approach is that of computing a probability distribution over all possible message sequences up to a certain length. Alternative approaches include:

- computing only very likely future sequences,
- conducting an exhaustive search (depth-wise), and
- considering only very risky or very profitable paths.

Even if a CS is used to model communicative expectation structures, these considerations are still not sufficient to speak of a true understanding of communication semantics. For this purpose, we have to look more closely at the meaning of symbols in the context of an empirical model of statistical correlation.

4.1.4 Symbols and meaning

So far, we are able to model the relationship between a perceived sequence of communicative actions in an encounter and its potential consequences. But what precisely is the meaning of the symbols used in these messages according to this view?

Symbols encode expectations

Symbols used in a message must be representative of the expectations the uttering party associates with them. In the most general sense, they encode the state of the world that will be brought about after the expected consequences of the message occur. Common ways to describe such states fall into two categories:

- Descriptions of states of the world that result from communication. These may concern the physical environment or mental states of the interacting parties. For instance, if agent A informs agent B of X, A hopes that this will result in B believing X.
- Descriptions of the elements of expectation structures themselves, e.g. encodings of action sequences, plans, etc. If A threatens B to do Y if B does X, then this effectively means that A is sharing a part of his own expectation network with B (by informing B of his own future conditional reaction).

With this respect, since the only reason for uttering a message for an agent would be that these expectations cannot be fulfilled by the utterer himself, any message has the meaning of a *request*. As this may seem quite restrictive at first sight, let us look at some examples of how other types of messages can be interpreted as requests:

- Proposal: The agent making the suggestion requests acceptance or a counter-suggestion.
- Promise: The agent who commits himself to doing something requests the other to believe he will.
- Rejection: The agent rejecting what someone else requested is requesting that the requester accepts the rejection and that the interaction ends with no further consequences.

Effectively, what is requested in all these cases is the expected effect of uttering the symbol as derived from the CS of the agent who is using the symbol (and hopefully shared by the other communicating parties).

Special cases

Looking back at the CS view, some special cases of the use of such requests can be identified. Firstly, the response expected from the other might be deliberately under-specified. This is the case with completely new symbols that are not yet in the EN and which can have arbitrary (unknown) consequences. By employing a new symbol, an agent is effectively allowing the other to "fill in the meaning" by generating any response. A similar situation occurs if a symbol becomes highly ambiguous after having been followed by many different action sequences in the past. Such ambiguity may be instrumental in spawning a creative process (a phenomenon often experienced in human society).

Secondly, symbols that have occurred before may be re-used in a different context. Unless every symbol is supposed to have different semantics in every context (which would prohibit modular re-combination of existing expectation sub-graphs), re-using the expectations already associated with a symbol is a reasonable strategy. However, we should not forget that this results in a drastic modification of the expectations associated with the symbols on the current path that were uttered *prior* to the re-used symbol (i.e. its ancestors in the EN). In the most extreme strategy of connecting the existing node for that symbol with the current encounter path (prefix) upon re-use, for example, the context would simply have a meaning *identical* to that of the re-used symbol (and hence lose its own meaning). It is therefore very important to carefully consider to which extent previous meanings of symbols are integrated into different contexts after their re-use. Also, if the expectations associated with a re-used symbol are modified by its actual consequences in the new encounter, we have to consider modifying them in all other places in the EN where the same symbol occurs.

Objects, actions, actors and expectation encodings

Choosing appropriate representations for communicative expectation structures is of paramount importance when it comes to using them in agent reasoning. They should be easy to derive from observation and computationally tractable so that they can be readily used to make predictions about the future behaviour of other agents.

Since symbols encode expectations, they basically describe world states in terms of objects of the physical world, agents, and actions. In accordance with the view put forward in *speech act theory* (Austin 1962, Searle 1969), where *performatives* are used to distinguish between different types of messages on the grounds of speaker intention (defined in terms of desired effect of the utterance), we argue that performatives are useful as labels of expectation graph nodes. This is because the distinction between "sender and receiver", "intention" ("desired effect") and "content" is a very powerful one, as the ways in which agents (jointly) *act towards* things in the world are fairly limited, even if these can be applied to a huge variety of issues (objects, beliefs, etc.) to *talk about* in the world. From the standpoint of generalisation, this means that it is reasonable to abstract from

- individual senders and recipients of messages (as communicative expectations should hold across different agents), and
- message content (because the objects talked about vary much more than the possible intended consequences of acting towards them).

At this point, a simple example is useful to illustrate why this is the case. First, consider an expectation structure that is solely based on a statistical distribution of consequences:

If I say to Peter "Please open the window" he does so in 85% of all cases.

Note that in the most general interpretation of symbols, the entire utterance *"Please open the window"* is a single distinct symbol.

Generalising over different actors, this expectation would become

If A says to B "Please open the window", B does so in 85% of all cases.

Now, I can expect the message *"Please open the window"* to have comparable effects regardless of its recipient. Moreover, by adapting the probability with every respective observation, I am *forced* to consider its global meaning (that is averaged over all agents) when using it.

Moving from these levels of generalisation to a generalisation over *content*, however, marks a real leap in expressiveness. If the stored expectation is

If A asks B to do X, B does so in 85% of all cases this can formalised as

$$request(A,B,X) \stackrel{0.85}{\rightarrow} do(B,X)$$

and is reminiscent of illocution expressed through performatives in speech act theory (see section 2.3.2). The difference between this kind of generalisation and other generalisations is that the expected reaction itself is encoded in the utterance that precedes it.

Clearly, X can be instantiated with many more different values than B and A can exhibit reactions to the request (e.g. no reaction at all, doing something else than what was requested). What is more, the different reactions *matter* much more to the agents than the objects talked about – it is the different paths of action that are pursued after the utterance that make the difference, not the subject of discourse. In our formalisation of InFFrA, we will therefore employ performatives and abstract from content and participating actors whenever this is possible.

We do not imply by this that there is a universal set of speech act types, or that their meaning is fixed in terms of shared normative content regarding their preconditions and effects. They are rather considered node labels powerful enough to generate compact expectation structures, as they naturally encode the different types of intentions agents might have, even if these may signify different expectations in different contexts (i.e. we expect the same performative to occur on different paths with slightly different meaning).

4.1.5 Content and context constraints

In the above example, if A asks B to open the window, B is not expected to reply with a statement about international politics (at least not in pragmatic agent communication). B should either react by opening the window, by expressing some reason why he cannot, should not or must not open it, by delegating this task to someone else, etc. In any case, the content of B's response should somehow relate to opening the window.

This illustrates that there is something still amiss in our model of expectation structures, namely *the link to the environment* in terms of (i) physical objects and (ii) mental states. For expectations to make sense to the agent using them, they need to be conditioned with constraints he can process cognitively (e.g. through inference on his knowledge base).

These constraints fall into two categories: (i) *content constraints* which refer to the admissible contents of messages that can be used by the interacting parties in combination with certain performatives and (ii) *context constraints* that concern the applicability of particular expectation structures in certain situations.

Content constraints

Content constraints restrict the scope of possible contents that can be used in subsequent messages. The most common type of such constraints is that of *topic* or *theme* of a conversation, which, as suggested above, narrows down the set of possible (viz. reasonable) responses by restricting them to utterances that refer to a particular set of objects, actions, and mental states. Of course, this scope depends on the respective types of performatives in the expectation structure. For example, a call for bids in an auction allows for a much wider range of responses than an offer to sell something at a particular price.

The second, very important type of content constraints is that of *rationality constraints*. Basically, these concern the consistency of message content with agent rationality. Agents require others to act in accordance with rationality constraints because they *are* rational agents, i.e. through an implicit homogeneity assumption by which agents assume others to have similar cognitive capacities like themselves. Typical examples for such constraints include

- not claiming the opposite of what has been asserted before unless sufficient justification can be provided,
- acting towards (and also, not acting against) a goal that has been claimed to have been adopted (unless a goal is reached or unachievable), or
- being committed to accept all the logical consequences of statements that have previously been accepted as true.

In a sense, these constraints are nothing but assumptions regarding mental states that cannot be observed directly but are virtually made public with certain utterances, e.g. expressions of belief. While the mental state itself cannot be verified by an external observer, subsequent observable actions are required to be in accordance with it.

It is important to understand that Rationality constraints are one of the main aspects of communication among deliberative agents that make it radically different from "signalling behaviour" between machines or, say, insects.

Context constraints

In all the above examples, the expected consequences were identical under any circumstances. Quite differently, we might use the expectation

> *If I say to Peter "Please open the window", he does so in 95% of all cases if the window is closed. If the window is already open he never tries to open it.*

Now, the aggregation of different observations into a probability distribution is parametrised with environmental (in the sense of "communication-external") conditions grounded in the world model of the observer and thus endowed with additional information. Unless expectations are parametrised in this way, different symbols might have to be used for every different physical state of the entire system, because agents react differently to identical symbols under different conditions, depending on their local knowledge and their private motives.

Unlike content constraints, context constraints may never become visible in communication. They are internal to agent cognition and serve as a means of distinguishing between different meanings (consequences) of the same message under different circumstances. Hence, they are an instrument that can be used to *organise* expectation structures so that these make sense to the agent in different situations.

4.1.6 Deviance, rejection and conflict

Basing semantics on expectations implies considering that expectations may be violated both by the agent who holds them and by his peers. Violating expectations can be regarded as *deviance* with respect to the normative content of existing expectations.

Trivially, any utterance modifies the status of an EN (unless it is deliberately ignored), so every message is deviant in a way, even if this deviance only consists of reassuring the agent about the most probable outcome. Therefore, we only speak of *real* deviance if agents exhibit a communicative behaviour different from everything that could realistically have been expected. What we mean by this is that when an agent considers different possible outcomes of an interaction, he usually focuses on a subset of them and normally filters out whatever seems extremely

- improbable (asking someone to open the window is rarely followed by a lightning that injures the person so that this experience will infringe our confidence in such requests),
- irrational (even if murder is common in some social context, it is irrational to commit murder if the sanctions are drastic and the probability of getting caught is high),
- disproportionate in effect (you don't expect to get beaten in the face for asking someone to open the window).

A measure that can be used as an indicator for these cases is whether the modifications to the EN *anticipated* before an action took place were greatly exceeded by those *actually experienced* (we shall return to this issue in section 5.3.4).

Rejection is a more complex form of deviance. Rather than simply reacting in an unexpected way, it enables agents to express that they are not *willing* to react as expected by their adversary. Worse still, rejection might mean that they are not even willing to provide any further information regarding their own expectations or their willingness to adhere to expectations. In other words, the rejecting agent is telling his peer that he might do virtually anything (rather than restrict himself to what the other expected as a reaction).

A "no"[3] implies that the agent who is uttering is not interested in any further cooperation, at least not in the context of the present encounter. However, if an agent issued a proposal and received a rejection, this does not have to bear any severe consequences – the encounter may simply end without an agreement, the agent may make a new proposal, etc. Only if he rejects the other agent's (first) "no" the agents are in trouble: In this case, this second "no" translates to "be prepared for any reaction on my side on whatever you do", thus denying the possibility of any alignment between the two agents' actions, and this is nothing else but open *conflict*.[4]

Given that deviance is the source of innovation in interaction but that it can also cause conflict, one of the central issues in achieving coordination through communication becomes *how agents can identify a potential for cooperation to keep the other from resorting to conflict behaviour or to get him back on the track of cooperation.* As described in our remarks on conflict in interactionist theories (page 24), "intelligent social reconstruction" is necessary in such situations to enable a joint return to cooperative patterns of interaction. In section 6.3 we show how frames can be designed to achieve this kind of conflict management and conflict resolution.

[3] Note that we only consider "no" in the sense of rejection, not as a negative answer to a question (e.g. "are you cold?"), where "no" is a perfectly expectable reply and makes a statement about the valuation of a formula/variable). Also, a "no" which was expected (if the agent who it is being directed towards has a plan regarding what to do in case of "no") is not a "no" in that sense.

[4] This is a simplified view of Luhmann's (1995) theory of conflict as "double no". Note also that we are using a purely *communicative* notion of conflict here, which is very different from the definitions frequently used in DAI which usually refer to goal and resource conflicts (Müller and Dieng 2000, Tessier et al. 2000).

4.1.7 Communication and coordination

As we have argued before (assumption 2.10, p. 28), agents are generally confronted with the dilemma of trying to maximise their own autonomy (viz. independence of others' expectations) while at the same time they want their peers to act as predictably as possible. Reviewing this issue in the light of our observations on communication, we are able to provide a more specific description of how agents reason about communication and how we can model the process of inter-agent coordination on this basis.

Most generally, agents communicate when they are unable to achieve their goals on their own. If they identify a potential for beneficial joint action, they contact the respective agent(s); if contacted by someone else, they will participate in communication if they expect to benefit from it in some way. Whenever an interaction becomes too costly or does not seem to lead to a positive outcome anymore, agents will abandon it.

During communication, agents use their expectation structures to determine paths of interaction that lead to desirable outcomes. Thereby, the final outcome of a communication process should include changes to the physical environment or to the mental states of agents, since we do not assume that agents communicate just for the sake of it (i.e. messages themselves offer no utility to agents).

In this sense, a model of expectations, for example an EN, is used like a library of "communication plans" as it contains information about different paths of joint action execution, their consequences and the context within which they can be performed. In comparison to normal plans, the expected behaviour of communicative partners is of course only approximate, ambiguous, generalised and not necessarily efficient, for example if a long argument is necessary to achieve a very simple agreement.

As described above, the utterances that agents generate are requests for jointly realising a certain state of the world, the meaning of which is derived from expectations. From a coordination point of view, this means that interacting agents iteratively exchange different proposals that contain descriptions of world states they want to achieve, each of them according to their goals. In other words, agents constantly *negotiate* about what state of the world to achieve, and use their expectation structures to indicate the course of joint action that they would like the other to take. For example, by accepting someone else's suggestion and taking into account that the other agent knows our acceptance will result in readiness to play our part in the execution of what is being suggested, we are agreeing to implement the suggested joint action. By rejecting a proposal, on the other hand, our counter-proposal may be "nothing", i.e. we will expect the other agent to know that no joint action is going to result from the conversation.

Unfortunately, because expectations are built from experience, things get more complicated as communicative action during this negotiation process constantly modifies the expectation structures. So selecting appropriate messages using an expectation structure must not only take the immediate, *first-order* effects of the messages into account, but also the long-term evolution of the expectation structures which we can view as *second-order* effects of communication (cf. assumption 2.9, p. 28).

To summarise, the coordination process that unfolds between communicating agents is characterised by three main features:

1. Agents with different but potentially overlapping goals negotiate over different possible interaction sequences by making communicative choices on the grounds of their private expectation structures.

2. The exchanged messages correspond to expectations regarding potential outcomes, and fruitful coordination can only be achieved if actors' expectation structures are suitable (esp. with respect to divergent expectations) and if they make the right utterances.

3. Communication alters existing expectation structures. Reasoning about the long-term usefulness of a meaning structure (and about the degree to which this is actually influenced by one's current decisions) plays an important role in strategic communication.

These observations conclude our rather theoretical view of communication that is based on the idea of consequentialist semantics and on grounding meaning in expectation structures derived from empirical observation. In the following section we will introduce a formal model for InFFrA agents that is in accordance with these principles.

4.2 m²inffra

The m²InFFrA model of a frame-based architecture is a formalism for representing a communication system with consequentialist semantics from the viewpoint of an agent observer, with the distinctive property of employing *frames* to encode and manage expectation structures.

This *dual* view of frames – as expectation structures that define an evolving communication semantics on the one hand, and as "pragmatic scripts" that can be instrumentalised to achieve one's goals on the other – is the foundation of the formal definition of m²InFFrA. Before presenting its rather intricate details, we shall provide an overview of its main characteristics, especially with respect to abstract InFFrA.

4.2.1 Overview

m²InFFrA is the result of building a computational model of InFFrA that is *minimal* in the sense that all features of InFFrA are present but realised in the least complex way. The intuition behind this is that m²InFFrA can be used as a starting point for more elaborate InFFrA agents by representing a kind of "greatest common denominator" between many possible InFFrA-compliant designs.

With respect to the design of interaction frames, this results in the following set of features:

- Trajectories are represented as fixed-length sequences of message and physical action *patterns* in a format similar to that of speech acts that may contain variables. They are strictly turn-taking and involve only two interacting parties. Frame validity is determined by checking whether a prefix of the trajectory of a frame matches the currently perceived interaction sequence.

- Roles and relationships, context, and beliefs are captured as logical statements in lists of *conditions*. They are all treated equally when it comes to assessing adequacy and validity in the sense that the agent verifies their logical satisfiability using his knowledge base.

- Status is expressed by means of variable *substitutions* which replace variables in trajectories/conditions by concrete values. This allows for the representation of concrete instances of abstract trajectory and condition descriptions.
- Representing frame history is made possible by storing *lists* of conditions and substitutions as collections of past enactments of the frame. Also, the frequency of the individual cases is counted.
- Links between frames exist implicitly by virtue of constantly updated "matching counters" in all frames that match during an encounter. Frames maintain information about which frame they were generated from and, additionally, semantic links can be used in frame conditions which are treated as ordinary logical constraints.
- Extension is realised in a very simplistic fashion. Epistemic constraints can be mixed with other logical constraints in the condition lists. If no such constraints apply, the framing agent assumes the frames he maintains to be commonly known among the interacting parties.

In short, m^{2}InFFrA frames consist of (i) a simple trajectory sequence (which may include variables and wildcards), (ii) lists of conditions and substitutions for past instances of the frame that have been experienced, and (iii) counters for the number of encounters in which prefixes of the trajectory matched the experienced message sequence.

All parts of the frame that have to be verified in a logical way are combined in the conditions attribute, i.e. the conceptual distinction between roles and relationships, context, beliefs, framing links, and extension made in 3.3 is abandoned here in favour of simplicity. Note that it is still possible to model all these different attributes, but the problem is now shifted to a general level of logical inference, the intuition being that "no matter what kind of conditions have to be satisfied, they all have to be verified at a logical level for the frame to be feasible". In an expectation structure view, the logical conditions associated with a frame trajectory reflect the *context constraints* and *content constraints* that are relevant for the respective communication pattern.

As concerns framing, m^{2}InFFrA is characterised by the following properties:

- It is assumed that an agent can estimate the utility of a future sequence of (ground) messages and actions at any time. Encounters are only started if there are frames that suggest an increase in utility, and the agent participates in interactions started by others only as long as there are still matching frames according to which a utility gain can be expected.
- The perceived frame consists only of the sequence of ground messages and actions that has been observed so far in the present encounter. Unlike general InFFrA, matching is performed for *all* frames in the repository (not just the active frame) but only with respect to trajectories (validity). This allows for narrowing down the choices of alternative frames during iterative framing cycles, thus reducing the complexity of trial instantiation.
- During the assessment phase, it is checked whether there are conditions in the active frame that can still be met (adequacy), and whether the frame offers encounter conclusions that are profitable utility-wise (desirability).

- In the adjustment phase, all frames that still match the perceived frame are iteratively assessed in the same way as the active frame. If no suitable frame can be found, the agent terminates the encounter.

- As for enactment, the agent picks that ground variant of the current active frame that promises the highest expected utility out of the possibilities offered by the (potentially non-ground) frame trajectory.

- Behaviour generation is trivial: If it is the agent's turn, he executes the next action on the trajectory, else he simply waits for the adversary's action. Social InFFrA choices override any action decisions the agent makes in his sub-social reasoning, i.e. the agent is always "controlled" by InFFrA unless this layer does not output any action.

Note that this description does not include an account of how frames are created, maintained, and adjusted, or, in other words, m^{2}InFFrA only provides a model for *applying* existing frames, but not for *learning* them. The reason for this is that the model should be kept as simple as possible at this point, since an entire chapter (chapter 5) is dedicated to frame learning and decision making.

It should also be remarked that m^{2}InFFrA does not specify low-level control issues such as control loops for receiving and dispatching messages, details of how parallel conversations are handled, criteria to discriminate between consecutive encounters, etc. These are considered implementation details and explained in chapter 6.

4.2.2 Preliminaries and notation

In order to introduce m^{2}InFFrA formally, some auxiliary definitions are required. The following paragraphs introduce formal languages for logical formulae and message patterns, some basic definitions regarding encounters, miscellaneous auxiliary predicates and general notational conventions.

Logical language

First of all, we need to define two formal languages: (i) a logical language that will be used for representing knowledge base contents and frame conditions, and (ii) a language for messages and message patterns.

$\mathcal{L}$ is a simple (essentially propositional) logical language that uses atomic propositions $Statement = \{p, q(X, s), \ldots\}$ which may contain variables (denoted by capital letters X, Y, etc.) that are implicitly universally quantified and range over finite domains). Atomic propositions are combined through the usual connectives $\vee$, $\wedge$, $\Rightarrow$ and $\neg$. Finally, $\mathcal{L}$ contains the logical constants "`true`" and "`false`", and braces $()$ for grouping sub-expressions together (the language is formally given by grammar G in table 4.1).

Given the set of all possible interpretations

$$\mathcal{I} = \{I : GroundStatement \to \{\mathrm{true}, \mathrm{false}\}\}$$

(where $GroundStatement = \{\varphi \in Statement | \varphi \text{ is ground}\}$) we define the relation $\models \subseteq \mathcal{I} \times$

$$
\begin{array}{rcl}
\mathit{Object} & \rightarrow & \texttt{table} \mid \texttt{stone} \mid \ldots \\
\mathit{ObjectVar} & \rightarrow & \texttt{O}_1 \mid \texttt{O}_2 \mid \ldots \\
\mathit{Agent} & \rightarrow & \texttt{agent\_1} \mid \ldots \mid \texttt{agent\_n} \\
\mathit{AgentVar} & \rightarrow & \texttt{A}_1 \mid \texttt{A}_2 \mid \ldots \\
\mathit{PhysicalAction} & \rightarrow & \texttt{move\_object} \mid \texttt{pay\_price} \mid \texttt{deliver\_goods} \mid \ldots \\
\mathit{PhysicalActVar} & \rightarrow & \texttt{X}_1 \mid \texttt{X}_2 \mid \ldots \\
\mathit{Message} & \rightarrow & \mathit{Performative}(\mathit{Agent}, \mathit{Agent}, \mathit{LogicalExpr}) \\
 & & \mid \texttt{do}(\mathit{Agent}, \mathit{PhysicalAction}) \\
\mathit{MsgPattern} & \rightarrow & \mathit{Performative}(\mathit{AgentTerm}, \mathit{AgentTerm}, \mathit{Content}) \\
 & & \mid \texttt{do}(\mathit{AgentTerm}, \mathit{PhysicalActTerm}) \\
\mathit{MsgVar} & \rightarrow & \texttt{M}_1 \mid \texttt{M}_2 \mid \ldots \\
\mathit{Performative} & \rightarrow & \texttt{accept} \mid \texttt{propose} \mid \texttt{reject} \mid \texttt{inform} \mid \ldots \\
\mathit{AgentTerm} & \rightarrow & \mathit{Agent} \mid \mathit{AgentVar} \\
\mathit{Content} & \rightarrow & \mathit{LogicalExpr} \mid \mathit{ContentVar} \\
\mathit{ContentVar} & \rightarrow & \texttt{C}_1 \mid \texttt{C}_2 \mid \ldots \\
\mathit{PhysicalActTerm} & \rightarrow & \mathit{PhysicalAction} \mid \mathit{PhysicalActVar} \\
\mathit{LogicalExpr} & \rightarrow & (\mathit{LogicalExpr} \Rightarrow \mathit{LogicalExpr}) \mid (\mathit{LogicalExpr} \vee \mathit{LogicalExpr}) \\
 & & \mid (\mathit{LogicalExpr} \wedge \mathit{LogicalExpr}) \mid \neg \mathit{LogicalExpr} \\
 & & \mid \mathit{Statement} \mid \mathit{MsgPattern} \\
\mathit{Statement} & \rightarrow & \mathit{Head} \mid \mathit{Head}(\mathit{TermList}) \mid \texttt{true} \mid \texttt{false} \\
\mathit{Head} & \rightarrow & \texttt{it\_rains} \mid \texttt{loves} \mid \ldots \\
\mathit{TermList} & \rightarrow & \mathit{TermList}, \mathit{Term} \mid \mathit{Term} \\
\mathit{Term} & \rightarrow & \mathit{Object} \mid \mathit{ObjectVar} \mid \mathit{Agent} \mid \mathit{AgentVar} \\
 & & \mid \mathit{Message} \mid \mathit{MsgVar} \mid \mathit{PhysicalAction} \mid \mathit{PhysicalActionVar} \\
\mathit{SubstList} & \rightarrow & \langle \mathit{SubstList}' \rangle \\
\mathit{SubstList}' & \rightarrow & \mathit{SubstList}'\ \mathit{Subst} \mid \varepsilon \\
\mathit{Subst} & \rightarrow & [\mathit{ObjectVar}/\mathit{Object}] \mid [\mathit{AgentVar}/\mathit{Agent}] \\
 & & \mid [\mathit{PhysicalActVar}/\mathit{PhysicalAction}] \\
 & & \mid [\mathit{MsgVar}/\mathit{Message}] \mid [\mathit{ContentVar}/\mathit{LogicalExpr}]
\end{array}
$$

Tab. 4.1: Grammar G defining the syntax of $\mathcal{L}$, $\mathcal{M}$ and $\mathcal{M}_c$ (terminal symbols are shown in `typewriter` font, ε denotes the empty word).
Logical formulae refer to four different kinds of things in the universe of discourse: (physical) objects, agents, messages, and physical actions. These (or typed variables for them) are used to form terms which, in turn, are the arguments of statements. Statements can be combined to logical formulae, with the speciality that *MsgPattern* objects can be used as statements to denote the event of a message pattern being performed (under existential quantification).
As a reified object, logical formulae are also used as message contents of type *Content*. In substitutions, these contents can be replaced as a whole by variables, as is the case for agents, messages, objects and physical actions.

$\mathcal{L}$ in the usual way by induction over formulae $\varphi \in \mathcal{L}$ and interpretations $I \in \mathcal{I}$:

$$
\begin{array}{rl}
I \models \varphi \text{ iff} & \varphi \in \mathit{GroundStatement} \text{ and } I(\varphi) = \text{true} \\
I \models \varphi \text{ iff} & \forall t. I \models \varphi[v/t], v \in \left\{ \begin{array}{c} \mathit{AgentVar} \\ \mathit{ObjectVar} \\ \mathit{PhysicalActVar} \\ \mathit{MsgVar} \end{array} \right\}, t \in \left\{ \begin{array}{c} \mathit{Agent} \\ \mathit{Object} \\ \mathit{PhysicalAction} \\ \mathit{Message} \end{array} \right\} \\
I \models \neg\varphi \text{ iff} & I \not\models \varphi \\
I \models \varphi \vee \chi \text{ iff} & I \models \varphi \text{ or } I \models \chi
\end{array}
$$

where is a $[v/t]$ is a variable substitution in the usual sense and we write $\varphi[v/t]$ for the formula that results from replacing all occurrences of variable v in φ by ground term t.

The operators $\wedge$, $\Rightarrow$ and $\forall$ can be defined as abbreviations through the other operators. Also, we write $\models \varphi$ if φ is a tautology that is satisfied by any $I \in \mathcal{I}$. A *knowledge base* $KB \in 2^{\mathcal{L}}$ can be any set of formulae from $\mathcal{L}$. For simplicity, we will often write $KB \models \varphi$ to express $\models (\wedge_{\varphi' \in KB} \varphi' \Rightarrow \varphi)$.

Messages and message patterns

The language $\mathcal{M}$ of message patterns defines the syntax of message patterns (or templates) we will use for describing frame trajectories. For the purpose of restricting certain definitions to ground (i.e., variable-free) messages, we further identify the language of *concrete* messages $\mathcal{M}_c$ as a subset of $\mathcal{M}$.

Actually observed, concrete messages (of type *Message* in table 4.1) can be either physical actions of the form $\mathtt{do}(a, ac)$ where a is the executing agent and ac is a symbol used for a physical action (of type *PhysicalAction*), or non-physical messages $\mathit{performative}(a, b, c)$ sent from a to b with content c. Thereby, the symbols used in the *Agent*, *Object* and *PhysicalAction* rules contain domain-dependent symbols the existence and proper grounding of which we take for granted.

The content of non-physical messages is given by the *LogicalExpr* type, where logical expressions are essentially formulae from $\mathcal{L}$ composed of predicates that refer to agents and objects in the world, with the speciality that we use *MsgPattern* expressions as action predicates. This is an abbreviated notation for referring to communicative actions in the sense that if message (or pattern) m is used as part of a logical expression in message content, then this part of the statement will be true if the action (or any action from the set of actions represented by the pattern) is executed (of course, the semantics of this can be augmented with a notion of time). [5]

Unlike concrete messages, the message patterns used in frame trajectories are of type *MsgPattern* and may also contain variables for agents (*AgentVar*), physical actions (*PhysicalActVar*), objects (*ObjectVar*) and message content (*ContentVar*), but not for performatives. The fact that performatives cannot themselves be replaced by variables has far-reaching consequences, because it implies that if a concrete message is matched against a

[5] There is an important difference between a logical expression $l \in \mathcal{L}$ used in an agent's internal reasoning process and a "logical expression" being the content of a non-physical action. The latter is not a logical expression, but an object of the universe that *denotes* a logical expression or – in the case of message patterns and in the presence of variables – a set of logical expressions. Hence, a message has to be *interpreted* by its recipient to derive the logical quality of the content; before that, it is just a string.

pattern that contains variables, a match will only occur if message and pattern share the same performative. The theoretical reasons for this restriction have been provided in section 4.1.4 (p. 86); in practical terms, this means that a suitable organisation of expectation structures into frames in m²InFFrA depends very much on performatives being telling with respect to the outcomes of interactions. In other words, m²InFFrA relies much more on predictions made on the grounds of the performatives perceived rather than the agents involved or the message content that appears in the messages.

Based on the rules of grammar G in table 4.1 we can formally define

$$\begin{aligned} \mathcal{L} &= \{\varphi \in \Sigma^* | LogicalExpr \Rightarrow_G^* \varphi\} \\ \mathcal{M} &= \{m \in \Sigma^* | MsgPattern \Rightarrow_G^* m\} \\ \mathcal{M}_c &= \{m \in \Sigma^* | Message \Rightarrow_G^* m\} \end{aligned}$$

if Σ is the set of terminal character symbols (i.e. all those symbols that do not occur on the left hand side of a rule, including operators, brackets, commas, and special symbols [,], $\langle$, $\rangle$ and /). For future reference, we also define $\mathcal{V} = \{ObjectVar, AgentVar, PhysicalActVar, ContentVar, MsgVar\}$ as the set of non-terminal symbols from which variable symbols are derived.

Encounters

m²InFFrA caters for discrete, turn-taking, two-party conversations only. Hence, we have to assume that m²InFFrA agents have the computational means to group perceived messages together to so-called *encounters* which we can write as words $w \in \mathcal{M}_c^*$. Encounter identifiers $\mathcal{E} = \{e, f, \ldots\}$ are used to explicitly refer to specific (past) encounters. Let $message(m, e, i)$ denote that m is the ith message in encounter e. The following statement captures the fact that each encounter occurs exactly once and contains a number of subsequent messages:

$$\begin{aligned} \forall e \in \mathcal{E}.\forall m \in \mathcal{M}_c.\forall i \in \mathbb{N}.message(m, e, i) \Rightarrow \Big(&(\forall m' \in \mathcal{M}_c.message(m', e, i) \Rightarrow m = m') \\ &\wedge \big(i > 1 \Rightarrow \exists m' \in \mathcal{M}_c.message(m', e, i-1)\big)\Big) \end{aligned}$$

$label(e) = m_1 m_2 \cdots m_i \cdots m_n$ yields the concatenation of all messages of an encounter e (i.e. $\forall i.message(label(e)[i], e, i)$ holds).

For any message $m \in \mathcal{M}$, $sender(m)$ and $receiver(m)$ return the sender and the set of recipients of the message respectively (which may be variables, if m is not a concrete message). Unless m is a physical action, which can theoretically be observed by every agent[6], $receiver(m)$ yields a singleton set. Further, for every encounter e there are two agent symbols (type *Agent* of grammar G in table 4.1) a_1, a_2 such that $initiator(e) = a_1$ and $responder(e) = a_2$ return the agent that initiated an encounter and his communication partner.

Uniqueness of senders/receivers and initiators/responders as well as strict turn-taking are captured by the following assumptions (where "=" denotes term equivalence in the

[6] For simplicity, we assume that observation of the effects of a physical action allows for unequivocal identification of the agent who executed the action.

usual sense):

$$
\begin{aligned}
\forall m \in \mathcal{M}.\forall a_1, a_2 \in Agent.a_1 = sender(m) \wedge a_2 = sender(m) &\;\Rightarrow\; a_1 = a_2 \\
\forall e \in \mathcal{E}.\forall a_1, a_2 \in Agent.a_1 = initiator(e) \wedge a_2 = initiator(e) &\;\Rightarrow\; a_1 = a_2 \\
\forall e \in \mathcal{E}.\forall a_1, a_2 \in Agent.a_1 = responder(e) \wedge a_2 = responder(e) &\;\Rightarrow\; a_1 = a_2 \\
\forall e \in \mathcal{E}.\forall m, m' \in \mathcal{M}_c.\forall i.(message(m, e, i) \wedge message(m', e, i+1) &\;\Rightarrow \\
sender(m) \in receiver(m') \wedge sender(m') \in receiver(m))&
\end{aligned}
$$

When we speak of the *encounter prefix* of an encounter e, we refer to the sequence $m_1 m_2 \cdots m_i$ where $i \leq n$ and $label(e) = m_1 m_2 \cdots m_n$. Given such a prefix, we refer to $m_{i+1} \cdots m_n$ as the *encounter postfix* or the *continuation* of e after $m_1 m_2 \cdots m_i$.

Miscellany and notational conventions

We shall make frequent use of a predicate $unify(\cdot,\cdot)$ that is true whenever two terms (or ordered lists of terms) can be unified. In this case, we write $unifier(\cdot,\cdot)$ for the most general unifier returned by a standard first-order unification procedure.

Also, we use

$$
\Delta(S) = \Big\{ f : \mathcal{S} \to [0;1] \;\Big|\; \sum_{s \in S} f(s) = 1 \;\wedge\; \forall s \in S.f(s) \geq 0 \;\wedge\; \exists S' \subseteq_{fin} S. \sum_{s' \in S'} f(s') = 1 \;\wedge\; \forall s' \in S'.f(s') > 0 \Big\}
$$

to denote finite-support discrete probability distributions over arbitrary sets S.

As for notational conventions, we use $s = \langle s_1, \ldots, s_n \rangle$ to distinguish ordered lists from sets. If we write them as *words* without enclosing braces, we leave out commas in favour of (sometimes omitted) concatenation/multiplication dots "$\cdot$". We write $\langle\rangle$ or ε for the empty list/word, s_i or $s[i]$ for the ith element of a list/word and $s[i{:}j]$ for the sublist/substring of s ranging from indices i to j inclusively. For substitutions (i.e. ordered lists of variable-term or variable-variable pairs $\langle [v_1/t_1], \ldots, [v_k/t_k] \rangle$), we mostly use Greek letters ϑ and χ and for sets/lists of these the respective capital letters, e.g. Θ. We denote application of a substitution to a formula $\varphi \in \mathcal{L}$ and lists thereof by simply "concatenating" the substitution(s) to these. For example, $\varphi\vartheta$ is the result of applying a substitution list $\vartheta \in SubstList$ to a logical formula $\varphi \in \mathcal{L}$.

Calligraphic capitals such as $\mathcal{L}$, $\mathcal{M}$, $\mathcal{E}$ are used for sets (as above), and we use `typewriter` font for strings in actual communication. For convenience we sometimes abuse some non-terminal symbols of grammar G in table 4.1 to denote the set of elements that would be generated with the respective non-terminal as starting symbol.

4.2.3 Interaction frames in m²inffra

Interaction frames in m²InFFrA consist of a sequence of communicative action patterns that describes a set of possible trajectories and of lists of variable substitutions and conditions which applied in previous encounters that matched the trajectory. Additionally, they count the number of times (i) that encounters occurred that matched prefixes of the trajectory and (ii) that a particular substitution occurred in a past encounter.

Definition

m²InFFrA agents maintain a frame repository $\mathcal{F} = \{F_1, \ldots, F_n\}$ in which they record knowledge about past interactions in the form of frames. These F_i are defined as follows:

Definition 4.1: A *frame* is a quintuple $F = (T, \Theta, C, h, h_\Theta)$, where

- $T = \langle p_1, p_2, \ldots, p_n \rangle$ is the frame *trajectory*, a sequence of message patterns $p_i \in \mathcal{M}$;
- $\Theta = \langle \vartheta_1, \ldots, \vartheta_m \rangle$ is an ordered list of substitutions $\vartheta_j = \langle [v_1/t_1], \ldots, [v_k/t_k] \rangle$ where $[v_i/t_i] \in \mathit{Subst}$ (as defined in table 4.1);
- $C = \langle c_1, \ldots, c_m \rangle$ is an ordered list of logical condition sets such that $c_j \in 2^{\mathcal{L}}$ is the condition set relevant under substitution ϑ_j;
- $h \in \mathbb{N}^{|T|}$ is a *trajectory occurrence counter* list that counts the occurrence of each sequence that matched a prefix of the trajectory T in previous encounters;
- $h_\Theta \in \mathbb{N}^{|\Theta|}$ is a *substitution occurrence counter* list counting the occurrence of each member of the substitution list Θ in previous encounters.

■

When speaking about frames, we will use some further notation:

- We write $T(F)$, $\Theta(F)$, $C(F)$, $h(F)$, $h_\Theta(F)$ for functions that return the respective elements of a frame F.
- The abbreviated syntax $T_h(F) = \xrightarrow{h(F)[1]} p_1 \xrightarrow{h(F)[2]} p_2 \cdots h(F)[n] \rightarrow p_n$ is convenient to combine $T(F)$ and $h(F)$ in one expression.
- Similarly, $\Theta_{h_\Theta}(F)[i] = \xrightarrow{h_\Theta(F)[i]} \Theta(F)[i]$ is used to combine $\Theta(F)$ and $h_\Theta(F)$.

In informal terms, the semantics of this frame are as follows: the agent who "owns" F has experienced $h_1 = h(F)[1]$ encounters whose first message matched the first element $p_1 = T(F)[1]$ of the trajectory. $h_2 = h(F)[2]$ of these h_1 encounters continued with a message that matched $p_2 = T(F)[2]$, and so on. This also implies that there was no encounter with prefix $p_1 \cdots p_n$ that continued after p_n (unless other frames suggest such a continuation, of course).

How are the condition and substitution lists to be interpreted? In the simplest case, if $w = \langle m_1, \ldots, m_n \rangle$ was the message sequence of the jth encounter that matched $T(F)$ (out of a total of $h(F)[n]$ encounters that matched the whole trajectory), ϑ_j is the substitution that unifies w with $T(F)$ (i.e. $\vartheta_j = \mathit{unifier}(w, T(F))$) and c_j is a set of conditions that held during this encounter under ϑ_j (i.e., $KB \models c_j\vartheta_j$ was true at that time). If used in this way, substitutions and conditions can be assigned to individual "cases" of frame occurrence in a one-to-one mapping.

At the same time, Θ and C can also be used to generalise over similar encounters, in which case $h_\Theta(F)[i]$ is the number of encounters that matched $T(F)\Theta(F)[i]$ (we write $T\vartheta$ for the trajectory that results from applying ϑ to every element of T). For example, a substitution may deliberately leave some variable in the frame trajectory unbound to allow for a number of respective concrete values in the cases represented by it.

Either way, C, Θ, and h_Θ capture the history of past encounters in which the frame was executed as a whole. By means of h, the frame also keeps track of "prefix encounters" that did no longer match or ended after some initial portion of the trajectory. Condition sets and substitution lists are not maintained for these incomplete executions of the frame.

Some important details should be noted that may not be obvious at first glance:

1. Whenever a message sequence occurs during an encounter, the h-vectors are updated for *each* frame that contains this sequence as a prefix of its trajectory:

   $$(v \in \mathcal{M}_c^* \text{ has occurred } n \text{ times} \wedge |v| = i) \Rightarrow \\ \forall F = (v'w, \Theta, C, h, h_\Theta) \in \mathcal{F}, w \in \mathcal{M}^*. \forall i \leq |v|. (unify(v', v) \Rightarrow h_i = n)$$

   If we view a collection of frames as an expectation network (cf. section 4.1.2), this means that prefixes of different frames which can be unified with k past encounters of length n refer to the same (pre-)path by virtue of having a trajectory occurrence counter value of k throughout their first n elements.

2. To keep $\mathcal{F}$ concise, we will assume that the trajectories of all frames are different, i.e.

   $$\forall F, G \in \mathcal{F}. F \neq G \Rightarrow T(F) \neq T(G).$$

   Note, however, that the trajectories of two frames may unify, i.e. the sets of encounters they represent may overlap.

3. Since a condition set c_j may contain conditions required for and/or precipitated by the actions of the trajectory, the agent must have action rules at his disposal (for physical actions) to discriminate between these two categories in order to be able to assess when a frame will be applicable.

   For example, if step p_{i+1} requires a precondition φ that is brought about by step p_i, the agent should know that this is the case and that φ does not have to be ensured by other means.

4. Application of a substitution $\vartheta \in \Theta$ to the trajectory need not, in general, yield ground messages. In this case, the frame does not provide concrete values for all trajectory variables. This can be useful for generalisation purposes, for example.

5. If $C[j]\vartheta$ contains variables for any j, these are implicitly universally quantified. Remember, though, that $\mathcal{L}$ is still essentially a propositional language, because this quantification only ranges over finite sets of objects.

6. A frame inherently distinguishes between *initiator* and *responder* party and assumes that these take turns. However, a frame need not necessarily make a precise statement regarding which agent will fill which of these roles in a given encounter (i.e. agent variables need not be bound to concrete values by substitutions).

The ingredients of InFFrA frames introduced in chapter 3 are present in this model in the following way: Roles and relationships, context and beliefs are all captured in the condition sets of C; the trajectory is reduced to a simple sequence T of message patterns; the history of the frame (and of previous successful completions) is stored in C, Θ, h and h_Θ, and links between frames are implicitly maintained by cross-counting the occurrence of prefixes of T. Before presenting the formal semantics of frames, let us briefly look at an example:

$$
\begin{aligned}
F = \Big\langle & \big\langle \xrightarrow{5} \texttt{propose}(A_1, A_2, \texttt{do}(A_1, X_1)) \xrightarrow{3} \texttt{accept}(A_2, A_1, \texttt{do}(A_1, X_1)) \\
& \xrightarrow{2} \texttt{do}(A_1, X_1) \big\rangle, \\
& \big\langle \{ \mathit{self}(A_1), \mathit{other}(A_2), \mathit{can}(A_1, \texttt{do}(A_1, X_1)) \}, \\
& \;\{ \mathit{agent}(A_1), \mathit{agent}(A_2), \mathit{action}(X_1) \} \big\rangle, \\
& \big\langle \xrightarrow{4} \langle [A_1/\texttt{agent\_1}], [A_2/\texttt{agent\_2}] \rangle, \\
& \xrightarrow{1} \langle [A_1/\texttt{agent\_3}], [A_2/\texttt{agent\_1}], [X_1/\texttt{deliver\_goods}] \rangle \big\rangle \Big\rangle
\end{aligned}
$$

For reasons of convenience, we use the syntax $\langle T_h(F), C(F), \Theta_{h_\Theta}(F) \rangle$ instead of $(T, \Theta, C, h, h_\Theta)$ here (as we will frequently do below). The frame summarises the following interaction experience:

- Five encounters started with a message matching $\texttt{propose}(A_1, A_2, \texttt{do}(A_1, X_1))$, three of them continued with $\texttt{accept}(A_2, A_1, \texttt{do}(A_1, X_1))$ and two of these were then concluded by agent A_1 performing the physical action X_1.
- Two encounters have terminated after the first message or were continued with a message that does not match $\texttt{accept}(A_2, A_1, \texttt{do}(A_1, X_1))$ and a further encounter turned out differently (or ended) after the second message.
- Two substitutions applied in these five encounters (the first of which is a generalisation over four encounters that leaves X_1 unspecified).

Note that while this specific example does not contain any duplicate conditions or substitutions, we do not require that each of them be unique in the general case.

Substitutions

If we look at the situation that occurs as an encounter unfolds, it is clear that given the already observed messages $\langle w_1, \ldots, w_k \rangle$ and the frame trajectory $T = \langle p_1, \ldots, p_n \rangle$, the scope of all variable substitutions for the remaining messages is progressively narrowed down, if the remaining messages/actions are to match $p_{k+1}, \ldots, p_n$. At this point, we should make some additional definitions that will be useful when talking about this process.

Definition 4.2: Let $F = (T, \Theta, C, h, h_\Theta)$ a frame and $w \in \mathcal{M}_c^*$. The substitution *fixed* by w in F is defined as

$$
\vartheta_{\mathit{fixed}}(F, w) = \begin{cases} \mathit{unifier}(w, T[1{:}|w|]) & \text{if } \mathit{unify}(w, T[1{:}|w|]) \\ \perp & \text{else} \end{cases}
$$

■

In other words, the fixed substitution imposed by w on F is the most general unifier that of the encounter prefix w with the initial part of the trajectory of F that is as long as w. With this, $\vartheta_{\mathit{fixed}}$ captures which variables in $T(F)$ have already been committed to certain values due to matching with w.

Starting from this, we would like to define $\Theta_{poss}(F, KB, w)$ to be the set of substitutions that are still possible if the remaining elements of $T(F)$ are still to be executed. Some auxiliary definitions are necessary to establish a definition of Θ_{poss}. Firstly, let

$$\mathit{postfix}(w, v) = \begin{cases} v' & \text{if } \exists v' \in \mathcal{M}^*.w = vv' \\ \perp & \text{else} \end{cases}$$

return the postfix v' of a sequence v in w if v is a prefix of v, and $\perp$ else. Further, let $\mathit{variables}(x) = \{v \prec x \mid V \Rightarrow_G^* v, V \in \mathcal{V}\}$ be the set of variables contained in a logical expression or a message pattern where the sub-term relation $\prec \subset \mathcal{L} \times \mathcal{L}$ is defined as follows for any two $\varphi, \varphi' \in \mathcal{L}$:

$$\varphi \prec \varphi' \Leftrightarrow \big(\exists s, s' \in (V \cup \Sigma)^*.\neg(s = s' = \varepsilon) \wedge (\mathit{LogicalExpr} \Rightarrow_G^* s\varphi s' \Rightarrow_G^* \varphi')\big)$$

Also, for any set S or list L let $\mathit{variables}(S) = \cup_{s \in S} \mathit{variables}(s)$ and $\mathit{variables}(L) = \cup_{1 \le i \le |L|} \mathit{variables}(L[i])$, respectively.

Definition 4.3: The set of *possible substitutions* in frame $F = (T, \Theta, C, h, h_\Theta)$ given an encounter prefix w and a knowledge base KB is defined as

$$\begin{aligned}\Theta_{poss}(F, KB, w) = \big\{\vartheta \in \mathit{SubstList} \big| &\vartheta = \vartheta_{\mathit{fixed}}(F, w)\vartheta' \\ &\wedge KB \models \forall (p_i = \mathit{postfix}(T(F)\vartheta, w\vartheta)[i]).\mathit{can}(\mathit{sender}(p_i), p_i) \\ &\wedge \exists 1 \le i \le |C(F)|.KB \models C[i]\vartheta \\ &\wedge \exists 1 \le j \le |C(F)|.\mathit{variables}(\vartheta) \subseteq \mathit{variables}(T(F)) \cup \mathit{variables}(C(F)[j])\big\}\end{aligned}$$

■

Hence, $\Theta_{poss}(F, KB, w)$ is the set of all substitution lists ϑ such that:

1. ϑ is an extension of $\vartheta_{\mathit{fixed}}$ (i.e. a substitutions that contains $\vartheta_{\mathit{fixed}}$ and – optionally – additional variable bindings).

2. All the remaining steps $p_i = \mathit{postfix}(T(F)\vartheta, w\vartheta)[i]$ of the trajectory under ϑ can be executed by their respective sender (in the case of physical actions, the sender is the agent who is executing them).

   A special predicate $\mathit{can} : \mathit{Agent} \times \mathcal{M} \to \{\text{true}, \text{false}\}$ has to be available to verify this condition. By assuming that $\forall a.\forall m.(m \notin \mathcal{M}_c \Rightarrow \neg \mathit{can}(a, m))$, we ensure that ϑ is *complete* in the sense that $T(F)$ can in fact be executed under ϑ, i.e. that $T(F)\,\vartheta \in \mathcal{M}_c^*$.[7]

[7] Although this may look like a domain-dependent restriction, it is reasonable in any domain, as it simply excludes the possibility of agent names being bound to variables in action/message terms in a way that the agents cannot execute the respective action at all (for instance, in $\texttt{do}(A, \mathit{spendMoneyOf}(B))$ a substitution $\langle [A/\texttt{agent\_1}], [B/\texttt{agent\_1}] \rangle$ makes little sense).

3. The substitution ϑ satisfies at least one of the condition sets in the list $C(F)$. This means that condition sets stored in a frame constitute hard applicability constraints for a frame so that Θ_{poss} is always empty unless at least one of these sets is satisfied under current belief.

4. The substitution ϑ is *minimal* in the sense that it substitutes (at most) the variables contained in the trajectory plus those of any condition set (clearly, as no more than one condition set needs to apply at the same time, the variables of different condition sets need not be mixed in the same ϑ).

   Apart from ignoring useless substitutions, this also ensures that the number of all such ϑ is bounded, while *SubstList* itself is infinite.

4.2.4 Retrospective semantics

As with any expectation structure used to represent empirical semantics, there are two dimensions to the meaning of frames. Firstly, in *retrospect*, they provide information about previous encounters (that may be filtered, condensed, generalised or may even include an element of oblivion). Secondly, they are used to make predictions by virtue of a *prospective* semantics. Since the retrospective case is simpler, it shall be treated first.

For this purpose, let $\mathcal{E}$ the set of encounters perceived so far and $\mathcal{F}$ a frame repository used to capture these encounters. We assume a discrete time scale $\mathcal{T}$ and a function $start : \mathcal{E} \rightarrow \mathcal{T}$ that determines the point in time when an encounter started. Further, let $KB(t)$ denote the contents of the knowledge base at time step $t \in \mathcal{T}$. Then, the following two invariants on $\mathcal{E}$ and $\mathcal{F}$ have to hold at any time:

1. Every encounter $e \in \mathcal{E}$ *as a whole* is accounted for by exactly one substitution of exactly one frame and the corresponding condition set that held when e took place:

$$\begin{aligned}\forall e \in \mathcal{E}.\exists_1 F \in \mathcal{F}.\exists_1 i \leq |\Theta(F)|.\Big(&unify(T(F)\Theta(F)[i], label(e))\\ &\wedge KB(start(e)) \models C(F)[i]\Theta(F)[i]unifier(T(F)\Theta(F)[i], label(e))\Big)\end{aligned} \tag{4.1}$$

   As noted earlier, execution of trajectory actions may itself modify the contents of KB. Therefore, according to this constraint, we actually would have to make explicit when these changes occur, and that some preconditions only have to hold after certain actions are executed. For example, if φ only has to hold before $T(F)[4]$ is executed and is brought about by $T(F)[2]$, it would be appropriate to formalise this as condition set

$$\{precondition(\varphi, T(F)[4]), postcondition(\varphi, T(F)[2])\}$$

   by using "lifting" predicates such as *precondition* and *postcondition*, so that the condition set itself already holds at the start of the encounter.

2. By virtue of h, every frame also states how often encounters have been observed whose prefixes matched the beginning of its trajectory:

$$\begin{aligned}&\forall F \in \mathcal{F}.\forall 1 \leq i \leq |T(F)|.\\ &\qquad h(F)[i] = k \Leftrightarrow \Big|\big\{e \in \mathcal{E}\big|unify\big(label(e)[1{:}i], T(F)[1{:}i]\big)\big\}\Big| = k\end{aligned} \tag{4.2}$$

In a nutshell, this denotes that if the ith element of $h(F)$ has value k, then k encounters have occurred whose i-long initial portion matched the respective prefix of the trajectory $T(F)[1{:}i]$.

The relevance of this constraint is that it can be used directly to derive a frequency for the occurrence of message sequences that match the respective message patterns. If we write $\Pr(w'|w)$ for the frequency with which an encounter whose prefix matched w will be continued with a sequence that matches w', we can use the following statement (which follows directly from the constraints above):

$$T(F) = ww'w'' \Rightarrow \Pr(w'|w) = \frac{h(F)[|ww'|]}{h(F)[|w|]} \tag{4.3}$$

for $w, w'' \in \mathcal{M}^*$ and $w' \in \mathcal{M}^+$.

Although these retrospective semantics are of rather theoretical importance, they provide the background for a definition of *prospective* semantics. It is this prospective semantics that are used to make predictions regarding future interactions.

4.2.5 Prospective semantics

Unlike retrospective semantics which only reflect the frequency of observed cases, a prospective semantics should fulfil two requirements:

1. It should express that the probability with which an observed encounter is expected to occur again is proportional to its frequency in the past.
2. On the grounds of past observations, it should allow for computing a probability for message sequences that have not occurred before.

The first requirement could be achieved by simply replacing frequencies Pr by probabilities P in equation 4.3. However, this would preclude any ability to generalise, since the probability of a message sequence never experienced before would be zero.

Therefore, we introduce a real-valued *similarity measure*[8] $\sigma(w, w')$ on message patterns (and sequences thereof) that adds a *case-based reasoning* (Kolodner 1993, Aamodt and Plaza 1994, Watson and Marir 1994) flavour to frames. Given a potentially non-ground frame trajectory T, the purpose of this similarity measure is to express to which degree a concrete instance of T is comparable to previously experienced cases as described by Θ.

Looking back at the example of page 100, we might for example want to compare the sequence

$$\begin{aligned}&\texttt{propose}\big(\texttt{agent\_3}, \texttt{agent\_2}, \texttt{do}(\texttt{agent\_3}, \texttt{deliver\_goods})\big)\\ &\qquad\qquad \to \texttt{accept}\big(\texttt{agent\_2}, \texttt{agent\_3}, \texttt{do}(\texttt{agent\_3}, \texttt{deliver\_goods})\big)\\ &\qquad\qquad\qquad\qquad\qquad\qquad \to \texttt{do}(\texttt{agent\_3}, \texttt{deliver\_goods})\end{aligned}$$

to the previously stored cases for trajectory

$$\texttt{propose}(A_1, A_2, \texttt{do}(A_1, X_1)) \to \texttt{accept}(A_2, A_1, \texttt{do}(A_1, X_1)) \to \texttt{do}(A_1, X_1)$$

[8] For the moment, we will not define σ formally, as it is considered a domain-dependent part of the model and simply assume it has been specified. An exemplary definition is provided in section 5.3.3.

given by the substitutions

$$\langle[A_1/\texttt{agent\_1}],[A_2/\texttt{agent\_2}]\rangle$$
$$\langle[A_1/\texttt{agent\_3}],[A_2/\texttt{agent\_1}],[X_1/\texttt{deliver\_goods}]\rangle$$

stored in the frame. The idea behind this reasoning by analogy is to represent the new instance as a substitution

$$\langle[A_1/\texttt{agent\_3}],[A_2/\texttt{agent\_2}],[X_1/\texttt{deliver\_goods}]\rangle$$

and to assume that its probability is proportional to its relative similarity to the previous cases.

Obviously, the similarity to each of the previous substitutions $\Theta[i]$ will have to be weighted with its frequency $h_\Theta[i]$. Also, the computation should take into account which aspects of the respective condition $C[i]$ are relevant in the current knowledge state of the agent. This yields the following formula for computing the similarity of a substitution to an *entire* frame:

$$\sigma(\vartheta,F)=\sum_{i=1}^{|\Theta(F)|}\overbrace{\sigma(T(F)\vartheta,T(F)\Theta(F)[i])}^{\text{similarity}}\ \overbrace{h_\Theta(F)[i]}^{\text{frequency}}\ \overbrace{c_i(F,\vartheta,KB)}^{\text{relevance}} \tag{4.4}$$

A function c_i is used to assign different weights to substitutions the respective conditions of which do or do not hold under current knowledge base contents ("relevance" of substitution ϑ in the above equation). A simple method of defining c_i is

$$c_i(F,\vartheta,KB)=\begin{cases}1 & \text{if } KB\models C(F)[i]\Theta(F)[i]\vartheta\\ 0 & \text{otherwise,}\end{cases}$$

so that σ only considers cases that took place under comparable circumstances.

Equation 4.4 can easily be turned into a probability distribution for any ϑ to occur if F takes place:

$$P(\vartheta|F)=\begin{cases}\alpha\cdot\sigma(\vartheta,F) & \text{if } \vartheta\in\Theta_{poss}(F,KB,\varepsilon)\\ 0 & \text{otherwise}\end{cases} \tag{4.5}$$

for a normalisation constant α.[9] Thus, if a message sequence w occurs that matches the trajectory of F, then the probability of $w=T(F)\vartheta$ is given by the similarity of ϑ to the past cases stored in F weighted by their frequencies h_Θ. This quantity is normalised over all other substitution lists that F could possibly be enacted given the current knowledge base.

We can now turn to the central construction for determining the probabilistic semantics of communication in m²InFFrA, which is based on computing the probability of any message sequence $P(w)$ by reasoning about its similarity to existing frames.

[9] We shall frequently make use of such constants for reasons of readability. In equation 4.5, for example

$$\alpha=\frac{1}{\sum_{\chi\in\Theta_{poss}(F,KB,\varepsilon)}\sigma(\chi,F)}$$

and the denominator sum is non-zero, as long as we assign some non-zero value to $\sigma(w,w)$ (which is reasonable to assume); Θ_{poss} is finite by definition (cf. p. 101).

Rather than knowing how probable ϑ is, we would like to know the probability of particular message sequences, if $\mathcal{F}$ is to provide any concrete guidance. With equation 4.5, we are able to postulate the core element of m²InFFrA semantics:

$$P(w) = \sum_{F \in \mathcal{F}, w = T(F)\vartheta} P(\vartheta|F)P(F), \quad w \in \mathcal{M}_c^* \tag{4.6}$$

where $P(F) = \frac{h(F)[|T(F)|]}{|\mathcal{E}|}$ is the probability with which a frame has matched any past encounter. What this formula suggests is that the probability of a message sequence w can be obtained by multiplying the probability with which a substitution ϑ may occur if frame F is executed with the prior probability of F. Thereby, ϑ is the substitution that turns $T(F)$ into w (if the trajectory and w match at all), and $P(\vartheta|F)$ is given by equation 4.5. Note that this equation assigns a probability of zero to the occurrence of any message sequence that is not covered by a frame in $\mathcal{F}$.

To summarise, the semantics of a frame lie in probabilistic expectations regarding encounter sequences to occur. They are derived from statistical observation while assuming that the probability of future message sequences is proportional to their similarity with past cases.

4.2.6 Framing in m²inffra

So far, we have not explained how m²InFFrA frames fit into a definition of m²InFFrA agents. In the subsequent paragraphs, we will state more precisely what "agent" means in the context of m²InFFrA, and how these agents employ frames to compute communicative expectations.

Framing agents

We have already mentioned that m²InFFrA agents operate on the grounds of a frame repository in which they locally store their interaction experience. Apart from this repository, a framing agent is characterised by a knowledge base in which he stores his current beliefs about the world, and a utility function that provides a numerical estimate for each concrete message sequence depending on currently held beliefs. The following agent definition contains these elements, and it also introduces functions that transform frame repository and knowledge base after new encounters. Furthermore, it specifies a "communication horizon" used to make predictions in a boundedly rational fashion, and a similarity function that is used to derive probabilities for future message sequences.

Definition 4.4: An *agent* is a structure $a = (\mathcal{L}, \mathcal{M}, \mathcal{E}, u, f, \kappa, \sigma, H)$ where

- $\mathcal{L}$, $\mathcal{M}$ are formal languages used for logical expressions and message templates,
- $\mathcal{E}$ is the set of encounters perceived so far,
- $u : \mathcal{M}_c^* \times 2^{\mathcal{L}} \rightarrow \mathbb{R}$ is the agent's utility function estimate, where $u(w, KB)$ is the estimated utility of w being executed with initial knowledge base KB;

- $f : \Phi \times \mathcal{M}_c^* \to \Phi$ transforms a frame repository $\mathcal{F} \in \Phi$ to a new repository upon experience of an encounter with label $w \in \mathcal{M}_c^*$;[10]
- $\kappa : 2^{\mathcal{L}} \times \mathcal{M}_c^* \to 2^{\mathcal{L}}$ transforms knowledge base contents after an encounter accordingly;
- $\sigma : \mathcal{M}^* \times \mathcal{M}^* \to \mathbb{R}$ is a *similarity measure* for message pattern sequences;
- $H \in \mathbb{N}$ is a horizon such that the probability for the occurrence of each $w \in \mathcal{M}_c$ with $|w| \leq H$ is positive.

■

The functions u, f and κ will rarely be defined formally, but almost in all concrete implementations, they should fulfil the following requirements:

- The utility $u(w, KB)$ (we sometimes write $u(w)$ where KB is obvious from the context) is largely domain-dependent and it is in fact a rather strong assumption that the agent has an estimate for the usefulness of message sequences in every knowledge state. Usually, it is sufficient if this utility function only returns non-zero values for physical actions that reflect to which degree the respective action contributes to the achievement of (sub-social) agent goals. Generally speaking, the following guidelines should be followed:

  – Message sequences which involve more useful actions should receive a higher rating, e.g. by using (discounted) sums of the individual utilities of the elements of a sequence,

  – The fact that the state of the knowledge base is altered by execution of physical actions in a message sequence should be accounted for. For example, if a sequence contains two actions the latter of which will not executable after the former has been executed, the overall utility should be zero since the sequence cannot be executed as a whole.

  – A small negative utility should be assigned to (non-physical) messages to prevent overtly long communication sequences (and to express that communication overhead will eventually exceed any potential profit of physical actions).

  Note that u need *not* capture how useful communicative actions that occur on a sequence are in social terms, as this is the very purpose of the m²InFFrA decision-making process itself.

- The frame repositories generated by f should capture the entire history of interaction, unless bounded computational resources enforce restrictions on the number of

[10] Formally, a frame F with i trajectory steps is taken from the set of frames ϕ_i where

$$\phi_i = \underbrace{\mathcal{M}^i}_{\text{trajectory}} \times \underbrace{\bigcup_{j=0}^{\infty} \left(\underbrace{(2^{\mathcal{L}})^j \times SubstList^j}_{\text{condition sets/substitutions}} \right)}_{\text{past cases}} \times \underbrace{\mathbb{N}^i \times \mathbb{N}^j}_{\text{counters}}$$

which allows for storing an arbitrary (but equal) number j of condition sets and substitution lists. If $\phi = \cup_{i=1}^{\infty} \phi_i$ is the set of all arbitrary-length frames, then the set Φ of possible (finite) frame repositories is given by $\Phi = \{\mathcal{F} | \mathcal{F} \subset_{fin} \phi\}$.

frames that can be stored (in which case those frames that are least commonly used should be deleted first).

As m²InFFrA allows for varying degrees of abstraction in frames, the generalisation strategy used to manage the repository plays a decisive role when it comes to computing expectations. Basically, different strategies can range from simply storing each different encounter in a new frame with ground trajectory (or incrementing the occurrence counters of an existing ground frame) to total abstraction from agents and content (so that a new frame is only created if non-compliant performative sequences enforce a distinction). We will return to this issue in section 5.2.3.

Most other requirements for f result directly from the definition of frames in section 4.2.3 and from the retrospective semantics they are supposed to have according to the definitions of section 4.2.4.

- The knowledge base modifications brought about by κ are inherently domain-dependent, but may refer (for example) to social commitments, belief revision, etc.

  As a minimal requirement, this function should update the list of known $message(\cdot,\cdot,\cdot)$ facts upon termination of an encounter so as to include a new encounter that contains precisely the message sequence just perceived.

The framing state

With the above definitions, we are now able to define the *framing state* of an m²InFFrA agent as the probability distribution over future message sequences it can compute using the frames it maintains. More precisely, the framing state provides a "snapshot" of agent expectations while an encounter is unfolding, i.e. it returns a probability distribution over all possible encounter continuations under the encounter prefix that is being experienced.

Definition 4.5: Let $a = (\mathcal{L}, \mathcal{M}, \mathcal{E}, u, f, \kappa, \sigma, H)$ an agent. A *framing state* of agent a is a function $[a] : \Phi \times 2^{\mathcal{L}} \times \mathcal{M}_c^* \to \Delta(\mathcal{M}_c^*)$ which maps every

- frame repository $\mathcal{F} \in \Phi$,
- current *knowledge base* $KB \in 2^{\mathcal{L}}$, and
- current encounter prefix sequence $w \in \mathcal{M}_c^*$

to a finite-support probability distribution $P \in \Delta(\mathcal{M}_c^*)$ over possible *encounter continuations.* ■

This definition does not yet specify *how* the framing state is computed. This is important because the definition only makes minimal assumptions with respect to our outlook on consequentialist communication semantics: the use of frames as expectation encoding data structures, the conditioning of the framing state with knowledge base contents, and the restriction of the scope of prediction to encounter continuations. It should be stressed that all other elements that contribute to the computation of a concrete framing state are heuristics that are specific to an instance of the model.

So how can definite $[a]$ to allow for the computation of a framing state in accordance with the semantics of m²InFFrA frames? Essentially, this question can be reduced to finding a way to turn the declarative constraints of equations 4.1 (p. 102) and4.2 (p. 102) into an *operational* function that yields a real probability distribution. These constraints stated that m²InFFrA agents expect instances of a trajectory to occur with the same frequency in the future as they have occurred in the past. However, given an infinite number of choices in the way a trajectory pattern might be instantiated, an agent has to make choices relying on relevance rather than completeness. We will now present one possibility of defining the framing state to derive probabilities for encounter continuations based on using

- the current knowledge base contents,
- information about substitutions already applied during the current encounter,
- remaining degrees of freedom in substituting variables, and
- similarities between these substitutions and past cases stored in frames.

Returning to equation 4.6 which stated that

$$P(w) = \sum_{F \in \mathcal{F}, w = T(F)\vartheta} P(\vartheta|F)P(F)$$

we have to take several further steps to achieve a tractable definition of the framing state:

1. We have to assign zero probability to all substitutions ϑ that are not feasible anymore.
2. Rather than computing the probability of a sequence w, we need to specify how the likelihood of a continuation w' of an encounter prefix w can be computed.
3. Prefixes not covered by any frame should have a “don't know” semantics that makes every possible continuation equally probable.
4. Continuations over a certain length have to be disregarded to avoid computing a distribution with infinite support.

The first issue can be addressed by using Θ_{poss} to determine the substitutions that are still possible. At the same time, we can replace ε by w on the right hand side of equation 4.5 so that we obtain the quantity $P(\vartheta|F,w)$ as follows:

$$P(\vartheta|F,w) = \begin{cases} \frac{\sigma(\vartheta,F)}{\sum_{\chi \in \Theta_{poss}(F,KB,w)} \sigma(\chi,F)} & \text{if } \vartheta \in \Theta_{poss}(F,KB,w) \\ 0 & \text{otherwise} \end{cases} \tag{4.7}$$

Issue 2 is then easily resolved by turning equation 4.6 into

$$P'(w'|w) = \sum_{F \in \mathcal{F}, ww' = T(F)\vartheta} P(\vartheta|F,w)P(F|w) \tag{4.8}$$

for the probability of an encounter that started with w to be concluded with w'. This is in keeping with 4.6 since $P(w) = P'(w|\varepsilon)$.

Unfortunately, though, P' does not yield a real probability distribution, because the sum is zero if w is not captured by any frame (as described in item 3 above), since then $P(F|w) = 0$ for all F which results in $\sum_{w'} P(w'|w) = 0$. To overcome this problem, let

$$W(w) = \{w' \in \mathcal{M}_c^* | P(w'|w) > 0\}$$

be the set of continuations that are instances of the trajectory of some frame and are hence assigned a non-zero probability. Further, let $\epsilon \leq 1$ be the total (minute) probability that is to be assigned to all words not covered by any frame (item 4 in the list above). Also, let $\mathcal{M}_c^{\leq H} = \cup_{i \leq H} \mathcal{M}_c^i$ the set of concrete message sequences with a length up to H.

To turn $\overline{P}'$ into a real probability distribution, we will uniformly distribute ϵ among all continuations if the perceived encounter prefix is not covered by a frame. To ensure that probabilities add up to one, the probabilities for the remaining "defined" continuations will be weighted by the quantity $1/(1+\epsilon)$. This yields:

$$P(w'|w) = \begin{cases} \frac{1}{1+\epsilon} P'(w'|w) & \text{if } w' \in W(w) \\ \frac{\epsilon}{|\mathcal{M}_c^{\leq H} - W(w)|} & \text{if } w' \in (\mathcal{M}_c^{\leq H} - W(w)) \\ 0 & \text{otherwise} \end{cases} \tag{4.9}$$

Now we are able to define the a similarity-based formula for the computation of a framing state in m²InFFrA:

$$[a](\mathcal{F}, KB, w) = \lambda w'.\, P(w'|w) \tag{4.10}$$

The function returned by the framing state is a finite-support probability distribution that is only non-zero in arguments bounded in length by H. It exploits the experience stored in the frame repository and also includes reasoning by analogy by means of the similarity measure σ. For message sequences that are encountered for the first time, a small probability with uniform distribution is assigned to all possible continuations, since no further information is available in that case.

Finally, it should be remarked that this framing state definition yields a probability distribution that has support over concrete message sequences only (i.e. of the language $\mathcal{M}_c^*$ generated with starting symbol *Message* rather than *MsgPattern*, cf. table 4.1 on page 94). This is because the $can(\cdot,\cdot)$ predicate (cf. p. 101) is only true if the action can be executed right away, and for this fact we are able to directly assess the utility of all future sequences with non-zero probability.

4.3 Discussion

The definition of m²InFFrA agents and, in particular, the method suggested for computing the framing state raise a number of issues which we shall briefly discuss:

- *Can the continuation probability distribution be computed effectively?*

  Although the fact that w' ranges over $\mathcal{M}_c^{\leq H}$ (which is a very large set even if the number of objects and agents talked about is limited and if H is small) seems to suggest that the computation $P(w'|w)$ in equation 4.9 has to be performed for very many values, this is not the case. In fact, the number of values for which probabilities have to be computed using equation 4.8 is bounded by the number of the ground postfix sequences that can be derived from all frames under all possible substitutions, i.e.

  $$|\mathcal{F}| \cdot \max_{F \in \mathcal{F}} |\Theta_{poss}(F, KB, w)| \quad .$$

  The probability of all other sequences is uniformly distributed (with their probability being a fraction of ϵ).

$\Theta_{poss}(F, KB, w)$ is nothing but a formalised version of the content and context constraints discussed in section 4.1.5. Therefore, its size will largely depend on how an agent applies context and content restrictions when determining the set of possible message contents. For many application domains, it turns out that this number is manageable, as will be shown when we devise negotiation frames for the linkage scenario (section 6.3).

- *In equation 4.8, the probability of a continuation is affected by the number of (matching) frames. Can this be a problem?*

  Unfortunately, yes. Although we have precluded the possibility of frames with identical trajectories, we have not prohibited the use of frames with matching trajectories, such that the sets of concrete encounter sequences subsumed by different frames may overlap. Then, summing over multiple frames would increase the probability of these continuations.

  However, this case can only occur if one frame is an abstract version of the other, and the problem is partly alleviated by the fact that the cases stored in frames with more *general* trajectories will have *smaller* similarity values with a given potential conclusion, whereas the past encounters stored in frames with more *specific* trajectories will have *higher* similarity values with it.

- *Is the computation of continuation probabilities not highly sensitive to the definition of σ?*

  Yes, and defining good similarity functions is a major challenge in the development of m²InFFrA agents. Typically, rich definitions of σ will require the use of domain knowledge regarding the similarity of concepts agents use to reason about their environment.

  For this reason, the definition of σ is not part of "core" m²InFFrA. However, we provide a simple, domain-independent similarity measure that can be used as a starting point for more elaborate measures in section 5.3.3.

- *Why does the trajectory occurrence counter h not occur in the computation of probabilities?*

  Because, implicitly, by using equations 4.4 (p. 104) and 4.6 (p. 105), every previous encounter and substitution is counted. In equations 4.1 and 4.2, on the other hand, they were used directly, because substitution lists are only stored for encounters which included the *entire* trajectory. The formulation of the retrospective semantics was more natural when including the frequencies of prefixes of the trajectory.

4.4 Summary

To turn InFFrA into a concrete computational model, it is necessary to abandon the conceptual level, to come up with formal representations for the central InFFrA components and to define their formal semantics at least for a simple instance of the abstract framework. In this chapter, we have shown that to comply with interactionist assumptions and, in particular, to endow agents with models that are able to handle the dynamic evolution of "meaning", a new theory of communication is necessary. While this theory was only described at an abstract level, it was an important prerequisite for defining m^{2}InFFrA.

The introduction of this communication theory was followed by a detailed treatment of m^{2}InFFrA. This included defining languages for communication and logical reasoning, interaction frames, framing agents and framing states for m^{2}InFFrA agents. The result of these definitions is a reasonably simple model for encounter-based interaction reasoning that is based on probabilistic communication semantics and is reminiscent of case-based reasoning methods. In this model, the probability with which a message sequence will occur in the current interaction depends on its similarity to past cases as stored in the agent's frame repository.

Obviously, being able to compute probabilities for the way future interactions will turn out paves the way for the application of different kinds of decision-making and learning algorithms. This subject will be dealt with in the following chapter.

5. Learning and Decision-Making with Frames

The m²InFFrA formalisation provides a model for predicting the behaviour of agents in two-party conversational interactions. In this chapter, we will propose learning and decision-making procedures that exploit this model. While these procedures rest on established decision-theoretic principles, they are additionally characterised by the properties of *social abstraction* and *transient social optimality* which are specific to the social nature of learning and decision-making in InFFrA.

We will start by describing how the decision problem in m²InFFrA can be formalised as a two-level Markov Decision Process. In the subsequent section, we will suggest methods for learning in m²InFFrA. A final section deals with issues related to using the proposed algorithms in an integrated reasoning architecture. The chapter ends with a short summary.

5.1 Framing as a Two-Level Markov Decision Process

Equipped with the definition of framing states in m²InFFrA, we could easily formalise the decision problem in m²InFFrA as a problem of *adversarial expected utility maximisation*: The possible encounter continuations correspond to paths in a probabilistic game tree labelled with utilities and successive nodes correspond to the two players' moves. Traditional solution concepts from game theory (Fudenberg and Tirole 1991) and game-playing (Russell and Norvig 2003, chapter 6) could then be applied to find optimal solutions for this problem.

However, proceeding in this way would render the whole endeavour of using interaction frames as appropriate encodings of expectation structures useless. If we could have encoded the whole expectation structure as a probabilistic game tree in the first place, why would we have gone through all the tedious business of splitting the expectations into "bits" that make the computation of communicative expectations only more complex?

In section 2.2.2 we used Goffman's theory as the primary justification for developing InFFrA and m²InFFrA. It turns out that some aspects of this sociological theory translate fairly naturally into methodological implications for the design of computational models, i.e. there are practical arguments in favour of our approach beyond the sociological justification. More specifically, the sociological considerations result in two main requirements for social reasoning methods: *social abstraction* and *transient social optimality*.

5.1.1 Social abstraction and transient social optimality

The principle of *social abstraction* is fairly easy to explain starting from our remarks on generalisation in expectation structures (section 4.1) and from our outlook on interaction frames as defined in chapter 3. There, it was clear that frames are thought to abstract from particular situations so as to capture the central distinctions between *classes* of these.

But what are the reasons for abstracting from individual situations, interaction partners, actions and message contents in terms of a framework for social decision-making? After all, we want to build socially intelligent agents the performance of which will be measured in terms of their ability to make decisions that increase their long-term utility – the use of frames cannot be an end in itself. There are three principal arguments in support of such abstraction:

1. *The argument from pre-structuration of the interaction problem*

   This argument states that even though the theoretical possibilities for different interactions in open environments abound in theory (in principle, any agent may utter anything at any time), there is only a certain number of relevant categories of interaction that occur over and over again. These are determined by the action and reasoning capabilities of the agents in a society, by the distribution of resources in the environment and by the available communication channels.

   In an organisational setting, for example, the different types of relevant interactions might refer to task delegation, reporting, project discussions, advice seeking, etc., while advertisement of goods and services, negotiation, contracting, financial transfers and customer support will be among the typical interaction types in market-like environments.

   The key issue here is not that a given domain does not offer a great variety of possible interactions, but that for agents who reason strategically it makes sense to categorise them according to the joint courses of action that they achieve. It is these courses of (re)actions that are rather limited in most domains.

2. *The argument from bounded rationality, by which agents have no other choice than to generalise from particular interactions*

   It is not reasonable to assume that agents have arbitrarily complex reasoning capabilities to store all interaction experiences and to consider all of that information to act optimally in a new encounter. Faced with constraints regarding computational resources, generalisation through social abstraction is a reasonable strategy because it relies on storing those aspects of interaction that occur repeatedly. Usually, this is much more effective than implicitly preferring certain experiences over others by storing selected pieces of information and disregarding others.

3. *The argument from volatility in large-scale open agent societies, in which encounters with particular agents are only occasional*

   Even if computational resources were unlimited, it seems rather unlikely that detailed information about each and every past encounter could be re-used directly. This is because – especially in open systems – encounters with the same interaction partners and under the same circumstances are only occasional in the best case. In the worst, they are one-time experiences.

> Note that this does not necessarily imply that agents *have* to forget detail information about particular interaction situations if it is available. In m²InFFrA, for example, similarity considerations take previous experiences into account when comparable situations occur. It merely means that the interaction models stored by a social reasoning mechanism should not focus on individual situations in the first place.

Of course, social abstraction is also backed by the social theory of Erving Goffman. The very fact that framing is spawned by communicative symbols implies that their meaning must be taken to have an inter-subjective dimension for the framing agents (otherwise, each symbol could have an entirely different meaning for each different agent). Thus, as symbols are used across different situations and interacting people, they must be indicators for a whole class of individual situations. From the standpoint of cognitive processing, it is reasonable to assume that humans are capable of performing this kind of abstraction, otherwise they would never be able to reuse existing symbols in new situations.

The case for *transient social optimality* is somewhat harder to make. From the InFFrA architecture, it is obvious that optimal social decisions strongly rely on making the right assessments regarding the (i) validity, (ii) adequacy and (iii) desirability of a candidate frame at the right time. In m²InFFrA, these components translate to (i) matching between encounter sequence and trajectory pattern sequence (ii) fulfilment of frame conditions with respect to the current knowledge base, and (iii) utility of the (set of) continuation(s) suggested by the frame, respectively.

As has been mentioned before (cf. assumption 2.10, p. 28, and our remarks in section 4.1.7) there are two conflicting goals that determine the quality of a framing decision, namely (i) *predictability* and (ii) optimal *utility*. On the one hand, an InFFrA agent wants to be able to predict others' imminent actions. On the other hand, the agent cannot stick to a particular predictable pattern of interaction if this pattern is not optimal utility-wise. In terms of the framing process outlined above, this conflict arises when adequacy, validity and desirability measures in frame assessment yield contradictory results. More specifically, it occurs when certain continuations seem relatively certain but yield low or negative payoff in the m²InFFrA model.[1]

Of course, this problem could be alleviated by weighing these components or using appropriate thresholds, but the problem goes deeper, since the agent's own framing choice also affects the reactions of other parties. If we assume that other agents are at least as socially intelligent as the agent in question, they will also record interaction experience and apply it strategically. So if we *deviate* from a given established behavioural expectation (in the form of a "safe", well-known, stable frame) because its consequences are not desirable in the current state of affairs for the sake of "trying something new", it is very probable that we will not obtain optimal results. This is because peer agents will be confused and unable to figure out how the interaction will turn out.

As explained in section 4.1.6, deviance within the limits of defined alternatives (which correspond to different known frames in m²InFFrA) can be acceptable if one party still understands what the other is doing. Yet, explorative or radically new behaviour can be perceived as a negation of existing expectations. The agent confronted with such behaviour

[1] Note that the probability of a continuation also reflects the expectations directed towards oneself, either because of similar behaviour in the same role (initiator/responder) in the past, or because roles have been swapped and the reasoning agent is assuming a part that has expectations associated with it because of others' previous behaviour.

may even interpret such behaviour as rejection, which, in turn, may lead to open conflict. Therefore, depending on the frame knowledge of interacting parties (given that they may have different frame conceptions), the balance between conforming to well-defined "social procedures" and diverging from them is a very subtle one.

Transient social optimality is one answer to this problem. Essentially, it is based on occasionally neglecting promising alternatives for the sake of being "socially comprehensible" for others. In the framing process, this simply means that we trade desirability for validity and adequacy. Thus, the agent can hope to ensure predictability by sacrificing short-term utility, because it is better to have predictable opponents who may not act as nicely as one would wish, rather than constantly trying to make optimal moves while the other might apply the same kind of strategic reasoning.

Goffman, in fact, stresses the strategic aspect of interaction, but the level of strategic reasoning he alludes to is different from, say, the traditional decision-theoretic notion of "strategy optimisation". Instead of selecting particular actions in a utility-maximising fashion in each and every decision-making step, human actors rather *adopt* socially established procedures in a strategic fashion. This means that behaviour during interactions is only rarely optimised by an individual by completely deviating from expectations. However, choosing which of the different expectation patterns to activate is a highly strategic process in which agents should compute optimal strategies before taking action. In other words, the *procedure* is determined by the social context, but its *adoption* is a strategic choice the individual makes.

Social theory put aside, there is also a simple "statistical" reason for this kind of transiently optimal (or, alternatively, "occasionally sub-optimal") behaviour: If we assume that all agents generalise over their interaction experience, then a single agent is simply not in the position to modify their expectations within a single encounter. In other words, an individual won't make a difference, at least not in the short term. For this reason, optimality is often traded for predictability, especially if encounters are only occasional and volatile.[2]

5.1.2 The two-level MDP view of m^{2}inffra

On the grounds of this discussion, we can now explain the relationship between m^{2}InFFrA and classical *Markov Decision Processes* (MDPs). The basic idea is illustrated in figures 5.1 and 5.2. In the normal (single-level) MDP model, as shown in figure 5.1, the dynamic model of the world is characterised by state transitions which depend on the agent's actions (unless we are talking about passive MDPs in which transitions are observed only and no action has to be taken). In some or all of these states, the agent may receive numerical rewards (sometimes also called payoffs). Transitions between different states may be highly non-deterministic, but their probability distributions are *stationary*, i.e. they remain constant over time.

In m^{2}InFFrA, on the other hand, the decision problem is split into two sub-problems through hierarchical decomposition. By using frames as representations of a set of possible interaction sequences we can break down the decision problem in two levels as suggested by figure 5.2. On the "upper" *framing level*, the agent has to dispose of an optimal

[2] This is not to say that individual actions can never have a huge impact on expectations. In close human relationships with repeated and ongoing interactions, for example, the expectations built from experience with a particular person can be very different from frame conceptions constructed in everyday social interaction,

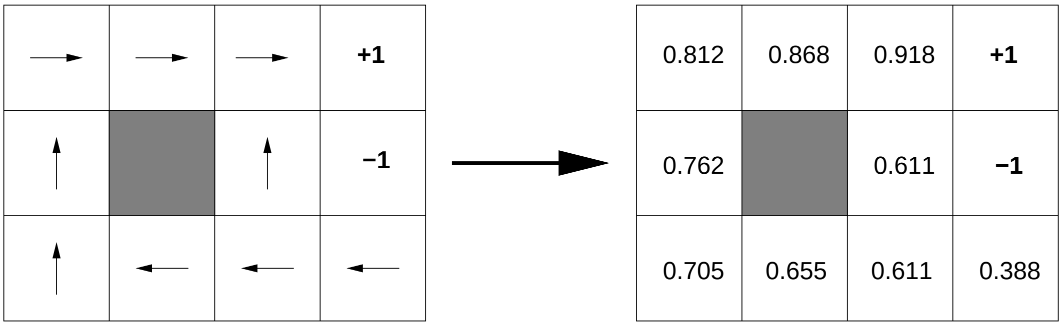


Fig. 5.1: Single-level MDP view: The agent perceives state transitions denoted by arrows that lead from an observed state/caret to a successor state, where some states may yield a (positive or negative) numerical reward (shown as a number in the respective caret). From the perceived sequences of states, actions and rewards, a utility function is computed that can be used to determine the optimal action in each state (right). Adapted from (Russell and Norvig 2003).

framing policy by which he can select the best frame in any given situation. The framing utility function that is used to make the right framing choices should take into account the long-term payoffs achieved by certain framing strategies. At the "lower" *action level* the agent should make optimal decisions about which action to take *within* the bounds of the currently active frame. Here, the utility of actions should correspond to the immediate payoffs achieved by applying a certain strategy. Alternatives suggested by other frames can be disregarded.

In this way, introducing interaction frames allows for coping with the vast state space of a general interaction decision problem (in the sense of an agent design problem in which we would have to determine a "life-long" policy that is optimal for the agent). Effectively, the huge single-level MDP of the global communication system (that corresponds to an unmanageable expectation network) is sub-divided into *two* smaller MDPs, which is also the reason for the name of the computational model of InFFrA: the "m^2" means nothing but "double-Markov".

Social abstraction is present in this model by virtue of reasoning by analogy – frame trajectory patterns implicitly represent large sets of instances, whose probabilities are derived using similarity measures. Transient social optimality is achieved by ignoring alternative frames while a particular frame is active – a portion of the search space is deliberately ignored for the sake of re-using established patterns of interaction.

After a brief introduction to MDPs, we shall explain how these considerations translate to the formal models developed in m^{2}InFFrA in the following paragraphs.

5.1.3 Markov decision processes

Formally,[3] a (discrete) *Markov Decision problem* is given by a finite set S of states, a finite set A of actions, a reward function $R : S \times A \to \mathbb{R}$ and a *transition probability function*

---

and may very well be seriously affected by a single action on either side.

[3] This section largely follows the introduction given in (Barto and Mahadevan 2003) and we also use the authors' notation.

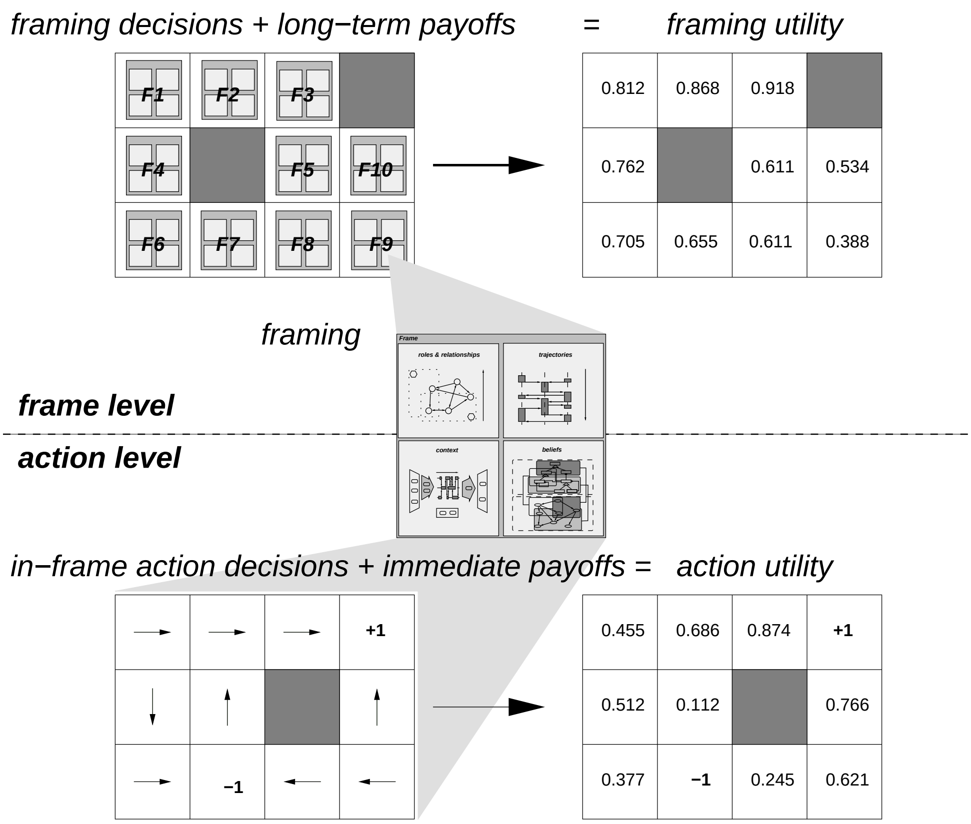


Fig. 5.2: Two-level MDP view of the framing process: The process of framing mediates between the framing-level MDP and the action-level MDP by restricting the set of applicable strategies to those sequences that are represented by the frame that is activated.

$P \in \Delta(S \times A \times S)$. The intuition is as follows: In a sequence of stages, an agent observes the current state $s \in \mathcal{S}$, executes an action $a \in A$ and receives an immediate payoff $R(s, a)$. With probability $P(s'|s, a)$, the next state the agent finds himself in will be s'. A (*stationary* and *stochastic*) *policy* is a mapping $\pi \in \Delta(S \times A)$ which specifies that the agent executes action a in state s with probability $\pi(s, a)$.

In a Markov decision problem, the goal of the agent is to determine an *optimal policy* π^* that maximises the long-term payoff of the agent. Following Puterman (1994), the definition of a *Markov decision process* additionally includes this optimality criterion, as "maximising long-term payoff" may have different interpretations.

A commonly used criterion is that of *infinite-horizon expected utility maximisation* where the payoff of state sequences is computed as the discounted infinite sum of individual rewards. According to this criterion, if $\gamma < 1$ is a discount factor, $E[\cdot]$ denotes the expected value, and r_t is the reward achieved at the t-th step by applying π, the *value function* V^π corresponding to π can be defined as this discounted infinite sum, such that

$$V^\pi(s) = E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| \pi, s_t = s\right]$$

is the value of state s. An optimal policy π^* is a policy that maximises V^π (which is then called the *optimal value function* V^*) in each state. *Action-value functions* that assign a value $Q^\pi(s,a)$ to each pair (s,a) by assuming that policy π will be followed after execution of a in s can be defined accordingly:

$$Q^\pi(s,a) = E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i} \middle| \pi, a_t = a, s_t = s\right]$$

Analogously, the optimal action-value function is denoted by Q^*.

Most solution methods for MDPs are based on dynamic programming techniques which exploit the fact that value functions satisfy the so-called *Bellman equations*

$$V^\pi(s) = \sum_{a\in A} \pi(s,a) \left[R(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V^\pi(s')\right] \tag{5.1}$$

and

$$V^*(s) = \max_{a\in A} \left[R(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)V^*(s')\right] \tag{5.2}$$

(similar equations can be derived for action-value functions).

It is common practice to approximate the *utility* of each state by a numerical function $U : S \to \mathbb{R}$ which can be used as a guide to choose the optimal action in any given state. In *value iteration*, for example, an approximation U_k of V^* is computed in the kth iteration of the algorithm, so that using

$$U_{k+1}(s) = \max_{a\in A} \left[R(s,a) + \gamma \sum_{s'\in S} P(s'|s,a)U_k(s)\right] \tag{5.3}$$

progressively transfers information regarding the approximate utility of successor states to their predecessors. It can be shown that starting from an arbitrary function U_0, value iteration converges to the optimal value function, i.e. the sequence $\{U_k\}_{k\in\mathbb{N}}$ converges to V^*.

A variant of MDPs that is important to the application of MDP theory to $\mathsf{m^2InFFrA}$ is that of discrete-event *semi-Markov Decision Processes* (SMDPs). In SMDPs, decisions can only be made after certain integer multiples of an underlying time step, i.e. we are dealing with *temporally extended courses of action.* This is modelled by introducing a random variable τ for the *waiting time* that passes after a is executed in state s for the transition to the successor state s' to occur. Writing $P(s',\tau|s,a)$ for the joint probability that the transition will occur after τ time-steps, and assuming that $R(s,a)$ now denotes the reward accumulated during that time, the Bellman equations become

$$V^*(s) = \max_{a\in A} \left[R(s,a) + \sum_{s'\in S,\tau\in\mathbb{N}} \gamma^\tau P(s',\tau|s,a)V^*(s')\right] \tag{5.4}$$

and

$$Q^*(s,a) = R(s,a) + \sum_{s'\in S,\tau\in\mathbb{N}} \gamma^\tau P(s',\tau|s,a) \max_{a'\in A} Q^*(s',a') \quad . \tag{5.5}$$

As we shall shortly explain, this provides the foundation for developing *hierarchical reinforcement learning* methods (Barto and Mahadevan 2003), and we will use one such method to develop frame-learning algorithms for m²InFFrA.

5.1.4 Choosing among frames

So how does the MDP formalism relate to the m²InFFrA model? Obviously, each frame captures a number of policies that are applicable in certain situations. The framing state computed after perceiving the initial portion of an encounter can be used as an approximation of the transition probabilities between states, depending on the selected action. Rewards can be derived by using the utility function of the m²InFFrA agent definition.

Therefore, selecting a particular frame means restricting oneself to a particular subset of policies. To describe this "higher" level of the *framing* decision process within the MDP formalism, we need a formal framework that is powerful enough to capture this hierarchical view.

One such framework is the *options* approach proposed by Precup (2000) in her PhD thesis.[4] In this framework, agents can choose between different options $\langle \mathcal{I}, \pi, \beta \rangle$ where $\mathcal{I} \subseteq S$ is the so-called *initiation set*, π is the policy of the option, and $\beta : S \rightarrow [0;1]$ is a stochastic termination condition. The idea is that an option is available at time t if and only if $s_t \in \mathcal{I}$. If it is chosen, then a_{t+1} is selected according to π, and $\beta(s_{t+1})$ determines whether execution of the option is terminated (whereupon the agent gets to chose a new option).[5] We can define $\mathcal{O}_s = \{o = \langle \mathcal{I}, \pi, \beta \rangle | s \in \mathcal{I}\}$ as the set of options available in state $s \in S$, and the set of all options $\mathcal{O} = \cup_{s \in S} \mathcal{O}_s$ as the union of these state-specific option sets. This allows for defining *policies over options* $\mu : S \times \mathcal{O} \rightarrow [0;1]$, according to which the agent will choose an option o in some state s_t. Having chosen this option, he will behave according to the local policy π of the option until o terminates at s_{t+k}, whereupon a new option is selected according to μ.

Letting $\mathcal{E}(o, s, t)$ denote the event that o was initiated at time t in state s and $p^o(s', \tau)$ the probability that o terminates in s' after τ steps, we obtain

$$r_s^o = E\left[\sum_{i=0}^{\tau} \gamma^i r_{t+i} \middle| \mathcal{E}(o, s, t)\right]$$

for the reward accumulated during execution of o and

$$p_{ss'}^o = \sum_{\tau=1}^{\infty} p^o(s', \tau)\gamma^\tau$$

for the probability of each state transition. With this, we can specify the state-value function of any state under a policy μ over options:

$$V^\mu(s) = \sum_{o \in \mathcal{O}} \mu(s, o) \left[r_s^o + \sum_{s' \in S} p_{ss'}^o V^\mu(s')\right]$$

[4] A summary of the main concepts can be found in (Sutton, Precup and Singh 1999); the approach is compared to other hierarchical reinforcement learning frameworks in (Barto and Mahadevan 2003).

[5] Note that options can be seen as a generalisation of "primitive" MDP actions $a \in A$, since primitive actions can be (trivially) written as options $\langle S, \pi_a, \beta_1 \rangle$ where $\pi_a(a) = 1$ and $\forall a' \neq a. \pi_a(a') = 0$ is the deterministic policy that always selects a, and $\forall s. \beta_1(s) = 1$, i.e. the option always terminates on any subsequent state.

As action-value function, we obtain

$$Q^{\mu}(s,o) = r_s^o + \sum_{s'\in S} p_{ss'}^o \sum_{o'\in\mathcal{O}} \mu(s',o')Q^{\mu}(s',o')$$

in which $p_{ss'}^o$ corresponds to $\gamma^t P(s',\tau|s,a)$ in the SMDP Bellman equations 5.4 and 5.5.

To apply this model to m²InFFrA, we can interpret frames as options, by defining $\mathcal{O} = \{o_F\}_{F\in\mathcal{F}}$ and actions during execution of a frame as primitive actions. Since m²InFFrA agents have actions $m \in \mathcal{M}_c$ at their disposal, $A = \mathcal{M}_c$. For the moment, we shall not define precisely the state set we will use (but see section 5.3.2), and shall simply use an equivalence relation $S \subseteq \mathcal{M}_c^* \times 2^{\mathcal{L}}$ that splits all possible knowledge base contents and perceived encounter prefixes into equivalence classes $s_1,\ldots,s_n$ by which states are identified. We use $s(w,KB) = s_i$ to identify the state that corresponds to a certain combination of knowledge state KB and encounter prefix w. With this and equation 5.5, we can write

$$Q^*(s,F) = E[R(s,F)] + \sum_{\tau=1}^{\infty} \gamma^{\tau} \max_{F'\in\mathcal{F}} \sum_{s'\in S} M_{ss'}^{F,\tau} Q^*(s',F') \tag{5.6}$$

where

- $s = s(w,KB)$, KB and w describe knowledge base and encounter prefix when frame F was selected,
- $s' = s(ww',KB')$ and KB'/ww' describe the situation at which a new framing decision had to be made (such that $\tau = |w'|$),
- $M_{ss'}^{F,\tau}$ is the probability that the next re-framing will take place when s' has been reached after τ steps, and
- $R(s,F)$ is the utility accumulated between s and s'; in the simplest case[6], this can be written as $u(w',KB)$ using definition 4.4 (p. 105).

Exploiting the fact that the knowledge base transformation function κ is deterministic and writing $\delta(s,w')$ for the state $s(ww',\kappa(KB,w'))$ that results from executing w' in KB after prefix w, we can re-write the above equation as

$$Q^*(s,F) = E[R(s,F)] + \sum_{w'\in\mathcal{M}_c^*} \gamma^{|w'|} \max_{F'\in\mathcal{F}} M_{s\delta(s,w')}^{F} Q^*(\delta(s,w'),F') \tag{5.7}$$

At the "framing" level of decision-making, the *optimal frame* F^* can thus be selected among all frames that maximise Q^*:

$$F^*(w,KB) = \arg\max_{F\in\mathcal{F}} Q^*(s(w,KB),F) \tag{5.8}$$

[6] This depends on the desired granularity of reward accumulation, and on the nature of the utility function u. If u takes some form of discounting into account when computed for an entire sequence w', $u(w',KB)$ is precise enough. At the other end of the granularity spectrum, we might use $KB_i = \kappa(w'[1{:}i],KB)$ (cf. p. 105) to update knowledge base contents upon each individual step and let

$$u(w',KB) = \sum_{1\le i\le|w'|} \gamma^i u(w'_i,KB_i)$$

5.1.5 Intra-frame decision-making

At the "action" level of decision-making, where agents only consider the policies offered by a single frame, m²InFFrA semantics provide us with a similarity-based probability distribution over possible encounter continuations. To make up for disregarding all other frames while the perceived message sequence matches the active frame (which can lead to globally sub-optimal solutions with respect to the "core" MDP that describes the entire communication process), we conduct an *exhaustive* search over all alternatives at this level.

This decision process can be modelled as a tree where subsequent "moves" are taken in turns by the reasoning agent and his adversary (rather than a table of transition probabilities), because we do not generalise over different re-visited situations *during* execution of the active frame. In each step, the framing agent may choose from a number of possible substitutions (cf. definition 4.3, p. 101) that are admissible according to the trajectory of the active frame. Then, the adversary makes a move by selecting a substitution in turn, which may further restrict the set of substitutions available to the first agent. This process is repeated until either the frame terminates or the agent has to re-frame. Thus, an optimal *intra-frame* strategy is a strategy that maximises the expected utility of the remaining frame steps under the probability distribution computed using the consequentialist semantics of the frame.

Formally speaking, if, at any stage, ϑ_s is a candidate "own" substitution, the expected payoff of executing F under ϑ_s given the current encounter prefix w is given by

$$E[u(\vartheta_s, F, w, KB)] = \sum_{\vartheta_p} P(\vartheta_p|\vartheta_s, F, w)\ u\big(\mathit{postfix}(T(F)\vartheta, w\vartheta), KB\big)$$

where $P(\vartheta_p|\vartheta_s, F, w)$ is the probability with which the other agent will choose some substitution ϑ_p depending on the agent's own choice ϑ_s. Thereby, the $u(\cdot, KB)$ term on the right hand side is nothing but the utility of the remaining steps $\mathit{postfix}(T(F)\vartheta, w\vartheta)$ (cf. equation. 4.2.3, p. 101) under the combined substitution $\vartheta = \vartheta_{\mathit{fixed}}(F, w)\vartheta_s\vartheta_p$ obtained by concatenating $\vartheta_{\mathit{fixed}}(F, w)$, ϑ_s and ϑ_p (the remarks of footnote 6 regarding utility discounting apply accordingly).

To determine $P(\vartheta_p|\vartheta_s, F, w)$, we will use the similarity-based posterior probability of a substitution as defined in equation 4.7 (p. 108). The product rule for conditional probabilities with some additional background evidence E,

$$P(A \wedge B|E) = P(A|B, E) \cdot P(B|E)$$

allows us to write

$$P(\vartheta_p \wedge \vartheta_s|F, w) = P(\vartheta_p|\vartheta_s, F, w) \cdot P(\vartheta_s|F, w)$$

In the above equation, $\vartheta_p \wedge \vartheta_s$ denotes the event of the peer selecting ϑ_p while the reasoning agent chooses ϑ_s, so that the substitution applied to F is actually $\vartheta_{\mathit{fixed}}(F, w)\vartheta_s\vartheta_p$. Furthermore, the probability of choosing an "own" substitution ϑ_s is given by the sum of the probabilities for the occurrence of complete substitutions that ϑ_s is a part of, such that

$$P(\vartheta_p|\vartheta_s, F, w) = \frac{P(\vartheta_p \wedge \vartheta_s|F, w)}{P(\vartheta_s|F, w)} = \frac{P(\vartheta_{\mathit{fixed}}(F, w)\vartheta_s\vartheta_p|F, w)}{\sum_\vartheta P(\vartheta_{\mathit{fixed}}(F, w)\vartheta_s\vartheta|F, w))}$$

which nicely reflects the fact that the probability of a peer substitution is proportional to its relative similarity to a frame (with ϑ denoting the alternative choices of the peer that we normalise over).

Applying equation 4.7 to both numerator and denominator finally yields[7]

$$P(\vartheta_p|\vartheta_s, F, w) = \frac{\sigma(\vartheta_{\mathit{fixed}}(F,w)\vartheta_s\vartheta_p, F)}{\sum_\vartheta \sigma(\vartheta_{\mathit{fixed}}(F,w)\vartheta_s\vartheta, F)}$$

This provides us with a method to compute the optimal substitution

$$\vartheta^*(F, w, KB) = \arg \max_{\vartheta_s \in \Theta_{poss}(F,KB,w)} E[u(\vartheta_s, F, w, KB)], \tag{5.9}$$

so that the optimal next message m^* can be determined by applying ϑ^* to the next step of the frame:

$$m^*(F, w, KB) = T(F)[|w| + 1]\vartheta^*(F, w, KB) \tag{5.10}$$

Returning to the options framework, we can now describe the option $\langle \mathcal{I}_F, \pi_F, \beta_F \rangle$ induced by a frame F. However, as the strategy represented by the frame depends on the entire history and the current contents of the knowledge base, we cannot apply any generalisation into encounter states, and the state space during framing becomes $S = 2^{\mathcal{L}} \times \mathcal{M}_c^*$ at the intra-frame level.

- The initiation set is given by those states in which the encounter prefix w matches (an initial portion of) the trajectory of F:

$$\mathcal{I}_F = \{(w, KB)|\Theta_{poss}(F, w, KB) \neq \emptyset\}$$

- The policy π_F assigns probability one to the optimal next action as defined in 5.10 and probability zero to all other actions:

$$\pi_F((w, KB), m) = \begin{cases} 1 & \text{if } m = m^*(F, w, KB) \\ 0 & \text{else} \end{cases}$$

- The termination criterion β_π is deterministic. It prescribes termination of o_F in either of the following cases:
    1. The peer[8] executes an action that causes a matching failure between the encounter prefix and the trajectory of F.
    2. The set of $\Theta_{poss}(F, KB, w)$ may become empty due to changes in the knowledge base, so that the remaining trajectory steps are not executable under any substitution.
    3. The utility values of the remaining steps may change so that their execution does not seem advantageous anymore.

[7] To preclude division by zero and non-zero probabilities for ϑ_p that are not in Θ_{poss}, we would actually have to set this probability function to zero if $\vartheta_p \notin \Theta_{poss}(F, KB, w)$ or $\exists \vartheta . \vartheta_{\mathit{fixed}}(F, w)\vartheta_s\vartheta \in \Theta_{poss}(F, KB, w) \quad \wedge \vartheta_{\mathit{fixed}}(F, w)\vartheta_s\vartheta_p \notin \Theta_{poss}(F, KB, w)$. We omit these details for reasons of readability.

[8] The reasoning m²InFFrA agent always "obeys" the frame during execution, for which reason a trajectory mismatch can only occur due to deviance on the adversary's side.

Trivially, the criterion is also met if there are no more steps left to execute and the trajectory is completed normally. All this taken together yields

$$\beta_F(w, KB) = \begin{cases} 1 & \text{if } \neg unify(T(F)[1{:}|w|], w) & \textit{(validity)} \\ & \vee \Theta_{poss}(F, KB, w) = \emptyset & \textit{(adequacy)} \\ & \vee u(\vartheta^*, F, w, KB) < b & \textit{(desirability)} \\ & \vee |T(F)| = |w| & \textit{(completion)} \\ 0 & \text{else} \end{cases} \tag{5.11}$$

for some desirability bound $b \in \mathbb{R}$.

The three[9] constraints of the termination criterion reflect the InFFrA notions of *validity*, *adequacy* and *desirability* respectively, as discussed in section 3.4.3.

5.2 Learning to Frame

In the previous section, we have formalised strategic decision-making in m²InFFrA using a hierarchical two-layer model of MDPs. At the intra-frame level of decision-making, the action selection mechanism can be implemented right away. For the framing level of decision-making, however, the Bellman equation in 5.6 (or, alternatively, equation 5.7) only provides a constraint for the true Q^* if this function is known to the agent.

In this section, we will describe how Q^* can be learned from experience using reinforcement learning methods. Also, as m²InFFrA is based on deriving frame models from experience, we have to explain how these frames are constructed and the frame repository is managed in the long term when new encounters are observed.

5.2.1 Learning frame transitions

To approximate the optimal framing-value function (i.e. option value for the framing MDP) we use a variation of the update rule for Q-learning (Watkins and Dayan 1992) in SMDPs proposed by Bradtke and Duff (1995)

$$Q_{k+1}(s, F) \leftarrow (1 - \alpha_k) Q_k(s, F) + \alpha_k \left[\hat{R}(s, F) + \gamma^\tau \max_{F' \in \mathcal{F}} Q_k(s', F') \right] \tag{5.12}$$

where

$$\hat{R}(s, F) = \sum_{i=0}^{\tau-1} \gamma^i \frac{R(s, F)}{\tau - 1} = \frac{\gamma^{\tau-1}}{\gamma - 1} \cdot \frac{R(s, F)}{\tau - 1} \tag{5.13}$$

is the discounted reward that has been accumulated in steps $t+1, \ldots t + (\tau - 1)$ averaged over the $\tau - 1$ individual steps. Averaging is a necessary approximation to the individual payoffs received in each step, since the total reward is only sampled after the next framing transition.

To ensure convergence of the Q-values using the above update rule, the learning rate α_k that is used to weigh the importance of the current framing value $Q_k(s, F)$ against the

[9] Actually, a fourth case may occur in which Θ_{poss} is non-empty, yet the optimal substitution ϑ^* is no more available due to the adversary's choice. In this case, we will assume that m²InFFrA agents do not revise their framing choice but merely determine a new ϑ^* in the next decision stage.

quantity propagated from the experienced transition has to decay over time. A commonly used function for such a learning rate is

$$\alpha_k = \frac{1}{visits_k(s, F)}, \tag{5.14}$$

where $visits_k(s, F)$ is the number of times F has been activated in state s. Parr (1998) has shown that Q-learning using the update rule 5.12 converges to Q^* with a probability of 1 if every action is executed in every state infinitely often.

However, ensuring convergence requires that the agent not simply maximises Q in each step, but that he applies a suitable exploration strategy, such as *Boltzmann exploration* (Mitchell 1997) which uses a temperature function T_k that decays over time and determines the probability $P(s, F)$ of choosing frame F in state s as

$$P(s, F) = \frac{e^{Q_k(s,F)/T_k}}{\sum\limits_{F' \in \mathcal{F}} e^{Q_k(s,F')/T_k}} \quad . \tag{5.15}$$

To sum up, the top-level framing process proceeds as follows. In the first iteration, $Q(s, F)$ is initialised to zero for each s and F, T_0 is set to an initial "temperature". After perceiving a state s, the agent selects a frame probabilistically according to equation 5.15. Frame execution proceeds according to the intra-frame action selection mechanism for the currently active frame F until $\beta(s', F) = 1$ for some s'. Using the discounted accumulated reward $\hat{R}(s, F)$ as defined in 5.13, the Q-values are updated taking the current learning rate α_k into account, and k is incremented. After that, the running reward $R(s, F)$ is reset to zero and the process is repeated iteratively.

Two final remarks should be made regarding the above procedure: Firstly, as the action-level decision-making process operates on complete knowledge-state encounter-prefix state descriptions (w, KB), the respective framing state $s(w, KB)$ has to be determined after each framing state transition. Secondly, frame termination may occur because the entire frame trajectory has been executed, and not only because of validity, adequacy or desirability problems. In this case, a transition to a new frame marks the start of a new encounter, and thus the Q-learning algorithm allows for learning useful strategies for consecutive encounters. After sufficient experience, the table may therefore also implicitly contain valuable information regarding meta-framing strategies.

5.2.2 Frame construction

When attempting to find a suitable frame in an ongoing encounter, it is quite probable that none of the frames in $\mathcal{F}$ matches the current encounter prefix. According to the framework presented so far, this would result in encounter termination. However, it may also be the case that applying a combination of existing frames is more adequate than ending the current encounter and starting a new one after further re-framing.

As an example, consider a conversation in a room where one of the participants suddenly stands up and leaves the room. To prevent disruption of the conversation, the other participant(s) might follow the fugitive and continue the discussion while they are walking. Naturally, it makes more sense to think of this unexpected continuation as an improvised modification to the existing conversation frame rather than a new encounter, especially because it is probably semantically linked to the previously ongoing discussion (in terms of subject, opinions, etc.).

This example illustrates that continuing an encounter (which has either ended or does not match the active frame F) with the trajectories of other frames $F'_1, \ldots, F'_k \subseteq \mathcal{F}$ is suitable whenever the original *goal* of the conversation cannot be achieved anymore by applying F, but can still be reached by executing $F'_1, \ldots, F'_k$ in the current state. What is important, is that in this process of "conversation re-planning", a new frame F' is created that combines the trajectories of F and $F'_1, \ldots, F'_k$. If the resulting frame is activated, it is stored in the long term, and the framing MDP utility function is updated according to the state transition that occurs between the state in which F' was activated and that in which it ends (or another re-framing becomes necessary). In other words, the agent "pretends" having disposed of F' from the beginning, and thus a new operator is introduced in the framing space that provides an alternative for reaching the same goal and lays the foundation for generating more complex frames out of simpler ones.

Formally, let wm be the perceived encounter prefix (where m is the last message that caused abandonment of the active frame F, i.e. $\mathit{unify}(w, T(F)[1{:}|w|]) \wedge \neg \mathit{unify}(wm, T(F)[1{:}|w|+1])$. Further, let s the state in which F was selected, and s'' the current state in the framing MDP (although prior to re-framing no state transition has occurred, we can apply $s(\cdot,\cdot)$ to the "full history" state (wm, KB) of the intra-frame MDP to obtain the respective equivalence class $s'' = s(wm, KB)$).

Let $F'_1, \ldots, F'_k \subseteq \mathcal{F}$ a set of frames and

$$T_i = T(F'_i)\vartheta$$

a ground instance of the trajectory of each frame F'_i that is obtained by binding the remaining variables by some substitution ϑ. We can define a new frame trajectory T' as the concatenation of the matching portion w of the encounter prefix wm and the T_i constructed above, i.e.

$$T' = w \cdot T_1 \cdots T_k$$

and require that $\beta(wm, KB) = 0$ (which implies that the resulting trajectory matches the current encounter prefix, that it can be executed and that it is desirable), we have achieved finding a continuation of the current encounter that is feasible and desirable under current conditions. We can now define

$$F' = (T', \langle\langle\rangle\rangle, \langle\emptyset\rangle, \vec{0}^{|T'|}, \langle 0\rangle)$$

as a provisional frame that will become part of the repository if activated (if it is aborted before its execution has been completed, obviously only a prefix portion of it will be stored).

An exhaustive search for such F' clearly has exponential time complexity, but a variety of heuristics can be thought of. For example, we can define the *goal* achieved by executing a trajectory T in knowledge base KB as the difference that T makes to the contents of the knowledge base in terms of facts that were not true before and have become true after executing T and vice versa:

$$\mathit{goal}(T, KB) = \{\varphi \in \kappa(KB, T) | KB \not\models \varphi\} \cup \{\varphi \in KB | \kappa(KB, T) \not\models \varphi\}$$

Of course, this set cannot be computed effectively for the general case, but since T is a simple sequence of physical actions and messages, it is easy to specify which preconditions will not hold anymore and which postconditions will be made true by consecutive execution of the steps in T (note that if T is not ground things might not be that easy due to the implicit quantification of variables).

Alternatively, we can restrict our search by considering only those frames(or ground instances thereof) that achieve the goal of the original frame F when concatenated, e.g. by conducting a forward-chaining search. By requiring that

$$goal(T(F), KB) = goal(T(F'), KB)$$

this would largely restrict the range of candidate frames F_i', because rather than merely assuring executability and desirability, we are actually looking for logical *alternatives* to F.

Returning to our example, following the participant that has left the room reflects that kind of alternative if one of the goals of the conversation is to get information from him. If letting him go and continuing the conversation without him seems desirable because the meeting can be used to discuss other matters, the original goal is abandoned, and any desirable continuation will be adopted – this corresponds to the exhaustive search method.

5.2.3 Frame generalisation

In principle, m²InFFrA imposes no restrictions regarding the level of abstraction at which frames are stored in the repository. As remarked on page 110, this implies that the sets of encounters represented by different frames may overlap, which may have a strong impact on the continuation probabilities derived from repository information.

Therefore, the choice of generalisation strategy is a crucial issue in the design of concrete m²InFFrA implementations: on the one hand, storing too many similar frames may increase the complexity of frame selection and result in disproportionately high probabilities for continuations suggested by more than one frame; on the other, coercing too many frames into a single, more abstract pattern may entail a matching behaviour that allows for too many potential trajectory instances which have actually never been experienced, thus blurring the semantics of the repository.

To define generalisation capabilities for m²InFFrA agents, two aspects have to be dealt with. Firstly, we have to define how frames can be merged to obtain a more abstract frame. In the second place, the generalisation strategies have to be discussed that determine when frames are merged.

Frame merging

In this section, we introduce a method for merging two frames into a more general frame that caters for the encounter instances captured by the original, more concrete frames. Once defined, this method can be applied to whole sets of frames by consecutively adding new frames to the result of previous merge operations.

Based on the above considerations regarding the choice of the right level of abstraction in frame management, we attempt a middle solution by introducing the notion of *most concrete common abstraction* (MCCA) $\Psi(m_1, m_2)$ of two message patterns $m_1, m_2 \in \mathcal{M}$, which is defined as that pattern $m \in \mathcal{M}$ that can be transformed into both m_1 and m_2 by application of "shallowest" substitutions:

$$\Psi(m_1, m_2) = m \Leftrightarrow \Big(\exists \vartheta_1 \vartheta_2 . m_1 = m\vartheta_1 \wedge m_2 = m\vartheta_2 \wedge \\ \big(\forall m' \vartheta_1' \vartheta_2' . m_1 = m'\vartheta_1 \wedge m_2 = m'\vartheta_2 \Rightarrow depth(m') \leq depth(m)\big)\Big)$$

where $\vartheta_1, \vartheta_2 \in SubstList, m' \in \mathcal{M}$ and the depth of a pattern is defined as the depth of argument nesting in an expression:

$$depth(m) = \begin{cases} 1 + \max_i depth(m_i) & \text{if } m = \langle m_1, \ldots, m_n \rangle \\ & \quad m = (m_1 \wedge m_2) \vee m = \neg m_1 \vee \\ & \quad m = (m_1 \Rightarrow m_2) \vee m = (m_1 \vee m_2) \vee \\ & \quad (m = h(m_1, \ldots, m_n) \wedge \\ & \quad h \in Head \cup Performative \cup \{\texttt{do}, \varepsilon\}) \\ 1 & \text{otherwise} \end{cases}$$

So Ψ is defined by looking at all patterns m' that subsume m_1 and m_2 and determining the pattern m out of these m' that is maximal in depth. In other words, the MCCA contains as few variables as necessary to subsume m_1 and m_2, namely exactly in those positions in which m_1 and m_2 differ. Assuming that

- the definition of Ψ is naturally extended to sequences of message patterns (using the "$m\langle m_1, \ldots, m_n \rangle$" case in the definition above)
- Ψ can be effectively computed (as described in (Fischer 2003)) and $\vartheta_{merge}(w, w') = unifier(w, \Psi(w, w'))$ is the substitution that is necessary to generate w from $\Psi(w, w')$,
- before merging two frames F and G we have first renamed their variables so that each variable appears only in one of the two frames,

we can now proceed to the definition of the frame $merge(F, G)$ that results from merging two frames F and G with $|T(F)| = |T(G)|$:

$$\begin{aligned} merge(F, G) = \; & \langle \Psi(F, G), \\ & C(F)\vartheta_{merge}(F, G) \cdot C(G)\vartheta_{merge}(G, F), \\ & \vartheta_{merge}(F, G)\Theta(F) \cdot \vartheta_{merge}(G, F)\Theta(G), \\ & hMax(h(F), h(G)), \\ & h_\Theta(F) \cdot h_\Theta(G) \rangle) \end{aligned}$$

where $\vartheta_{merge}(F, G)\Theta(F)$ is the result of "prepending" $\vartheta_{merge}(F, G)$ to each element of $\Theta(F)$.[10]

As for the trajectory occurrence counter of $merge(F, G)$, this is determined through defining a special operation $hMax(F, G) = \langle h_1, h_2, \ldots \rangle$ by which

$$h_i = \begin{cases} \max\left(\max(h(F)[i], h(G)[i]), \sum_k h_\Theta(merge(F, G))[k]\right) & \text{if } i = |T(F)| \\ \max\left(\max(h(F)[i], h(G)[i]), h_{i+1}\right) & \text{if } i < |T(F)| \end{cases}$$

[10] In (Fischer 2003), normal substitution concatenation is replaced by substitution application whenever redundant variables can be eliminated (e.g. the resulting frame would not contain substitutions of the form $\langle \ldots, [X/Y], [Y/Z], \ldots \rangle$.

The intuition behind this definition is as follows: When merging two frames, it is impossible to determine locally (based on F and G alone, that is) how many past encounter prefixes would have matched prefixes of the trajectory of $merge(F, G)$. A precise computation of $h(merge(F, G))$ would require comparison with all other frames in $\mathcal{F}$. However, deriving lower bounds for trajectory occurrence counter values is straightforward:

- For the last step of the trajectory, a lower bound is provided by the sum of the substitution counters of each frame (since stored substitutions refer to successfully completed frame executions).
- For all other steps h_i the subsequent value h_{i+1} provides a lower bound since there exist at least h_{i+1} past encounters that matched up to the ith step.

Finally, for any step, if the maximum of the respective elements of the trajectory occurrence counters of the two argument frames provides a tighter bound than the above, it is obviously preferred. Quite naturally, *merge* is undefined for any two frames that have a different length, i.e. $|T(F)| \neq |T(G)| \Rightarrow merge(F, G) = \perp$.

The example in table 5.1 taken from Fischer (2003) is useful to illustrate these rather complex definitions. It shows how two frames F_1 and F_2 can be merged into a frame F_3 that subsumes the original frames and covers as few additional encounters as possible.

Generalisation strategies

Given the above definition of a merging procedure for frames, it remains to specify when two (or more) frames will be generalised in the course of m^{2}InFFrA reasoning. As frame generalisation mechanisms are one of the main topics of (Fischer 2003), we shall restrict ourselves to a brief summary of the methods proposed therein.

Frame generalisation strategies have to be developed for two situations:

1. Addition of newly perceived frames: If an encounter is experienced that is not covered by any repository frame, the perceived frame is added to the repository. In this case, it is reasonable to look for possibilities of abstracting from this frame and other, similar repository frames so as to keep the repository concise. This is particularly important if computational resources prohibit arbitrarily-sized frame repositories.
2. Long-term repository management: Regardless of current or recent conversations, the agent can review the contents of the frame repository from time to time to evaluate whether the frame knowledge could be expressed more compactly without too great a loss of information.

Fischer (2003) uses the terms *online* and *offline* merging to refer to these cases, as the former occurs during the framing process when an encounter has just been completed, while the latter resembles a method for long-term repository optimisation and be applied at any point in time.

Fischer suggests generalisation heuristics based on cluster validation techniques (Jain and Dubes 1988). Each frame is interpreted as a set of points in the *encounter space* which correspond to the instances of concrete encounters it represents. On the grounds of this interpretation, it is possible to define a *compactness* measure for the cluster of points (or rather cluster of sets of points, as each stored previous case may represent a whole set of

$$
\begin{aligned}
F_1 = \Big\langle & \big\langle \xrightarrow{5} \texttt{request}(A, B, \texttt{inform}(B, A, \mathit{price}(Y, X))) \xrightarrow{2} \texttt{inform}(B, A, \mathit{price}(Y, Z)) \big\rangle, \\
& \big\langle \{\mathit{variable}(X), \mathit{car}(Y), \mathit{number}(Z)\}, \\
& \ \ \{\mathit{variable}(X), \mathit{car}(Y), \mathit{number}(Z)\} \big\rangle, \\
& \big\langle \xrightarrow{1} \langle [A/\texttt{a1}], [B/\texttt{a2}], [Y/\mathit{sedan}], [Z/10000] \rangle, \\
& \ \ \xrightarrow{1} \langle [A/\texttt{a1}], [B/\texttt{a3}], [Y/\mathit{sportsCar}], [Z/20000] \rangle \big\rangle \Big\rangle \\
F_2 = \Big\langle & \big\langle \xrightarrow{1} \texttt{request}(A, B, \texttt{inform}(B, A, \mathit{color}(Y, X))) \xrightarrow{1} \texttt{inform}(B, A, \mathit{color}(Y, Z)) \big\rangle, \\
& \big\langle \{\mathit{variable}(X), \mathit{car}(Y), \mathit{color}(Z)\}, \\
& \ \ \{\mathit{variable}(X), \mathit{car}(Y), \mathit{color}(Z)\} \big\rangle, \\
& \big\langle \xrightarrow{1} \langle [A/\texttt{a1}], [B/\texttt{a3}], [Y/\mathit{sportsCar}], [Z/\mathit{red}] \rangle \big\rangle \Big\rangle . \\
F_3 = \Big\langle & \big\langle \xrightarrow{5} \texttt{request}(A, B, V) \xrightarrow{3} \texttt{inform}(B, A, W) \big\rangle, \\
& \big\langle \{\mathit{variable}(X), \mathit{car}(Y), \mathit{color}(Z)\}, \\
& \ \ \{\mathit{variable}(X), \mathit{car}(Y), \mathit{color}(Z)\}, \\
& \ \ \{\mathit{variable}(X), \mathit{car}(Y), \mathit{color}(Z)\} \big\rangle, \\
& \big\langle \xrightarrow{1} \langle [A/\texttt{a1}], [B/\texttt{a2}], [V, \mathit{price}(Y, X)], [W, \mathit{price}(Y, Z)], [Y, \mathit{sedan}], [Z, 10000] \rangle, \\
& \ \ \xrightarrow{1} \langle [A/\texttt{a1}], [B/\texttt{a3}], [V, \mathit{price}(Y, X)], [W, \mathit{price}(Y, Z)], [Y, \mathit{sportsCar}], [Z, 20000] \rangle, \\
& \ \ \xrightarrow{1} \langle [A/\texttt{a1}], [B/\texttt{a2}], [V, \mathit{color}(Y, X)], [W, \mathit{color}(Y, Z)], [Y, \mathit{sportsCar}], [Z, \mathit{red}] \rangle \big\rangle \Big\rangle .
\end{aligned}
$$

Tab. 5.1: Merging two frames into one

instances, in turn) based on the pair-wise similarity values between the stored cases in the frame. Also, the average distance (i.e. inverse similarity) between the trajectory of the newly perceived frame F and the cases in a repository frame G can be used to measure cluster *isolation*. Combining these two measures allows for an assessment of the *validity* of a hypothetical frame (cluster) $merge(F, G)$. If this validity is sufficiently high, F and G are merged. As for the choice of G, that frame is chosen which would result in a maximally valid new cluster when merged with F.

Currently, this heuristics is only used for online merging, which is always performed when an encounter is over. Regardless of whether the encounter is an instance of a repository frame or not, we simply compare the perceived frame to all repository frames and merge it with the best match, if the resulting cluster validity is high enough.

This simple generalisation rule is also applied because it does not require modifications to the framing Q-value table, since, at most, a frame is replaced by a new one, but no two frames are merged into one. Note, however, that in the general case (especially in offline merging) such an update of the Q-value table is necessary (again, Fischer (2003) proposes methods to perform this update). Also, it has to be remarked that frame generalisation leads to substitution sets and occurrence counters that may violate the constraints 4.1 (p. 102) and 4.2 (p. 102) that were used to define retrospective frame semantics.

5.3 Implementation

In the presentation of the learning and decision-making algorithms used in m^{2}InFFrA so far, we have deliberately omitted certain aspects that require a more concrete specification to yield an implementable computational model. Partly this was done to highlight the core aspects of the developed formalism in a fairly generic fashion, but also to avoid too many implementation-specific details, many of which do not make up part of the proposed methods in a strict sense.

The following sections cover these implementation details. First, we show how the elements of m^{2}InFFrA laid out in the previous sections can be integrated to obtain a readily implementable procedural model of the entire framing cycle. Then, we explain the implemented framing state definitions and similarity measures, as the algorithms were only parametrised with them until now. Finally, we introduce heuristic methods for defining concrete frame desirability criteria that are in keeping with our theory of communication and the concept of transient social optimality.

5.3.1 The framing cycle in m^{2}inffra

The m^{2}InFFrA reasoning cycle is shown in figure 5.3. While referring to the framing phases summarised in figure 3.8 (p. 69), it makes the m^{2}InFFrA computations and their position in the overall InFFrA framing process explicit. To obtain the full picture of social reasoning in m^{2}InFFrA, we will now go through these phases one by one.

Situation Interpretation and Matching In every reasoning cycle, an encounter is either running or the InFFrA reasoner is in an idle state (see ❶ in figure 5.3).

In the former case, the processing cycle is spawned upon (a) reception of a message from a peer the agent is currently interacting with or (b) upon utterance of a message on the side of the reasoning agent himself. The new message m is appended to the trajectory of the perceived frame, and the knowledge base is updated to $\kappa(KB, m)$ to capture the effects of m. If m has utility effects, $u(KB, m)$ is added to the running reward $R(s, F)$ of the active frame F in state s for the accumulated reward to be available when updating the Q-table (see equations 5.6, 5.12 and 5.13, p. 121 and 124).

In the latter case, a decision has to be made regarding whether an encounter should be initiated or not. Although the reason for starting an encounter is usually provided by subsocial reasoning components of the agent, it is also possible to search the repository for frames with desirable post-conditions whose pre-conditions are currently met and simply start one of these frames. If the agent chooses not to start an encounter, the remaining reasoning cycle can be circumvented, and no action is executed.

As for matching (❷), m^{2}InFFrA uses a trivial difference model based on a clear-cut criterion regarding the matching status of the active frame compared to the perceived frame. However, what needs to be done prior to the assessment phase is to update the state description of the framing MDP $s(w, KB)$, at least if it includes information about the goal of

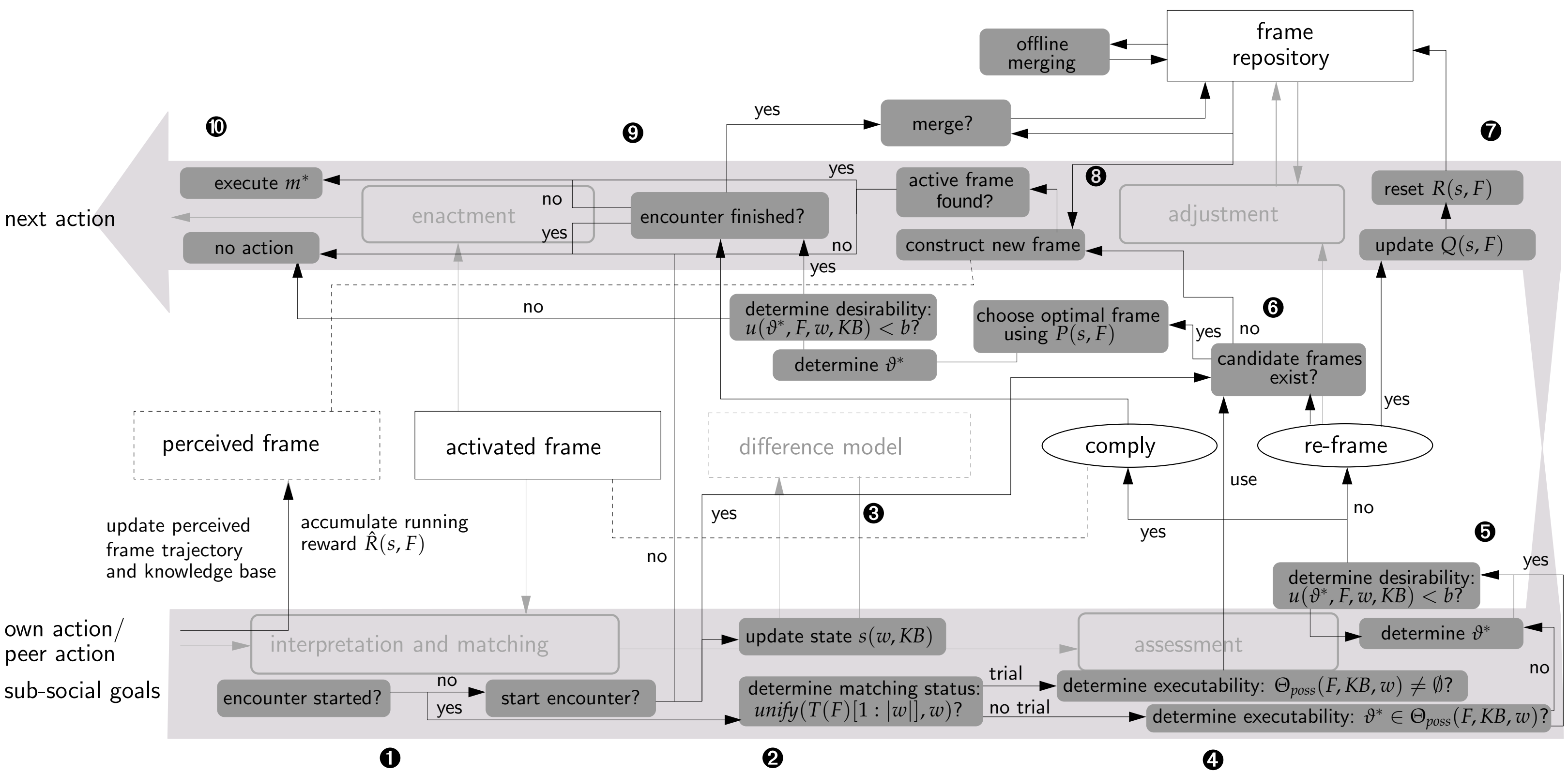


Fig. 5.3: The m²InFFrA framing cycle.

the current encounter, since this goal may have changed in the meantime due to InFFrA-external events (see also section 5.3.2).

If an encounter has just been started (❸), the agent has not selected an active frame yet. Therefore, assessment of the active frame is skipped and the agent acts as if in a re-framing situation (in other words, the process of selecting an initial frame at the beginning of a conversation is identical to that of choosing a new frame during an encounter). Else, the cycle proceeds with the assessment phase (❹).

Framing assessment Framing assessment consists of several steps that correspond to assessing the validity, adequacy and desirability of the active frame.

First, it is checked whether the (trial) active frame trajectory prefix-matches that of the perceived frame, i.e. if $unify(T(F)[1 : |w|], w)$ holds (❷). If this is not the case, the agent has to re-frame or to discard the current trial frame, if already in a re-framing procedure. The remaining assessment procedure distinguishes between two cases: (i) trial instantiation and (ii) "normal" framing (i.e. assessment of an active frame that has previously been selected).

In case (i), it first has to be verified that the remaining steps of a trial frame trajectory are executable given KB which amounts to $\Theta_{poss}(F, KB, w) \neq \emptyset$ (see equation 4.3, p. 101) because only then will it be considered as a candidate for activation. If the test succeeds, the frame is included in the set of candidate frames from which the new active frame is chosen stochastically according to $P(s, F)$ (equation 5.15, p. 125).

During "normal" framing (case (ii)), the agent has previously selected an optimal substitution (equation 5.9) under which the active frame is being executed. In this case, framing assessment is a bit more complex as it consists of two phases: First, it is checked whether $\vartheta^*(F, w, KB) \in \Theta_{poss}(F, KB, w)$, i.e. if the previously optimal substitution ϑ^* can be maintained. If F is no longer executable under ϑ^*, the second phase (❺) is entered, in which the agent iteratively searches for a substitution that is both executable ($\vartheta^*(F, w, KB) \in \Theta_{poss}(F, KB, w)$) and desirable ($u(\vartheta^*, F, w, KB) \geq b$). These steps ensure that action-level MDP optimisation is performed as described in equation 5.9 (p. 123). In real terms, this means that frame desirability depends on the profitability of the *best* conceivable variant of the active frame.

If all these constraints hold, the re-framing criterion $\beta_F(w, KB)$ is not met, and the agent can directly proceed with enactment (❾) of the active frame, i.e. with execution of m^* (equation 5.10). Else, the agent has to re-frame. Note that, obviously, the attempt to retain the frame also fails if $\Theta_{poss}(F, KB, w) = \emptyset$, since then no ϑ^* exists.

Frame adjustment In the event of re-framing, the Q-table has to be updated according to update rule 5.12 (p. 124) since a state transition has occurred. Also, the accumulated reward $R(s, F)$ has to be reset to zero, because a new frame has been chosen (see ❼).

Trial instantiation proceeds as follows: All frames that can be applied in the (new) state $s(w, KB)$ are checked for validity and adequacy (i.e. steps ❷ and ❹ are repeated for all of them), but desirability assessment is replaced by consultation of the Q-table (effectively, this is the point at which transient social optimality makes a real difference in terms of adaptation). This has a huge impact on the computational resources spent on framing assessment. Instead of searching the entire space of encounters covered by each candidate frame for an optimal substitution ϑ^*, it is merely checked whether there *exists* an executable substitution for each of them.[11] The stochastic frame selection criterion of equation 5.15 is used to choose the next active frame among all valid and adequate candidate frames, and before we proceed with enactment (❾), the optimal substitution ϑ^* has to be determined within this active frame. Again, a desirability check like that of phase ❺ is performed prior to actually selecting the respective frame.

The adjustment procedure described so far does not cater for the case in which no candidate frame is found (❻). In this case, the frame construction methods suggested in section 5.2.2 come into play. Effectively, this stage of the framing cycle is very similar to "planning" as it seeks to combine existing "operators" (that is, frames) to a message and action sequence that achieves a useful goal. If a usable frame can be constructed, it is activated. Since the methods for frame construction we have proposed only combine ground trajectory sequences, the constructed frame contains no variables and it is not necessary to determine an optimal substitution – the frame can be enacted "as is". If no frame is found at all, the whole re-framing process has failed and the agent can do nothing but terminate the conversation in the behaviour generation phase (❿). As for long-term repository management, the offline merging process described in section 5.2.3 runs concurrently with the main reasoning cycle and constantly optimises the repository. Also, after completion of a frame (❾), the perceived frame is added to the repository using online merging strategies.

Frame enactment Frame enactment in m²InFFrA is straightforward. If the frame is finished (cf. condition $|T(F)| = |w|$ in equation 5.11, p. 124), the agent simply terminates the encounter. The same reaction is spawned if framing fails and no active frame could be found. Of course, the agent might alternatively utter an *arbitrary* message in this situation, but we choose to discontinue the conversation simply because communication is not considered to be worthwhile if no reasonable pattern of messages can be identified that makes sense to the agent. Else, the next message is chosen according to equation 5.10 (p. 123), i.e. by applying the optimal substitution to the upcoming step in the trajectory of the active frame.

5.3.2 Encounter states

In section 5.1.4, it was claimed that an equivalence relation $S \subseteq 2^{\mathcal{L}} \times \mathcal{M}_c^*$ can be used to induce different equivalence classes $s_1, \ldots, s_n$ of knowledge base contents and perceived encounter prefixes that serve as the elements of the state space for the framing MDP. However, we have not yet proposed a method for determining appropriate state definitions in practice. Although we are far from disposing of a comprehensive theory of "state" in conversation processes (which would explain what leads an agent to prefer one frame over another), there are practical considerations that can be helpful in developing *encounter state* definitions.

Firstly, to prevent state-space explosion (a common problem in reinforcement learning), the state descriptions derived by the S function should yield a fairly small set of possible encounter states. Otherwise, the agent will need countless learning samples to converge to a reasonable framing strategy.

[11] In most practical situations, this is a constraint that can easily be verified. In a frame that describes an exchange of arguments, for example, it is met if there exist *any* concrete arguments that match the conversation pattern. Obviously, this is a question that is much easier to answer than that of finding an *optimal* argument.

Secondly, there should be an efficient method to infer the current state $s(KB, w)$ from KB and w. If, for example, the state depends on the satisfiability of logical statements that are hard to prove, the supposed complexity reduction of introducing the framing level will be sacrificed by adding computational complexity at the level of state determination.

Thirdly, states should be distinguished if differences between them *matter* to the applicability of a frame. For the Q-table to provide useful information, the entries for two frames F and F' should be different in a state s if one of the two frames is a good choice while the other is not. If all frames are considered almost equally good (low Q-value variance), then obviously determining the current state does not provide much guidance. A notion that appears to satisfy these requirements and that is in accordance with intuition from the world of human communication is that of conversation *theme* (we have already discussed the importance of theme to our theory of communication in section 4.1.5). Informally speaking, a very simple working definition for theme is "the sum of the agents, objects, actions, mental states and other concepts talked about in a conversation"[12]. From an expectation-based perspective, the theme is nothing but an alternative, often generalised description of the consequences of an interaction. For example, when A tries to convince B of his political views in a discussion, the theme "U.S. foreign policy" is nothing but a generalisation of expectations regarding the possible changes of B's belief related to the foreign policy of the U.S. (i.e. the potential outcomes of the conversations: B adopts A's view, B retains his own view, etc.).

In human communication, the theme (subject, topic) of a complex conversation is often a single concept, object or action, and it seems quite plausible that the "art" of framing lies in finding the appropriate abstraction of theme to base one's framing decisions on. Although these intuitions cannot be turned into a formal domain-independent model right away, it is usually possible to derive state definitions for a particular domain depending on the kinds of frames employed by thinking about the key attributes that framing success depends on.

In the link exchange domain, for example, if we picture frames that allow agents to negotiate over the execution of linkage actions, it seems reasonable to use an encounter state definition that contains information about

- the linkage actions that will be executed as a consequence of the encounter that can be compared to private agent goals (for example, if different frames are available for requesting link addition and link deletion, it is important for the agent to determine his current state by checking whether addition or deletion of a frame would serve his goals), and

- the role of the agent in the actions performed (it makes a big difference for the appropriateness of a frame whether the agent is the party executing the linkage actions or the party affected by them as a link "target").

In other words, identification of the current state should provide enough information to align current agent goals with the rewards that can be expected from activating a frame according to the Q-table. We will return to this issue in section 5.3.2.

[12] Where "talked about" does not necessarily mean "verbally" expressed: Sometimes communication is actually about what is being concealed while talking.

5.3.3 Similarity measure

Another issue that has not been dealt with concretely is that of defining an appropriate similarity measure $\sigma : \mathcal{M}^* \times \mathcal{M}^* \to \mathbb{R}$ as introduced in definition 4.4 (p. 105). By using σ to define the similarity $\sigma(\vartheta, F)$ of a substitution ϑ to an entire frame F (equation 4.4, p. 104), the definition of similarity largely influences the probability $P(\vartheta|F, w)$ with which a particular substitution ϑ will occur if a frame F is executed (equation 4.5, p. 104). We will now present a simple domain-independent definition of σ that has been proposed in (Fischer 2003). It based on recursively comparing arguments of operators while assigning a similarity of one to equal arguments and variable arguments, and zero similarity to all other elements.

Formally, for any two message patterns $m, m' \in \mathcal{M}$, $\sigma(m, m')$ is defined as follows:

$$\sigma(m,m') = \begin{cases} \frac{1}{arity(op(m))} \sum_{i=1}^{arity(op(m))} \sigma(arg(m,i), arg(m',i)) & \text{if } op(m) = op(m') \\ 1 & \text{if } m \in Var \vee m' \in Var \\ 0 & \text{else.} \end{cases}$$

Thereby, $op(m)$ returns the top-level logical operator symbol or function/predicate-/performative symbol of m, $arity(s)$ returns the integer arity of the respective operator[13]. Further, $arg(m,i)$ returns the ith argument of m and $Var = AgentVar \cup ContentVar \cup ObjectVar \cup MsgVar \cup PhysicalActVar$ is the set of all available variables.

By further defining

$$\sigma(w,w') = \begin{cases} \frac{1}{|w|} \sum_{i=1}^{|w|} \sigma(w[i], w'[i]) & \text{if } |w| = |w'| > 0 \\ 0 & \text{otherwise} \end{cases}$$

we can naturally extend σ to message pattern sequences $w, w' \in \mathcal{M}^*$. Note that dividing by $|w|$ restricts the range of σ to $[0:1]$, which is very useful to obtain a balanced weighting of past substitutions in equation 4.4 (p. 104). This provides us with a simple yet (as we will show in chapter 6) effective similarity measure that operates by pure syntactical comparison of message patterns and their sub-terms. Fischer (2003) has shown that a distance

[13] These auxiliary functions are defined as follows:

$$op(m) = \begin{cases} \neg & \text{if } m = \neg m_1 \\ \wedge & \text{if } m = m_1 \wedge m_2 \\ \vee & \text{if } m = m_1 \vee m_2 \\ h & \text{if } m = h(m_1, \ldots, m_i) \\ \perp & \text{otherwise.} \end{cases} \qquad arity(s) = \begin{cases} 1 & \text{if } s = \neg \\ 2 & \text{if } s = \wedge \text{ or } s = \vee \\ n & \text{if } s \in Statement \text{ is } n\text{-ary.} \\ 0 & \text{otherwise.} \end{cases}$$

$$arg(m,i) = \begin{cases} m_i & \text{if } (m = \neg m_1 \text{ or } m = m_1 \wedge m_2 \text{ or} \\ & \quad m = m_1 \vee m_2) \text{ and } i \leq arity(op(m)) \\ m_i & \text{if } m = h(m_1, \ldots, m_j) \text{ and } i \leq j \\ \perp & \text{otherwise.} \end{cases}$$

where m and m_i are arbitrary message patterns from $\mathcal{M}$, and $S(\mathcal{M})$ is the set of all function/predicate/-performative symbols in $\mathcal{M}$.

measure ρ defined as $\rho(w, w') = 1 - \sigma(w, w')$ constitutes a metric[14] and uses this observation to derive the heuristics for online and offline merging mentioned in section 5.2.3.

5.3.4 Desirability heuristics

The final issue that has to be discussed to enable a concrete implementation of m²InFFrA is a very decisive one: the definition of frame desirability criteria. While we were able to specify frame validity and adequacy in a completely domain-independent fashion in the re-framing criterion $\beta_F(w, KB)$ (equation 5.11), fulfilment of the desirability condition $u(\vartheta^*, F, w, KB) < b$ depends on the bound b. Unfortunately, no general statement can be made regarding the quantities (and the relationship between different utility outcomes) of u, so it seems that b will have to be determined individually for each specific application. However, for a wide range of applications we are able to propose a meta-heuristics for choosing the right value for b. This heuristics is based on our previous considerations regarding deviance, rejection and conflict (section 4.1.6) on the one hand and on communication and coordination (section 4.1.7) on the other. More specifically, it builds on the idea of balancing the *rejection* and *affirmation* character of any utterance.

As explained before, any utterance modifies an existing expectation structure, and agents are constantly caught in a dilemma regarding whether they should reinforce existing expectations to make communication more predictable or whether they should deviate from existing expectations to avoid low-utility actions. The latter option bears a risk, because a certain communication pattern can be expected to occur again in the future. Especially if roles are swapped, the agent might loose potential future profit because he is establishing a new expectation by his current behaviour.

We have pointed out in (Rovatsos et al. 2003a) that *entropy-based measures* are a useful means to capture the (un)certainty regarding expectation structures. Roughly speaking, they express how predictable or chaotic the effects of a particular utterance are in a given expectation network, both in terms of (i) contingency regarding potential outcomes (*expectation entropy*) and (ii) their utility range (*utility deviation*).

Message repository trees

To define these measures, we first need to transform m²InFFrA repositories into message pattern trees annotated with utilities. For the sake of simplicity, we shall only look at the performative sequences contained in repository frames, i.e. neglect message senders/receivers and content, the rationale being that such a repository tree should provide a very general view of the meaning of performatives.

The construction of such a tree $\mathcal{T}(\mathcal{F}) = (V, E)$ is fairly straightforward. Its nodes $v \in V$ represent performatives in trajectories and are labelled by virtue of a function $label : V \rightarrow Performative \cup \{\mathtt{do}, \triangleright, \perp\}$ (referring to the grammar of table 4.1, p. 94). Edges signify transitions between subsequent messages and are labelled with probabilities $prob : E \rightarrow [0 : 1]$. For every trajectory of a frame$F \in \mathcal{F}$, there exists an edge from the root node labelled with $\triangleright$ to a node labelled with the performative $T(F)[1]$. For every $1 \leq i \leq T(F)[|T(F)|] - 1$, if $T(F)[i{:}i+1] = \langle p(\ldots), q(\ldots)\rangle$ and $p, q \in Performative \cup \{\mathtt{do}\}$

[14] Actually, this observation is true of an entire family of distance measures of which this ρ is the most basic one.

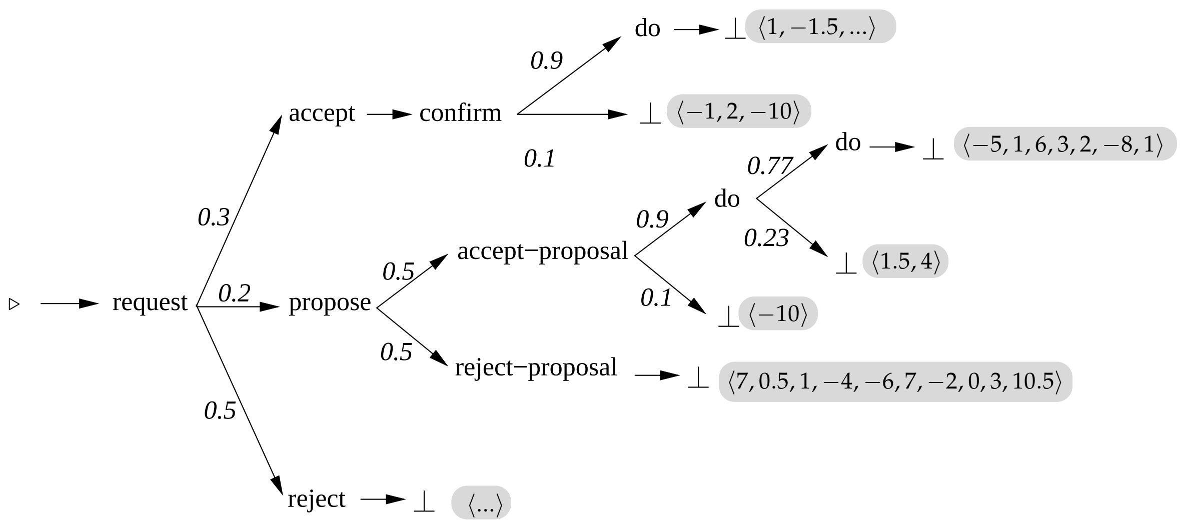


Fig. 5.4: Example frame repository tree. Edges are labelled with transition probabilities derived from observation. Leaf nodes are labelled with numerical utility lists U in shaded boxes. The label ▷ is used to identify the root node of the tree ("encounter start").

then $\mathcal{T}(\mathcal{F})$ contains an edge (v_p, v_q) with $label(v_p) = p$ and $label(v_q) = q$. For each $p(\ldots) = T(F)[|T(F)|]$, the tree contains an edge (v_p, v_f^F) with $label(v_p) = p$ to a leaf node $label(v_f^F) = \perp$ that indicates encounter termination. Leaf nodes labelled with $label(v) = \perp$ are additionally tagged with a list of $n(v) \in \mathbb{N}$ encounter utility values $U(v) \in \mathbb{R}^n$ which are derived from the total utility received after each of the $n(v)$ encounters that lead to v.[15]

As the example in figure 5.4 shows, if frames are consecutively added to such a tree starting with a tree that only consists of a root node labelled with ▷, this definition of $\mathcal{T}(\mathcal{F})$ enables us to build a tree in which

- nodes labelled with performative/`do` identifiers represent message performatives such that the ith step of a trajectory appears in the tree at depth i,
- edges represent transitions between subsequent messages labelled with transition probabilities, and the out-edges of a node branch whenever encounter prefixes can be continued with different performatives,
- leaf nodes labelled with the special symbol $\perp$ are used to denote encounter termination at the end of each path and are associated with a list U of numerical utilities that were experienced in the past whenever an encounter matched the performative sequence leading to the leaf node.

Entropy measures

Now we are ready to define entropy-based measures on such trees. Let $P(w)$ be the probability of a performative sequence $w \in (\mathit{Performative} \cup \{\texttt{do}\})^*$ that can be easily determined

[15] These definitions lead to a simplified version of the graph formalism suggested for communication systems in (Nickles and Rovatsos 2004).

by multiplying the probabilities along the corresponding path $path(w)$ in $\mathcal{T}(\mathcal{F})$. For a prefix label sequence w, we can define

$$EU[w] = \begin{cases} \frac{1}{n(v)} \sum_{i=1}^{n(v)} U(v)[i] & \text{if } v = last(path(w)) \text{is a leaf node} \\ \sum_{w' \in C(w)} P(w'|w) \cdot EU[ww'] & \text{else} \end{cases}$$

as the average utility to be obtained after prefix w. In this equation, $w' \in C(w)$ ranges over all performative postfix sequences covered by $\mathcal{T}(\mathcal{F})$ (i.e. $C(w) = \{w' \mid path(ww')$ exists and $last(path(ww'))$ is a leaf node$\}$).

With this, the *expectation entropy EE* and the *utility deviation UD* measures for an encounter prefix performative sequence w that has just been perceived can be computed as follows:

$$EE_{\mathcal{F}}(w) = \sum_{w' \in C(w)} -P(w'|w) \log_2 P(w'|w) \tag{5.16}$$

$$UD_{\mathcal{F}}(w) = \frac{1}{|C(w)|} \sqrt{\sum_{w' \in C(w)} (EU[ww'] - EU[w])^2} \tag{5.17}$$

The two measures can be combined to yield a total entropy $\mathcal{E}(w)$:

$$\mathcal{E}_{\mathcal{F}}(w) = EE_{\mathcal{F}}(w) \cdot UD_{\mathcal{F}}(w)$$

How can we interpret these measures? The expectation entropy assesses the information-theoretic value of having performed/perceived a certain sequence w of performatives. By computing the information value of all potential continuations, EE expresses the entropy that is induced by w in terms of potential continuations of this encounter prefix: The lower EE, the higher the value of w with respect to its ability to reduce the uncertainty of upcoming messages/actions. Thus, by comparing expectation entropies for different performatives in the process of selecting which message to utter, the agent can compare their values or regard the system of all possible messages as an "encoding" for future reactions.

Utility deviation, on the other hand, is defined as the standard deviation between the utilities of all possible continuations of the encounter given w. This allows for assessing the importance of the potential consequences of w. The power of this measure lies in being closely related to the expected utility of the encounter, while at the same time providing a measure for the *risk* associated with the performative sequence perceived so far.

By combining these two measures into $\mathcal{E}$, the agent can trade off the reduction of uncertainty against sustainment of autonomy depending on his willingness to conform with existing expectations or to deviate in order to pursue goals that contradict the expectations held towards him.

It has to be emphasised that while the tree view of the repository is based on (roughly) the same prediction mechanisms as the semantics of m²InFFrA frames described in section 4 it constitutes a rather crude simplification. Instead of predicting the probability of each actual continuation (i.e. a message/action sequence), it only makes very rough predictions regarding the occurrence of performatives within these continuations. Also, it does not take the dependence of continuation probabilities and utilities into account (which could be done by comparing substitutions and weighing the relevance of frame conditions.

Short name	*Path label*
"success":	`request` $\rightarrow$ `propose` $\rightarrow$ `accept-proposal` $\rightarrow$ `do` $\rightarrow$ `do` $\rightarrow \perp$
"A cheats":	`request` $\rightarrow$ `propose` $\rightarrow$ `accept-proposal` $\rightarrow$ `do` $\rightarrow \perp$
"B cheats":	`request` $\rightarrow$ `propose` $\rightarrow$ `accept-proposal` $\rightarrow \perp$
"rejection" :	`request` $\rightarrow$ `propose` $\rightarrow$ `accept-proposal` $\rightarrow \perp$

Tab. 5.2: Some interesting paths on the frame repository tree of figure 5.4.

Example

To illustrate the meaning of entropy measures, let us look at the example of figure 5.4 taken from (Rovatsos et al. 2003a). The tree shown in this figure summarises experiences with seven different performative sequences that have occurred so far in a total of one hundred encounters (with utility values different from those displayed in figure 5.4). The upper part of the tree represents a series of "request-accept-confirm-do" encounters, in which the "responder" agent executes a physical action (in 90% of all cases) that has been requested by the "initiator" agent after additional confirmation from the initiator. In the middle part, the responder may make a counter-proposal that obliges the initiator to also execute some physical action (hence the "do-do" sequence at the end of one path), while the initiator has the choice to accept or reject this counter-proposal. The lower part of the tree is concerned with direct rejection of the requested action, which is the most common type of encounter in this example.

In our discussion of this example, we shall concentrate on the "successful" path in the middle section of the tree, in which both agents (marked as A and B in the tables and plots below) execute the actions they have agreed to perform. Table 5.2 introduces names for some paths the entropy effects of which on the "success" path we shall analyse. This means that we are going to assess the different effects of executing one of these paths on the entropy values along the "success" path. of Apart from this successful path which indicates that both agents have "done their duty", the table lists paths for cases in which

- "A cheats" (by not doing his[16] part of the deal);
- "B cheats" likewise by not executing the first `do` action, and
- "rejection", in which the initiator does not accept the responder's counter-proposal.

The analysis will be performed using the *entropy change* $\Delta\mathcal{E}$ of w induced by execution of w'. It is defined as

$$\Delta\mathcal{E}_{\mathcal{F}}(w, w') = \mathcal{E}_{\mathcal{F}'}(w) - \mathcal{E}_{\mathcal{F}}(w) \tag{5.18}$$

where $\mathcal{F}'$ is the frame repository obtained from adding the new experience w to F (remembering, however, that the entropies are actually defined on the respective repository trees).

Figure 5.5 shows the values of $\Delta\mathcal{E}_{\mathcal{F}}(w, w')$ for continuations of all prefixes of the "success" path where w' ranges over all different paths in table 5.2. A first thing to note is the shape of the entropy curve which is typical of meaningful trajectories. As illustrated by the

[16] Note that due to argument omission in node labels, it is necessary to determine which party is executing which action by tracking the turn-taking procedure. In the above example, the first `do` action is performed by the responder, while the second one is executed (or not) by the initiator.

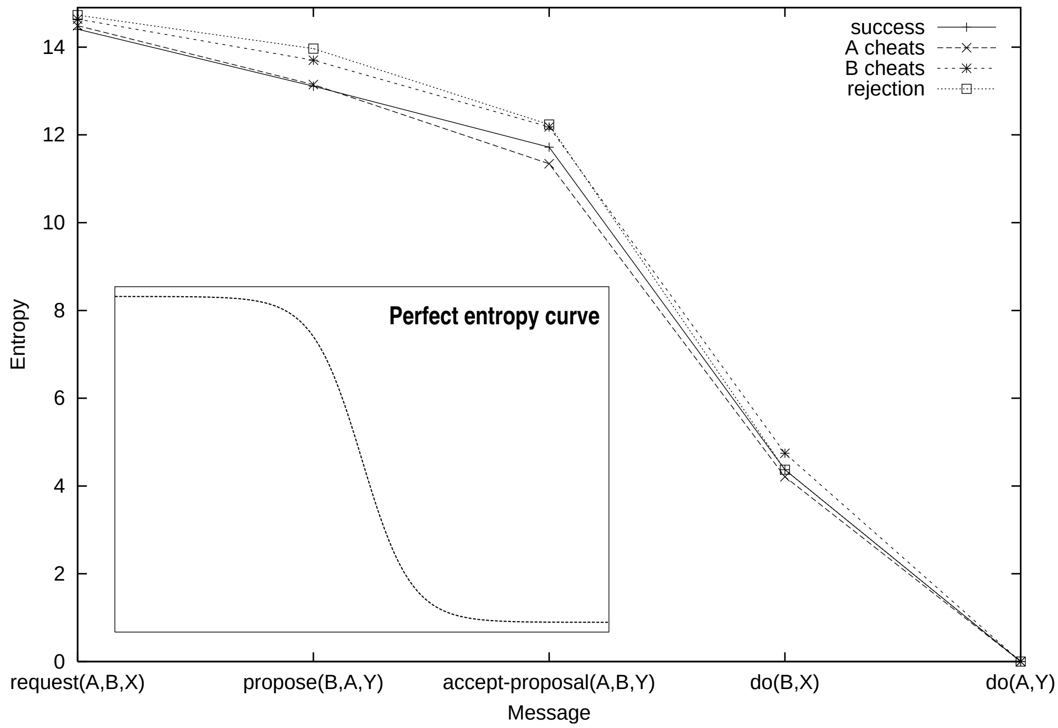


Fig. 5.5: Entropy effects of different trajectories on the “success” path and “perfect” entropy curve (boxed)

boxed “perfect” entropy curve, reasonable trajectories should start with an “autonomy” part with high entropy which gives agents several choices, and then continue with a “commitment” part in which entropy decreases rapidly to make sure there is little uncertainty about the consequences of the interaction further on. This is also necessary to make sure that communication pays off as the encounter is progressing and communication cost is increasing.

Secondly, cheating has a negative impact on entropies in the sense that the entropy values of “propose” and “accept proposal” exhibit a disproportionate growth. This is shown in detail in figure 5.6, which depicts the changes to node entropies before and after the respective interaction sequence. Furthermore, the effects of “A cheats” are much worse than those of “B cheats”, which conforms with our intuition that the closer utterances are to the final outcome of the encounter, the more critical will the expectations about them be. In the present example this happens because in the case of “A cheats” B has already invested in the interaction by performing some physical action in order to get something in return.

Thirdly, as before, the “rejection” dialogue and the “success” dialogue are acceptable in the sense of decreasing entropies of `propose` and `accept-proposal` (note that the small entropy increase of `request` is due to the 0.1/0.23 probabilities of cheating after `accept-proposal` and `do`(B, X)). The fact that “success” is even better than “rejection” suggests that, in a situation like this, there is considerable incentive to compromise, if the agent is willing to sacrifice current payoff for low future entropies.

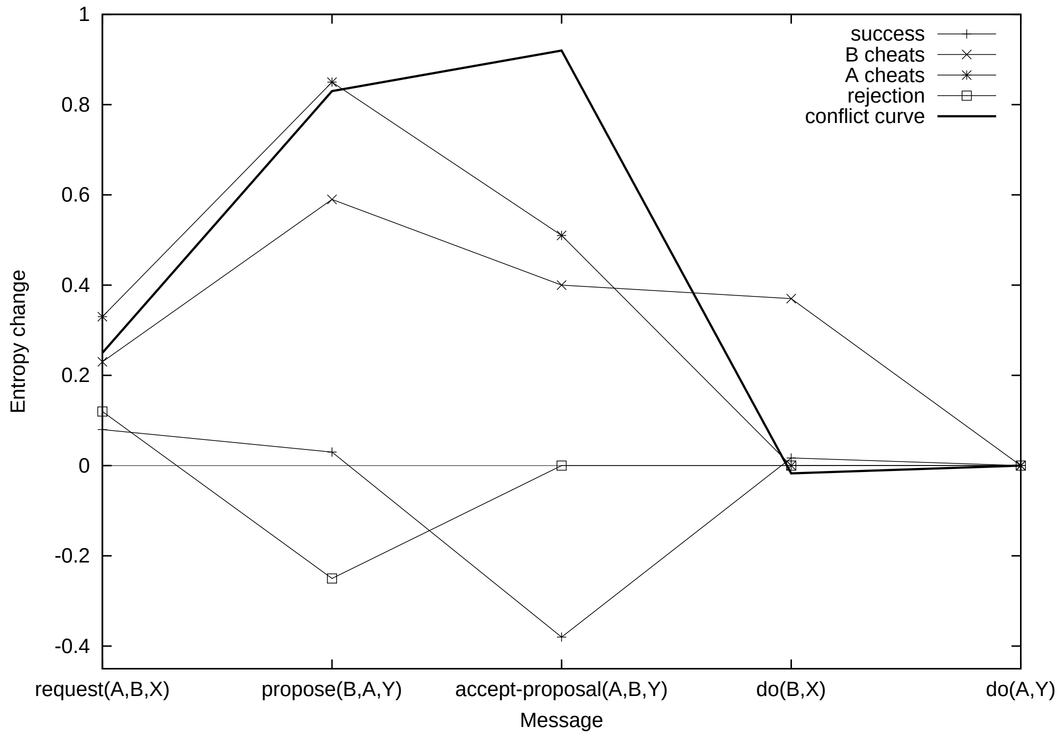


Fig. 5.6: Entropy changes to the "success" path and resulting conflict curve

Conflict potential

Looking at the plots in figure 5.6, a parallel becomes evident between trajectory entropies and reasoning about the long-term effects of a particular behaviour on the uncertainty associated with a frame repository.

Let $\mathcal{F}'$ be the result of adding a new encounter w' to the current repository $\mathcal{F}$. The entropy change $\Delta\mathcal{E}_{\mathcal{F}}(w, w')$ (equation 5.18) provides a measure of the *expectation-affirmative* or *expectation-negating* character of an utterance. In other words, it expresses to which degree the agents are saying "yes" or "no" to an existing expectation (see section 4.1.6).

The *conflict potential* of an encounter can be derived by comparing the *expected* entropy change to the entropy change that actually *occurred* due to the perceived trajectory. Analysing the difference between these two entropy changes reveals the degree to which the agents exceeded the expected change to expectation structures. We can define the conflict potential exerted on encounter w by the occurred encounter w'' if the expected encounter was w' as

$$\mathcal{CP}_{\mathcal{F}}(w'', w', w) = \int_{w[1]}^{w[|w|]} \Delta\mathcal{E}_{\mathcal{F}}(w, w'') - \Delta\mathcal{E}_{\mathcal{F}}(w, w')dw_i$$

This is the area under the so-called *conflict curve* in figure 5.6 that computes as

$$\Delta\mathcal{E}(\text{“success”}, \text{“A cheats”}) - \Delta\mathcal{E}(\text{“success”}, \text{“success”})$$

This curve shows how the difference between expected and actual entropy change grows larger and larger, until the encounter is terminated unsuccessfully. This increases the probability that the participating agents will stop trusting the expectation structures, and that this will inhibit the normal flow of interaction, especially if $\mathcal{CP}$ is large for several paths.

A noteworthy property of this view of conflict is that in cases where entirely new performatives are tried out, the conflict potential is zero. This is because the expected entropy change (which is very large, because the agents know nothing about the consequences of the new performative) is identical to that actually experienced. So what matters about conflict is not whether the expectations associated with a message are clear, but rather whether the effect of uttering them comes close to our expectations about that effect on the expectation structures – a property we might call *second-order expectability*.

Defining a desirability criterion

On the grounds of these considerations, we can now define a very simple desirability criterion, i.e. a concrete quantity for b in the inequality $u(\vartheta^*, F, w, KB) < b$. In principle, requiring that $u(\vartheta^*, F, w, KB) > 0$ is sufficient to ensure that the agent's utility will increase after executing the remainder of F under ϑ^* with encounter prefix w.

To add a tendency towards the reduction of conflict potential, we relax this constraint to allow certain trajectories to be considered desirable even if their utility effects are negative but they also contribute to the evolution of stable interaction patterns. For this, we simply define b as follows:

$$b = -\Delta\mathcal{E}_{\mathcal{F}}(\varepsilon, \mathit{postfix}(T(F), w)) \tag{5.19}$$

This means that an agent will adhere to the frame not only if the expected utility is greater than zero, but also if this expected utility plus the *decrease* in total repository tree entropy (cf. the use of ε and the negative sign) is positive. In other words, the agent is willing to sacrifice some immediate utility for the sake of decreasing the overall entropy of his repository.

5.4 Summary

The purpose of this chapter was to extend the formal model of InFFrA laid out in chapter 4 by ready-to-implement learning and decision-making algorithms.

Starting from the principles of social abstraction and transient social optimality, which we proposed as necessary ingredients of social reasoning and learning algorithms, we used results from the area of hierarchical reinforcement learning to model framing as a two-level Markov Decision Problem. Precup's (2000) "options" framework was subsequently combined with m^{2}InFFrA semantics to develop a framing mechanism which ensures that the learning heuristics used by m^{2}InFFrA agents comply with the decision-theoretic foundations of reinforcement learning.

As concerns frame management, sections 5.2.2 and 5.2.3 proposed heuristics for frame construction and frame generalisation that add a planning/clustering flavour to m^{2}InFFrA, respectively. Although there exist many more operations that can be defined on frames, we believe that the goal-based combination of frames in the context of frame construction and the generalisation of frames that is facilitated by frame merging operations are not only functionally the most important operations. They also seem very adequate in the face of the conceptual principles for InFFrA described in chapter 3.

All this was put together in section 5.3 to obtain an implementable framing procedure, while also covering "everything amiss" in the general model. These remaining elements proposed to supplement the learning and decision-making methods (similarity measure, encounter state definitions, and desirability heuristics) were deliberately considered part of the implementation, as we cannot claim that they carry over to all application domains.

In the following chapter, we will show how intelligent social reasoning agents built using these heuristics and the overall m²InFFrA framework perform in a complex application domain.

6. Experimental Results

The algorithms and heuristics used for m²InFFrA in the previous chapter are sufficiently concrete to allow for an implementation of the principles of InFFrA in a real system. Based on this concretion, this chapter summarises the results of the extensive experimental validation of our methods in a multiagent system inhabited by m²InFFrA agents. The application scenario for this system is chosen from the domain of Web linkage management that has already been briefly touched upon in section 1.1.

First, we provide a description of the *L*ink *E*xchange *S*imulati*ON* system LIESON which is the implemented system used for empirical validation together with a discussion of the application domain and of the evaluation methodology pursued.

The subsequent presentation of experimental results *per se* consists of two sections that summarise our findings in basic and more advanced experiments, respectively. The basic experiments are concerned with analysing the effectiveness of our methods in fairly simple "proof-of-concept" scenarios and to explain how and why m²InFFrA works in principle. The section on advanced experiments deals with tests in which complex negotiation frames were used and the resulting communication processes were much more intricate. These experiments prove that m²InFFrA can be successfully used at different levels of complexity, and they highlight the close interplay between InFFrA-based social reasoning and local agent rationality.

6.1 The LIESON System

Prior to reporting on the experiments conducted with LIESON, we provide a description of the system and discuss the methodology employed in empirical validation.

6.1.1 Linkage liaisons

The LIESON[1] system is a MAS in which agents who represent the owners of individual Web sites try to increase the dissemination of their owners' opinions.

These "opinions" include (i) the views expressed in the contents of the owner's site and (ii) the owner's preferences regarding other Web sites he knows of. To further the spreading of this opinion on the Web, each owner of a site tries to increase the popularity of his own site and that of other sites that contain similar (or favourable) views. Certainly, this popularity is affected by the traffic the respective sites attract among Web users. This traffic, in turn, depends *inter alia* on the link structures in the Web since users follow existing links when surfing the Web.

[1] Pronounce as "liaison".

Considering the vast amount of existing sites and the changes their contents constantly undergo, it is only natural to infer that it is almost impossible (or at least very tedious and time-consuming) to optimise the links to and from one's site as a human Web site owner. It is precisely for this reason that the idea of agent-based Web linkage management seems very appealing: Rather than having to explore new sites as these appear, to constantly track the contents of known sites and to discuss the possibility of linkage between those sites and one's own site with their owners, it would be very convenient to delegate as many of these tasks as possible to an intelligent agent. Such an agent would constantly explore the Web and gather knowledge regarding existing sites and the hyperlinks between them. Using information about the preferences of his "owner", the agent would then either lay/delete outgoing links to other sites self-responsibly or enter linkage negotiations with agents who represent other sites.

In technical terms, such functionality could be realised through an interoperable (e.g. FIPA-compliant (FIPA 1999b)) agent platform on which agents representing different users meet and discuss the possibilities of mutual linkage. The LIESON system is a prototypical implementation of this functionality, while clearly focusing on the "strategic communication" aspect, which is its most important ingredient as far as our work is concerned.

The Web linkage domain has several characteristics that suggest the application of InFFrA-based social reasoning techniques:

1. Self-motivation of agents: Different agents have different motives, and, in general, they need not care about others' welfare.

2. Openness: Arbitrary numbers of agents can enter the scene, they may have been designed by different people, and they are free to change their own hyperlinks as they wish.

3. Dynamics: Links can be modified at any time by Web site owners without prior notice, new sites appear and others disappear, etc.

4. Latent structures: The linkage structures are a visible sign for underlying relationships between different contents. These relationships are rarely made explicit in the contents of the sites.

5. Culture: The success of communication strategies depends on existing rules of conduct, "netiquette", etc.

In systems with these properties, it seems appropriate to learn how to employ and shape communication patterns strategically, and this is essentially what frame-based social reasoning is aimed at.

It should be noted that while LIESON focuses on the link-based management of relationships between Web sites and their owners (so-called "linkage liaisons" (Malsch et al. 2002)), such an application scenario is representative of a much wider class of applications concerned with *communication-based relationship management*. Other application scenarios in this category are:

- Commercial banner trade: This is very similar to opinion-based linkage management, the only differences being that advertisement banners are exchanged for money, and that agent success solely depends on the popularity of one's own site (customer traffic).

- "Social" citation management: Citations in the scientific literature increase one's reputation and popularity in pretty much the same way as links do in the case of Web sites.
- Decentralised management of ad-hoc communication networks: If agents can autonomously choose how to route messages, the quality of service offered when communicating with an agent representing a mobile device depends on his "popularity" and the willingness of others to forward messages to and from him.

6.1.2 System components and agent design

LIESON is a testbed for InFFrA experiments that simulates the evolution of a hypothetical portion of the World Wide Web. The system incorporates:

1. A representation of a number of Web sites and of the link network that results from hyperlink connections between these sites. Based on a model of assumed Web user[2] behaviour, every possible link configuration leads to a particular distribution of user traffic among these sites.
2. A set of agents that correspond to these sites in a one-to-one fashion. These agents reason and communicate with each other to improve their linkage situation in accordance with the "opinion dissemination" goal described above. They are endowed with the capacity of changing the outgoing links of their owners' Web site and have information about their owners' preferences regarding other sites. Also, they can exchange textual messages with each other which do not affect the link environment.
3. A so-called *system manager* which is a centralised, omniscient entity that maintains the link network, computes agent utilities and mediates between agents and link environment by executing changes to the network and conveying world state information to the agents.

The linkage network

The ensemble of Web sites represented by InFFrA agents forms a directed graph whose nodes correspond to Web sites and the edges are hyperlinks pointing from one site to another, the so-called *linkage network*. While we abstract away from the actual content of sites (and coerce all pages/URLs that belong to the same stakeholder into one abstract "site node"), we allow for *weighting* links with numerical weights. These weights are thought to express the attitude of the referring site towards the site referred to by the link as publicly expressed on the Web page. In the real world, this weight can be adjusted by employing special visual means or a textual comment. Examples include the use of different banner sizes with links to distinguish between less important and more important advertisement clients, labelling links with captions such as "the best site on subject X", "my girlfriend's homepage", etc.

[2] When we speak of *agents*, we will henceforth refer to agents that represent Web sites (or their owners/stakeholders, respectively). Web users, on the other hand, are normal human individuals who surf the Net, automated Web crawlers, or search engines. Neither of these latter kinds is simulated by agents in the LIESON system.

In our model of Web user behaviour, we will assume that the probability with which a person/agent surfing the Web follows a link depends on this numerical *public rating* value attached to the link as a weight. The higher the rating value, the greater the likelihood that a link will be followed.

It is important to understand that the *public ratings* that are visible to all other agents and users as link weights need not coincide with the real opinion held by the source site regarding the target site (the *private rating*). In fact, it is only for this reason that strategic linkage behaviour is even possible, because agents are able to negotiate over linkage actions and use their "linkage power" to persuade others to take action beneficial for themselves. If, for example, A has a low private rating for B but B's site attracts a lot of traffic, A might display a high public rating for B to get B to lay a link towards A in turn (which would increase A's popularity). However, as the long-term goal of agents is opinion dissemination and not a simple maximisation of traffic on one's own site, the amount of "strategic pretence" cannot grow arbitrarily – after all, popularity is pointless if you cannot make your point!

Agent knowledge, action capabilities and user interaction

Apart from their m²InFFrA functionality, LIESON agents are characterised by the following properties:

- They start out with no knowledge whatsoever about other sites or existing links. They can query the system to obtain a reference to a random site at any time by sending an `explore()` message to the system manager. Once they know that an agent X exists, they may query what the outgoing links of X are by sending an `update`(X) message to the manager. On receipt of such a message, the manager will send a list of `existsLink`(X, Y, R) facts to the sender, one for each link from X to Y with public rating value R.

  Note that this way of obtaining link information implies that agents almost always have incomplete knowledge of the linkage network. This is the case because they are not automatically notified of linkage actions but have to actively request an information update themselves.

- Apart from explore- and update-actions, agents can execute three kinds of physical linkage actions:

  1. *addLink*(X, Y, R): If no link between X and Y exists, X can lay a link to Y with rating value R. This link will continue to exist until a *deleteLink*(X, Y) action is performed.
  2. *deleteLink*(X, Y): If a link between X and Y exists, an execution of this action by X will cause deletion of this link.
  3. *modifyRating*(X, Y, S): If a link between X and Y exists with rating R, this action will modify its rating value to S. For the action to be executed, it is mandatory that $R \neq S$.

  Agents' knowledge bases contain rules regarding their action capabilities that allow them to infer from the *existsLink* facts obtained from the manager which actions they may execute. For example, the following rule is used for *modifyRating*:

$$existsLink(X, Y, R) \wedge number(S) \wedge R \neq S \Rightarrow can(X, modifyRating(X, Y, S))$$

So, since agent X knows that his name is X, he can infer that he is able to modify the respective link. Note the use of the *number* predicate that is only true for admissible rating values, and the implicit universal quantification of variables. Logical negation is realised thrrough "negation as failure".[3]

Furthermore, add- and delete-lists of facts (in the sense of traditional AI planning systems (Russell and Norvig 2003)) are used to revise beliefs after the execution of physical actions. In the above example, the fact $existsLink(X, Y, R)$ would be retracted from the knowledge base, and $existsLink(X, Y, S)$ would be added to it subsequently. Naturally, update- and explore-actions have belief revision effects that are quite different from those of physical linkage actions, because the replies obtained from the system manager cannot be predicted before executing the action.

- The agents interact with their human "proprietor" to obtain rating information. Every time an agent encounters a new site, he asks his human owner for an assessment of the contents of that site. This assessment is given as a numerical value taken from a fixed range of possible values.[4] In the actual simulation system, however, interaction with the user is implicit, as we are not concerned with agent-human interaction but only with agent-agent interaction. Practically speaking, this means that private ratings can be generated randomly or be retrieved from data files.

  It should be remarked that no agent ever knows anything about the private ratings of another agent; he only "sees" the displayed public rating values (link weights).

With this, the overall simulation process proceeds as follows: Agents enter the system with no knowledge and gradually obtain more and more information about the linkage network. At the same time, the network itself is evolving as agents are adding, deleting and modifying links in pursuit of higher utility while taking their owners' preferences into account. Next, we describe the method by which these utilities are computed in more detail.

Utility computation

In our application scenario, the utility an agent obtains during a simulation (which we also refer to as his *score* below) depends on three things:

1. the popularity of the agent's own site,
2. the popularity of other known sites weighted by the opinion the agent('s owner) has of them,
3. the degree to which the ratings towards third-party sites expressed on peer sites are similar to those of the agent.

[3] This means that $\neg\varphi$ is true if proving φ fails. If φ involves (implicitly universally quantified) variables, negation implies that the formula is wrong for *all* variable substitutions.

[4] In a more advanced system, the agent might, for instance, derive the numerical rating from a more detailed list of attribute values the human user can assign to some Web site to express his opinion of it.

In the following, we are first going to explain how this agent score is computed. After this, we will present our method of determining the *popularity* of a site that is necessary for combining the above score constituents and which builds on a model of predicting the "traffic" on a Web site.

For this purpose, let $G = \langle \mathcal{A}, E \rangle$ the linkage network, i.e. a graph whose set of vertices is the set of agents, and let $E \subseteq \mathcal{A} \times \mathcal{A}$ the set of links, where $r(a_i, a_j) \in [-r{:}r] \subset \mathbb{Z}$ is an integer-valued public rating attached to link (a_i, a_j) that ranges between $-r$ and r (and $r > 0$). In contrast to the displayed rating $r(a_i, a_j)$ of a link (a_i, a_j), the private rating maintained by agent a_i for agent a_j is given by $r_i(a_j)$. Also, let $p(a_i) \in [0{:}1]$ the popularity of agent a_i('s site). We define the score of agent a_i as

$$score(a_i) = \alpha \cdot r_i(a_i) \cdot p(a_i) + \beta \cdot \sum_{a_j \in \mathcal{A}, j \neq i} p(a_j) \cdot r_i(a_j) \cdot \frac{1}{\Delta(a_i, a_j)} \tag{6.1}$$

where

$$\Delta(a_i, a_j) = \max \left\{ 1, \sum_{(a_j, a_k) \in E} |r_i(a_k) - r(a_j, a_k)| \right\} \tag{6.2}$$

is the cumulative difference between all link ratings of outgoing links of a_j with target a_k and the private ratings of a_i toward these a_j (but at least 1 to avoid division by zero in equation 6.1).

We should take a minute to explain this definition. Firstly, the two parts weighted by α and β denote the quantities contributed to the overall score by the popularity of the agent's own site and known peers' sites, respectively. The two weights enable us to vary the importance of these two factors in the design of a particular utility function. As for the first part, this simply computes as the agent's own popularity $p(a_i)$ weighted with the maximally possible rating $r_i(a_i) = r$ (trivially, the agent has the best possible attitude towards his own site).

Now let us look at the second part of the right hand side, which is somewhat more complicated. As with the agent's own site, this quantity takes the popularities $p(a_j)$ of peer sites and the corresponding ratings $r_i(a_j)$ into account. I.e., the more a peer's site is favoured, the more does the popularity of that site contribute to the agent's score. Apart from this, another quantity plays a decisive role in assessing the importance of other agents' success: $\Delta(a_i, a_j)$. This distance measure computes as the sum of differences between the ratings of links pointing from a peer a_j to a third-party site a_k. That is, Δ measures how different the opinion expressed on a_j's site (in terms of out-link ratings) differs from that held by a_i.[5] If $r_i(a_j) < 0$, this implies that the larger Δ, the smaller $p(a_j) \cdot r_i(a_j) \cdot (1/\Delta(a_i, a_j))$, so that the negative utility impact of a disliked peer being popular decreases with increasing Δ. This reflects a need for clear "separation" of different opinions; if the contents of a site are considered unfavourable, this site should ideally express an opinion towards third-party sites that is very distinct from that of the agent for which the score is being computed. This is consistent with our intuition of opinions expressed on Web sites: In a consistent system of different views, it would be most desirable for X if Y did not share the views expressed

[5] Note that the difference is computed on the grounds of one's own *private* rating towards a_k. This is based on the idea that even if one is forced to express a certain opinion that is different from his private rating in public, he will still be more satisfied if others express his actual opinion openly. For example, a dissident who succumbs to oppression and agrees to refrain from further protest will always be happy to see others express similar views in public.

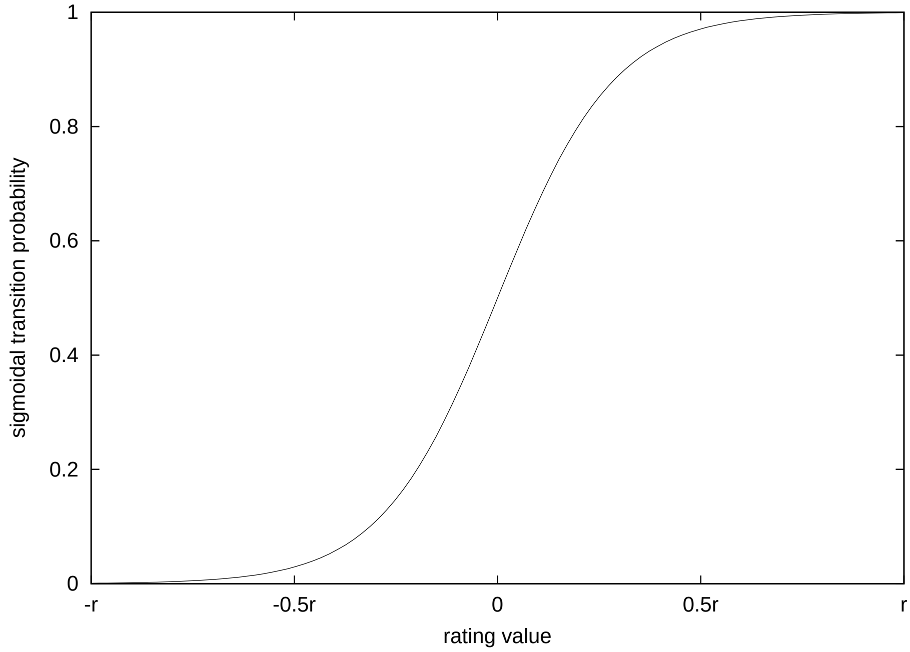


Fig. 6.1: Sigmoidal link transition probability curve for $v = 7$

by Z if X does not endorse Y's views but is a proponent of Z's opinion. After all, it is in the best interest of each agent if Web users are able to distinguish between different views (or "camps") when looking at different Web pages.

Finally, the question remains of how to compute popularities $p(a_i)$. Here, we suggest the following (purely hypothetical) model of Web user behaviour: We assume that a user starts browsing the Web at an arbitrary site a_i with uniform distribution, i.e. $|\mathcal{A}|^{-1}$. Once at a_i, we expect the user to follow a link to another site a_j with probability $P(a_i, a_j)$. This probability is computed as a sigmoidal function of the rating value of the link (a_i, a_j) as defined by

$$P(a_i, a_j) = \frac{1}{1 + e^{-vr(a_i,a_j)/r}} \quad . \tag{6.3}$$

Thereby, the factor v calibrates the probability values to reasonable quantities and division by r ensures that the range of probabilities is the same regardless of r. In other words, different choices of r results in a more fine-grained resolution of different link transition probabilities, while maximal and minimal probability remain identical. Figure 6.1 shows this probability curve, which has the nice property that link ratings around zero make a big difference as to whether a user is likely to follow the link, whereas there is a tendency towards "saturation" for more extreme rating values which seems intuitively appealing.

If the inverse value of $P(a_i, a_j)$ is used as an edge weight, we can extend $P(a_i, a_j)$ to pairs (a_i, a_j) that are not directly connected by edges. This can be done by defining $p(a_i, a_j)$ as the product of all probabilities along the shortest (viz maximally probable) path that connects a_i and a_j. Formally,

$$p(a_i, a_j) = \begin{cases} 1 & \text{if } a_i = a_j \\ P(a_i, a_j) & \text{if } (a_i, a_j) \in E \\ max_{(a_i,a_k)\in E} p(a_i, a_k) \cdot p(a_k, a_j) & \text{else} \end{cases} \tag{6.4}$$

can be used to derive a general probability for a user to visit a_j after a_i.

With this, the total probability of a site a_i being visited by some user (and which we take to be the popularity of the agent) is

$$p(a_i) = \sum_{a_j \in \mathcal{A}} \frac{1}{|\mathcal{A}|} \cdot p(a_j, a_i) \tag{6.5}$$

Because computing shortest paths can be computationally expensive, we also provide an alternative, much simpler way of computing site popularities which does not involve computing all-pairs shortest paths. Instead, it only considers the immediate predecessors of a_i when computing his popularity:

$$\hat{p}(a_i) = \sum_{\{a_j | (a_j, a_i) \in E\} \cup \{a_i\}} \frac{1}{|\mathcal{A}|} \cdot p(a_j, a_i) \tag{6.6}$$

This simplified popularity computation method can be applied just like that of equation 6.5 in total agent score computation.

Either way, the interesting thing about this computation of agent popularities and scores is that by laying links to other agents, the agent is effectively *decreasing* his own probability of being visited. This is quite realistic, because any reference to another site is likely to distract a Web user's attention from one's own site. It is quite interesting to note that this is also true of negatively rated links, so that there is no way at all to "harm" someone directly in terms of user traffic if linkage is the only means available.

With this respect, the question arises how an agent will link his pages to those of others in a beneficial way although they only distract those visiting his site? This is essentially the dilemma agents are faced with in the "linkage game" from a decision-theoretic perspective. We will return to this issue in section 6.2 when we discuss possible score distributions for specific private rating configurations that we used in our experiments.

Finally, we have to explain how popularities and scores are utilised in LIESON:

- From the system manager perspective, equations 6.1 and 6.5 are used to measure the performance of agents during simulations, given a global and omniscient view of the linkage network. Thus, agents' scores are regarded as the primary performance indicator, and they constitute the basic unit of performance analysis since they are the quantity to be maximised from an individual agent perspective.

  Note that under realistic circumstances, a system manager would not be able to assess total agent score because he does not have any knowledge of agents' private ratings. For simulation purposes, however, it is useful to permit access to this private agent knowledge to evaluate agent performance objectively (i.e. given the *actual* linkage network, and not the partial view agents have of it).

- From an agent perspective, local linkage network models are maintained by each agent. We assume that agents have a possibility to estimate their own hypothetical score for any given link configuration. By modifying their local (partial and usually incorrect) graph and re-computing their score, they can assess the utility of a linkage action, and it is on the grounds of these utility predictions that they make their decisions. To maintain a "boundedly rational" attitude in agent design, we only allow them to use the simpler method of popularity computation (equation 6.6). This will give them a hint for the usefulness of an action while not providing perfect information about the effects of that action.

Despite our efforts to devise this utility computation methods in a realistic way, it has to be remarked that it remains a heuristic approximation that is not based on empirical data regarding user behaviour. Also, it does not take fluctuations in Web user behaviour into account. For the sake of reducing computational complexity we refrain from simulating Web users themselves by agents, which would be necessary to model more complex Web "surfing" behaviour.

It should be remembered, however, that the above function fulfils our basic requirement, namely that there has to exist a potential for strategic linkage. Ultimately, we are not interested in whether the computed popularity values resemble real Web traffic measurements, but whether the behaviour of these popularity distributions causes agents to engage in strategic communication.

Local BDI reasoning

As mentioned in earlier chapters, InFFrA is not a complete agent architecture *per se* – it only provides a framework for the social reasoning capacities of an agent.

Therefore, it has to be supplemented with a local rational reasoning component. In the case of LIESON, we have chosen to use a BDI (Rao and Georgeff 1992, Georgeff and Rao 1995) architecture according to which agents generate a number of possible goals and use a goal queue to pick the most appropriate goal in every reasoning cycle.

To keep things simple, we let LIESON agents only generate goals that would be fulfilled by execution of just one action. Thus, there is no real planning process as each plan consists of just one primitive linkage action (*addLink*, *deleteLink* or *modifyRating*). As a consequence, the goal queue is equivalent to an action queue.

What the agent does in each reasoning cycle can basically be summarised as follows: First, the set of all link modification actions that can be performed is computed given the agent's current state of beliefs (the agent maintains a knowledge base that contains his beliefs and is constantly updated with incoming information). Then, a fixed number of actions is randomly selected from the set of all linkage actions, both from the actions the agent can execute himself and from those other agents might perform. The agent mock-executes these actions *hypothetically* on his local (incomplete and/or incorrect) model of the linkage network. Score computation on the resulting network allows for estimating the utilities of these projected actions.

After this, all of these envisaged actions that would decrease the agent's score are deleted. The remaining actions are enqueued into the so-called *action queue* Q. Q is always ordered decreasingly in predicted future score, so that its topmost element can be selected for execution after each step (and dequeued subsequently), since it is expected to maximise the agent's score compared to all other considered alternatives. However, since the topmost element may stem from previous reasoning steps, its executability is tested once more right before execution to ensure that actions which cannot be executed currently are ignored.[6]

Also, actions whose effects have already been achieved and actions that will not increase the current score anymore (because it might already be higher than the one they suggest) are discarded.[7] Finally, it should be mentioned that the length of Q is bounded by parameter l, so that the least promising elements are eventually deleted – actions formerly enqueued and never executed are eventually "forgotten". Table 6.1 summarises the steps of BDI decision making in LIESON. It also describes the very simple communication process incorporated in the BDI component of LIESON agents that is *only* activated when the m²InFFrA component is not used. This "naive" kind of communication is quite useful when it comes to assessing the contribution of m²InFFrA to agent performance (see the experiments in section 6.2.2). For this reason we should take a minute to describe it.

Naive communication basically consists of agent a_i sending a

$$\texttt{request}(a_i, a_j, A)$$

message to agent a_j whenever an envisaged future action A cannot be executed by the agent himself that is generated in the above reasoning cycle. Receipt of this message causes a_j to add it to his private message queue R.

For each agent to consider the requests of others, an additional step (step 4 in table 6.1) has to be added to the decision-making procedure, in which a certain number of requests is processed. The way in which they are treated is identical to that of the actions randomly generated and projected by the agent himself: they are mock-executed, their alleged utility is calculated, and they are enqueued into Q as if the agent had "thought of them" himself.

Clearly, this kind of communication behaviour is highly benevolent, as agents treat others' desires as if they were their own. This is not to say that agents are willing to sacrifice their own welfare for the sake of others, but they will try to satisfy every request as far as their own utility constraints and computational resources allow them to.

In an entirely cooperative society, this simple mode of communication would be a reasonable solution, as it allows agents to exchange information regarding profitable link configurations. This would result in a distributed search for a globally optimal linkage network. However, since agents receive no feedback as to whether the requests they issue are processed by their peers, it is not a viable solution for agent societies in which selfish agents might not bother about others' requests. Thus, "naive" agents might end up requesting actions from others instead of executing slightly less profitable alternatives themselves. Even if they eventually choose to execute link modifications themselves, they would (in the best case) lose time and waste resources on costly but fruitless communication. In the worst case, they might even never consider other actions, because linkage actions of others are always more profitable (this is true, for example, of link additions from others towards one's own site). This effect will be further pondered on in section 6.2.2.

[6] However, its utility under current linkage conditions is not re-assessed. This may cause the agent to execute actions that have been enqueued earlier and that would not have been considered profitable under the linkage network model the agent has at present.

[7] This process of deleting obsolete and redundant actions is repeated until a non-obsolete, non-redundant action is found; if no such action exists, the agent remains inactive.

1. Generate a number n_a of own actions to consider for execution in the next step.
2. For each of n_o known peers, generate a number n_{oa} of action that the respective peer can currently execute.
3. Assess the hypothetical value for all generated actions (by computing $score(a_i)$ for oneself as if the action had already been executed and comparing this quantity to one's current score).
4. Process n_{req} requests obtained from others that have been stored in the requests queue R.
5. Filter all actions that do not increase the current score, all actions whose effects have already been achieved, and all those which are already contained in Q from the union of these three sets.
6. Repeatedly select the topmost action from Q and dequeue it until all of the following conditions hold or the queue is empty:
   (a) The effects of the action have not yet been achieved.
   (b) The action is currently executable.
   (c) The action will presumably increase the agent's score.
7. If no action could be found and Q is empty, do nothing; else, execute the selected action or send a request to the agent who is able to execute it.
8. Update the knowledge base with information about the effects of the executed action.
9. If $|Q| > l$, delete the bottom $|Q| - l$ elements of Q until it has reached its maximally admissible length l.

Tab. 6.1: The BDI reasoning cycle in LIESON, parametrised with constants n_a, n_o, n_{oa}, n_{req} and l.

Exploration and exploitation

The BDI reasoning process describes active utility-oriented decision making at the sub-social level. To obtain and update information about other peers and existing hyperlinks it is also necessary to perform explore- and update-actions occasionally.

Therefore, agents choose to "exploit" with a fixed probability p_e in each round, in which case they deliberate in the way described above. With probability $1 - p_e$, they choose to explore the link environment. This means that, with a fixed probability p_u they send an `update`(X) message to the system manager to update their knowledge regarding the outgoing links of agent X's site (where X is a randomly chosen peer they already know). With probability $1 - p_u$, they send an `explore`$()$ message to the system manager to obtain information about other existing peers they might not be acquainted with (note, though, that there is no way to avoid that they obtain information about agents they may have heard of already).

Social InFFrA layer

When m²InFFrA is put into operation, encounters start with the same kind of request as above, i.e. the initial message is generated by the BDI layer. In contrast to "pure BDI" mode, however, this is followed by the m²InFFrA reasoning process taking over as described in section 5.3.1. Issuing or receiving such a request spawns an encounter start, whereupon

- agents choose their next message or action according to m²InFFrA until the encounter is terminated;
- they terminate an encounter if a certain fixed amount of time has expired while waiting for a response from their peer;
- they send a "busy"-message to any third party that attempts to start an encounter with them while another encounter is still running (i.e. every agent engages only in one conversation at a time);
- BDI actions from the queue (apart from requests) are only executed during the encounter if the agent is waiting for a reply;
- agents refrain from exploration (explore- and update-actions) until the encounter is over.[8]

This means that during an encounter the InFFrA layer has priority over the BDI layer, unless it makes no suggestion because the agent is waiting for a reply.

Naturally, as communication takes place asynchronously, encounters may fail unwittingly if wait states expire due to overly long reasoning on the peer's side. Since the reasons for these "broken" encounters are unknown for the agent who is experiencing them, the perceived encounter sequences are stored in the repository just like any "normal" frame. Also, quite obviously, conversations may span an extended period of time, during which the link network may change. This adds to the complexity of the application, as agents may be forced to cancel the execution of actions they had committed themselves to during a conversation because these actions do not appear desirable anymore.

Finally, it should be made clear that there is no facility to store social commitments that result from previous interactions. Hence, if agents discover that an action executed earlier has caused a link configuration that does not seem advantageous any longer, they may undo the effects of this action at any time (this effect is discussed in section 6.2.2, p. 182).

6.1.3 Implementation

LIESON is a full-fledged simulation system that embodies the above functionality and all the algorithms and data structures described in chapters 4 and 5. The system is entirely Java-based, platform-independent, distributed and enriched with full graphical user interface (GUI) support for inspection and manipulation of many system components. It is characterised by a variety of features which are discussed in detail (Rovatsos 2002–2004). Here, we only list the most important ones:

[8] This is not the case in the advanced experiments, where update actions can be used as a result of the *check*$(\cdot)$ predicate, cf. section 6.3.3 (p. 200).

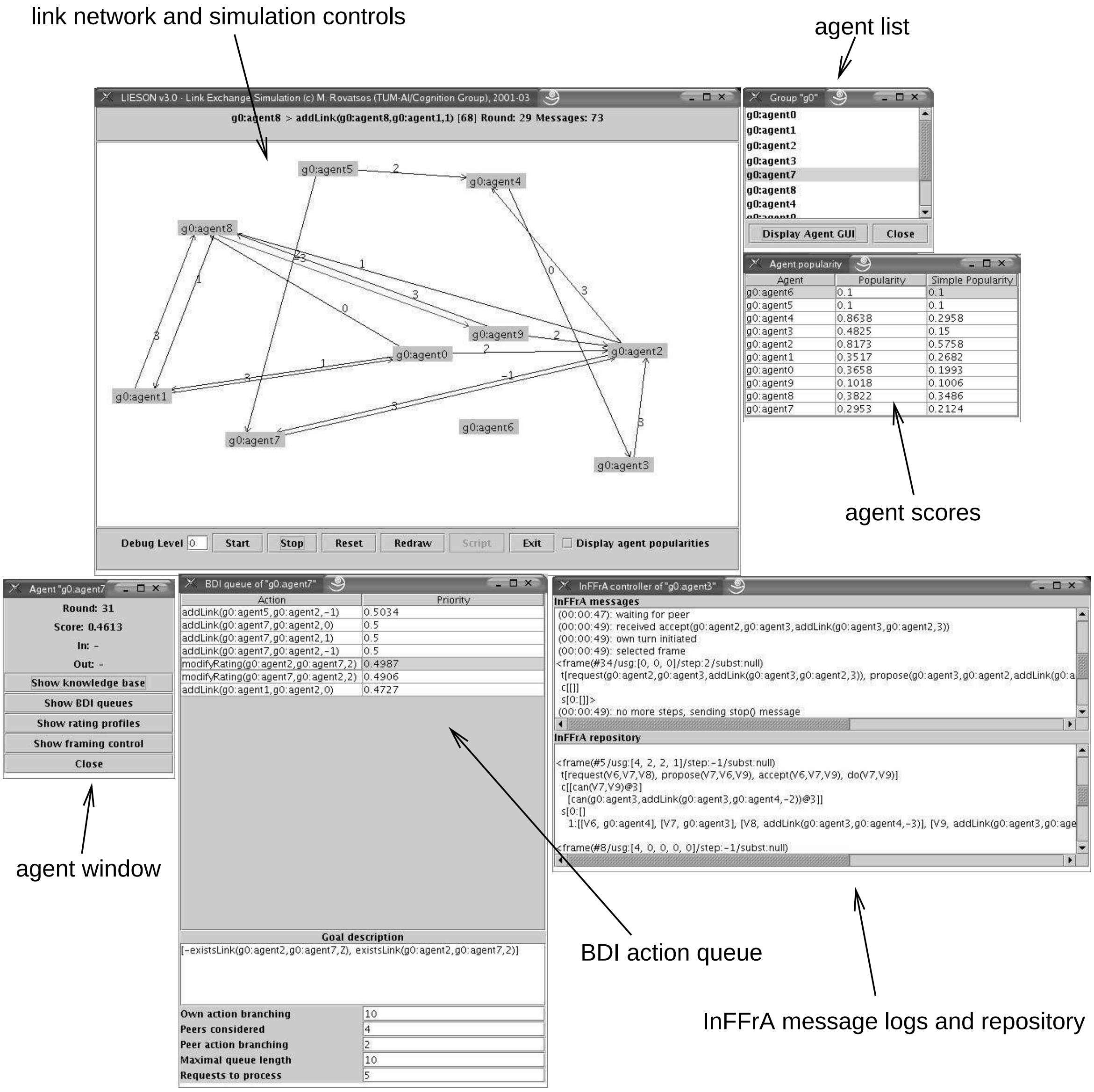


Fig. 6.2: Screenshot of the LIESON system

- LIESON can be used to perform arbitrary-length simulations in single-threaded, multi-threaded and (by utilising the JADE (2002) agent platform) multi-host modes. While single-thread mode synchronises agent communication and action execution for ease of analysis, multi-threading allows for genuinely asynchronous agent operation and communication. Multi-host mode offers the additional possibility of spreading arbitrarily-sized groups of agents across different hosts which communicate with the system manager that runs on a single agent platform (and host). This enables us to run simulations with larger agent societies, and also to add new agents to the system over time.

  The system manager maintains the global linkage network, manages agent addresses that are necessary for networked message transport and computes objective popularities (and scores, if in possession of private rating information for all agents, cf. our remarks on p. 152).

- Various GUI components visualise
  - the linkage network (as a graph whose nodes are randomly positioned but can be interactively moved by the user), agent popularities and scores,
  - BDI data (current subjective score, knowledge base contents, executed actions and obtained messages, the action queue) for each agent; also, all internal BDI parameters mentioned in the sections above (all of them can be modified, and interactively when using the GUI)
  - m^2InFFrA logs for each agent which allow for tracing communication processes and observing the contents of the frame repository.

  The GUI also allows for interactively starting, suspending and continuing a simulation run. A "simulation scripting" facility allows for recording the link modification actions that occurred during a simulation to a file, so that they can be promptly "re-played" after an experiment to visualise the evolution of the linkage network without having to waste time on intra-agent reasoning.

  All the information provided in GUI components can also be written to the standard output which is convenient to conduct data analysis when running simulation experiments offline without GUI support. Also, by virtue of JADE functionality, GUI components can be hidden and re-visualised when running batch simulations offline.

- Agents in LIESON employ a Prolog-like logical reasoning engine that allows for the insertion of arbitrary rules in the form of Horn clauses in agents' core knowledge bases. This engine is used to conduct logical inference needed for both BDI and m^2InFFrA purposes. The system provides facilities for reading private rating information from files, generating random private ratings, and for recording agent scores to files for analysis. Also, agents can write various statistical information to data files, such as the contents of their Q-value tables, their re-framing frequency over time, etc. Finally, frame repositories and Q-tables can be stored persistently allowing for experiments with agents that have prior interaction knowledge.

Figure 6.2 shows a screenshot of the LIESON system to give an impression of the GUI components that are available. All in all, LIESON is a very flexible and complex simulation testbed for InFFrA experiments that provides a rich functionality and simulates a complex application problem. In the following section, we will elaborate on how it has been used to evaluate the proposed social reasoning methods.

6.1.4 Evaluation methodology

The methodology we apply in validating m^{2}InFFrA agents in LIESON is based on the following principles:

Two levels of communication complexity With the primary goal of developing an architecture for learning and strategically using interaction patterns in mind (see chapter 1, p. 1), we have to provide evidence that our agents are capable of handling and combining such pre-specified patterns to further their goals. Since this should be the case regardless of the level of elaboration of a given communicative context, we conduct two series of experiments.

First, in our *basic* experiments, we endow agents with knowledge about fairly simple *proposal-based* frames. Effectively, what such frames achieve is that they enable agents to (i) exchange information about their private preferences by requesting actions from others and (ii) to agree on joint actions in case the individual contributions to such a joint action are not mutually beneficial but the combined action is. These experiments, which prove that using m^{2}InFFrA has a concrete positive utility impact for communicating agents in complex domains, serve as a *proof-of-concept* application of our methods. They show that m^{2}InFFrA works as a strategic communication-learning algorithm.

The second series of experiments – which we refer to as *advanced* experiments – is conducted using more complex negotiation frames. More precisely, we apply elements of the theory of *interest-based negotiation* (Rahwan, Sonenberg and Dignum 2003) to develop negotiation frames which involve argumentation about (i) beliefs held of the environment and (ii) the interacting parties' internal goal structures. The rationale behind this series of experiments is twofold: On the one hand, we want to examine whether agents are able to adapt to a more complex pre-specified communication regime that is characterised not only by the visible communication signals that are used but also by logical constraints that reflect reasoning about social and mental states. This is particularly important as agents in applications such as LIESON might be thrown into a pre-existing, complex social context with which they should be able to cope. On the other hand, developing negotiation frames serves as a case study for the application of InFFrA to a particular communication problem. This means that these advanced experiments implicitly also describe the process of applying the general architecture to a specific communication scenario in practice.

Comparison between different types of agents For InFFrA to make sense as a social reasoning architecture, we have to justify that adding an m^{2}InFFrA module to an intelligent agent will improve his performance in real terms and offer an advantage over other agent designs.

Of course, it is impossible to make definitive statements regarding the superiority of InFFrA over other architectures, especially because it has only been implemented in one domain, and also for the lack of comparable "communication reasoning and learning" architectures. However, we can gain a deeper understanding of the advantages and limitations of InFFrA if we compare it to certain simple types of agents, such as:

1. *randomly acting agents* whose (poor) performance reflects the complexity of the linkage domain and puts the performance of m²InFFrA agents into perspective,
2. *non-communicating BDI agents* who improve their linkage status solely using their own action capabilities and are susceptible to local utility minima since they are unable to compromise, and
3. *non-empirical, communicating BDI agents* who direct requests to others but do not learn when these requests are honoured and when they are ignored (as described on page 154).

This comparison is useful to understand which role is precisely played by communication in the context of rational agent reasoning, especially with respect to our theory of empirical communication semantics (cf. section 4.1). It explains how the communicative expectations derived from observation capture previously unknown social interaction structures.

Apart from these comparisons to non-m²InFFrA agents, we also examine the role certain m²InFFrA elements play in the overall reasoning mechanism. To this end, we compare slight variations of the m²InFFrA design presented so far, which result in different sub-types of m²InFFrA agents by virtue of the following distinctions:

- *Desirability tests:* The effects of including a desirability test during (i) action selection (as described in section 5.3.1), (ii) frame selection, (iii) at both levels or (iv) at neither level are compared.
- *Desirability criterion:* Agents who apply a strict desirability criterion (that requires an action to increase the current score to be considered desirable) are compared to agents with the more lenient, entropy-based criterion introduced in section 5.3.4.
- *Frame selection strategies:* We compare the performance of agents who use Q-value optimisation to select frames as described in chapter 5 to non-learning agents.

Using the results of simulation experiments with these variations, we can justify the design decisions that have lead to our specification of the m²InFFrA model, as they highlight the contribution of its core components to the overall performance of the system.

Fixed agent preferences, environment parameters and performance measures In validating the proposed methods, it is not our aim to fine-tune all parameter settings so as to achieve optimal performance for arbitrary population sizes, prior agent popularities (i.e. private rating distributions), existing communicative conventions, etc., let alone for application domains other than Web linkage. We have rather chosen a domain that is sufficiently complex to ensure that we can*not* control the effects of each parameter with the rationale that if agents are able to exhibit *satisficing* behaviour even for potentially sub-optimal settings, this reassures our belief that further fine-tuning will – in the worst case – not produce results that are any worse than the ones we obtained.

Therefore, unless explicitly stated (whenever different configurations were chosen to analyse the effects of particular changes to the simulation environment), we will adhere to fixed parameter settings as concerns:

- Internal agent reasoning parameters:
  - importance of own popularity vs. popularity of peers with similar opinions (parameters α and β in equation 6.1, p. 150)
  - BDI: exploration probabilities (i.e. probabilities of `update` and `explore` actions), BDI queue size, number of projected own and others' future link modifications
  - InFFrA: learning rate, exploration policy and discount factors in Q-learning, similarity measure (sections 5.15 and 5.3.3)
- Global parameters:
  - population size
  - global rating profile, i.e. private ratings held by each agent towards other agents

As for performance measures, our primary unit of analysis is the running utility/score of agents as measured by the (supposedly omniscient) system manager. Thereby, we look at the average performance of all agents, but also at the standard deviation between agents' scores that provides a measure for the divergence between individual scores. Also, we examine the best and worst scores obtained by agents in a simulation taking into account that the *a priori* popularity of agents (as reflected by the private ratings other agents hold of them) determines the utility that they can achieve in theory.

In the analysis of more complex negotiation frames, we shall also take a more qualitative look at particular conversations. While this does not necessarily directly explain their utility performance, it is very helpful in understanding how m^{2}InFFrA works.

6.2 Basic Experiments

In the first series of experiments, a group of ten agents with a very specific set of private ratings exchanges requests and counter-proposals for linkage actions. Despite the simple structure of the frames used in these simulations, our m^{2}InFFrA agents clearly outperform (non-communicating as well as communicating) BDI agents, thus proving that adding communication learning capabilities can substantially improve agent performance in open multiagent systems. Prior to presenting the results of these simulations, we explain the system configuration for these experiments in detail.

6.2.1 Experimental Setup

Basic BDI and utility computation parameters

Throughout all experiments (including the advanced experiments described in section 6.3) we adhere to fixed parameter settings for the BDI reasoning components and utility computation, as shown in table 6.2. This means that agents explore the link environment (rather than engage in link manipulation or communication actions) with 20% probability in each reasoning cycle, and base their decisions on BDI- or InFFrA-based decision making with a probability of 80%. If they decide to explore, then there is a 10% probability that they look for new sites, else they update their information regarding the outlinks of a site they

Parameter	*Value*
exploration probability $1 - p_e$	0.2
probability of `update`(X) in exploration mode p_u	0.9
probability of `explore`$()$ in exploration mode $1 - p_u$	0.1
range of possible link rating values r	3
importance of popularity of own site α	1.0
importance of popularity of other site β	1.0
number of own projected actions n_a	10
number of peers for which actions are projected n_o	4
number of projected actions per peer n_{oa}	2
number of peer requests processed in each iteration n_{req}	5
maximal length of BDI queue l	10

Tab. 6.2: Basic BDI parameter and utility computation settings. The symbols used refer to those introduced in the respective sections on utility computation (p. 149) and local BDI reasoning (p. 153)

already know (chosen randomly from their acquaintances). Quite deliberately, we have chosen not to decrease the exploration rate over time since – despite the fact that we are using constant population sizes – it is unrealistic to assume that in an open MAS, the entire linkage network (or even the set of existing sites) will ever be exhaustively explored.

When basing their decisions on the BDI reasoning component (i.e. (i) in the case of non-InFFrA agents, (ii) if the InFFrA component makes no suggestion because it awaits a reply in an ongoing conversation, or (iii) if the agent has no reason to start an encounter), agents generate ten own actions at random and project the utility these might yield if executed. The same is done for two actions of four peer agents (or less, if the agent is aware of fewer peers), where both the peers and their actions are chosen randomly. So, on the whole, each agent predicts the utility of eighteen actions in each simulation round (reasoning cycle). Additionally, in the case of "non-empirical, communicating BDI agents" (cf. pp. 154 and 175), at most five requests from the requests queue are processed in each round (less, if the queue is shorter). This results in projecting another five (requested) actions utility-wise, so that twenty-three utility values are compared to those of actions contained in the goal queue Q altogether, and the queue is rearranged so that it contains at most ten goals (which correspond to link manipulation actions).

As regards utility, agents have seven link rating values to choose from for each link, i.e. all integers from -3 to 3. When computing their total score, they weight the contributions of their own popularity and that of other agents (in relation to how much the agent likes those other agents and taking into account how much the opinions they express in terms of links to third parties differ from his own) equally. Looking back at equation 6.1, this means that the *total* effect of others' popularities on the overall score effect of an agent is effectively much stronger than that induced by changes to one's own popularity.

InFFrA setup details

Although we have devoted several sections to the details of how to implement m²InFFrA in practice in chapter 5 (especially in section 5.3), we still need to specify (i) concrete settings for the numerical parameters used in frame learning and (ii) more structural aspects of the m²InFFrA design used in our experiments (the utility function used to assess message and action sequences, the initial frame repository, and encounter state definitions).

Frame learning parameters As concerns (i), the parameter values used in the Q-learning based frame-learning procedure (cf. section 5.2) are shown in table 6.3. The only two aspects that require some clarification here are the cooling policy which determines how the temperature T_k used for Boltzmann exploration (equation 5.15, p. 125) decreases over time and the values $Q_0(s, F)$ with which the Q-table is initialised. The decay of T_k uses a "cooling rate" θ slightly less than 1 as is common practice in the Q-learning and simulated annealing literature (see (Mitchell 1997)) and (Russell and Norvig 2003), respectively), and initialising the Q-table with zero values results in taking Q_1 to be the (cumulative, discounted) reward $\hat{R}(s, F)$ experienced during the first encounter in which F was selected in state s (this follows directly from equations 5.12 and 5.13 on p. 124). Again, this is common practice in Q-learning, and it reflects the fact that in the absence of any further knowledge, the first experience with a (framing) choice is used as an initial approximation of the utility of that option.

Parameter	*Value*
initial Q-value $Q_0(s, F)$	0
initial exploration temperature T_0	1
cooling rate θ	0.95
cooling policy	$T_k = \theta^k T_0$
discount factor γ	0.95

Tab. 6.3: InFFrA learning parameter settings. Notation as introduced in section 5.2.

Message sequence utility The aspects listed under (ii) require a somewhat more extensive treatment. First of all, although section 6.1.2 motivates and describes the computation of agent popularities and scores at length depending on the configuration of the current linkage network, this does not say anything about the utility assigned by agents to a certain sequence of messages and physical actions $u(w, KB)$ introduced in definition 4.4 (p. 105). Recalling that this utility is used as an estimate for the desirability of frames and concrete, ground trajectories it is clear that the way u is computed has a decisive impact on the behaviour of the system.

In our experiments we choose to compute the utility of a sequence $w = w_1 \cdots w_n \in \mathcal{M}_c^*$ as the sum of the utilities of all w_i where

- the utility of a physical do-action is the difference between the agent's total score after (hypothetical) execution of the link addition/deletion/modification and his current score (not obtained through environment feedback but predicted solely using one's own subjective knowledge of the linkage network), and

- the (negative) utility of any other non-physical message is a small quantity that corresponds to the cost incurred by uttering a message.

This entails that if w contains k messages and $|w| - k$ physical actions, the total cost of messages in w is k times a small negative quantity (that is almost negligible compared to the utility effects of physical actions). The remaining $|w| - k$ physical actions are executed iteratively in the order in which they occur in w, and the final score resulting from these linkage modifications diminished by the cost of the communicative messages involved is then used to compute the utility $u(w, KB)$ of w by comparing the agent's total score after execution of w to that prior to the execution of the sequence. This ensures that useless communication without any physical consequences does will not go on forever.

To add an element of "bounded rationality" to the estimation of utilities, agents use the "simple popularity" computation of equation 6.6 (p. 152) which only considers incoming edges in computing the popularity of a site (rather than all shortest paths from other sites). In this way, we can ensure that agents have only incomplete information about utility values, while we use the more complex variant of popularity computation (based on equation 6.5, p. 152) to evaluate agent performance from an external point of view (see p. 152).

Frame repository initialisation Next, we have to discuss which set of initial repository frames is actually used in these basic experiments. The three proposal-based frames we use are shown in table 6.4.

$$
\begin{aligned}
F_1 = \; & \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{accept}(B,A,X) \xrightarrow{0} \texttt{confirm}(A,B,X) \xrightarrow{0} \texttt{do}(B,X)\big\rangle, \\
& \big\langle can(B,X)@3, effects(X)@4\} \big\rangle \\
& \big\langle \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle \\
F_2 = \; & \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{propose}(B,A,Y) \xrightarrow{0} \texttt{accept}(A,B,Y) \xrightarrow{0} \texttt{do}(B,Y)\big\rangle, \\
& \big\langle \{can(B,Y)@3, effects(Y)@4\} \big\rangle \\
& \big\langle \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle \\
F_3 = \; & \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{propose-also}(B,A,Y) \xrightarrow{0} \texttt{accept}(A,B,Y) \\
& \quad \xrightarrow{0} \texttt{do}(B,X) \xrightarrow{0} \texttt{do}(A,Y)\big\rangle, \\
& \big\langle \{can(B,X)@3, effects(X)@4, can(A,Y)@4, effects(Y)@5\} \big\rangle \\
& \big\langle \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle
\end{aligned}
$$

Tab. 6.4: Basic proposal-based frames

The first frame F_1 represents conversations in which the second agent B agrees to perform the requested action X and executes it after A has confirmed the original request. In F_2, B can (counter-)propose an alternative Y that he prefers over X and execute that action instead of X. This enables B to suggest that he can do something for A, even if this is not necessarily what A requested originally (but might still find useful). F_3 finally, is the most powerful of all three frames, as it involves actions by both agents. According to this frame,

B may suggest an action for A to be executed in addition to the action X that B agrees to perform himself, i.e. the two agents effectively agree on a *joint* action. With this, F_3 represents the simplest form of joint planning. At the same time, F_3 is the riskiest of the three frames because it involves physical actions *before* it is completed. This is because agent A may refrain from executing Y after B has executed the action X that A originally requested (a situation similar to that discussed in section 5.3.4, esp. figure 5.4).

As explained in the description of the InFFrA layer in LIESON (p. 156), each conversation starts with a `request`-message spawned by the BDI layer, and therefore each frame has to start with a `request` message as well. Also, to avoid defining one frame for each "unsuccessful" execution of each frame agents send a final $\texttt{reject}(\cdot,\cdot,C)$-message (where C is the content of the previous message) whenever they cannot find a suitable frame. This means that we do not need to add frames to the repository for "broken" sequences such as

$$\begin{array}{l}\langle\texttt{request}\rightarrow\texttt{accept}\rightarrow\texttt{confirm}\rangle\\ \langle\texttt{request}\rightarrow\texttt{accept}\rangle\\ \langle\texttt{request}\rightarrow\texttt{propose}\rightarrow\texttt{accept}\rangle\\ \langle\texttt{request}\rightarrow\texttt{propose-also}\rightarrow\texttt{accept}\rightarrow\texttt{do}\rangle\\ \vdots\end{array}$$

and is also in accordance with the view of rejection laid out in section 4.1.6. If no (valid, adequate and desirable) frame can be found in the repository (or constructed by composing repository frames), the agent has no clear expectations about what is going to happen next. For this reason, he utters a `reject` message to indicate that he cannot comply with any expectation on the other's side. In our basic experiments, such rejection simply ends an encounter (agents are not allowed to reply to a `reject` message).

This means that in the first series of experiments, agents are not able do deal with rejection and communication is simply blocked by it. We shall see in the section on advanced experiments below how frames can be constructed in a way such that agents can use them to deal with rejection and conflict explicitly.

Frame conditions The conditions of all three frames contain the *can*-predicate introduced in chapter 4 (equation 4.3, p. 101) for physical actions[9] and the dynamic predicate $\mathit{effects}(X)$ that adds the effects of action X to the knowledge base. Also, conditions in LIESON frames include a notion of time denoted by labelling a predicate P with a time-stamp i, i.e. writing $P@i$. The semantics of statement $P@i$ is "P has to be true *before* the ith step of the trajectory can be executed" where for the first step of the trajectory $i = 0$. If such i is omitted, P has to hold throughout execution of the trajectory.

In more operational terms, the agent proceeds as follows when determining $\Theta_{poss}(F, KB, w)$ (p. 101). For each condition set $C_j = C(F)[j]\Theta(F)[j]$ that results from applying the respective frame substitution, he performs the following steps:

- Let $\Theta_j = \{\vartheta | \vartheta = \vartheta_{\mathit{fixed}}(F, w)\vartheta', C_j\vartheta \text{ and } T(F)\vartheta \text{ ground}\}$
- Let $C_j = \{d_0, \ldots, d_l,\ c_0@0, \ldots c_{k_0}@0,, \ldots c_{k_{|w|-2}+1}@(|w|-1), \ldots, c_{k_{|w|-1}}@(|w|-1)\}$

[9] For messages, fulfilment of this constraint is trivial, as any agent can utter any of the trajectory messages as long as it is his turn.

- Iterate over i for the remaining steps of the trajectory $T(F)[|w|+1]\ldots T(F)[|T(F)|]$:
  - Prove time-independent facts $d_0,\ldots,d_l$
  - Prove facts $c_{k_{i-1}+1}@i,\ldots,c_{k_i}@i$ relevant for next step i
  - Restrict Θ_j to the substitutions returned from these proofs

What this procedure does is to prove all conditions for the remaining time-steps $T(F)[|w|+1]$ to $T(F)[|T(F)|]$ (if w is the running encounter prefix) to determine the substitutions that are still possible under a certain condition/substitution pair.

Thereby, time-independent facts (those without @) are proven in each time-step i while all $P@i$ are proven only at time-step i. The set Θ_j is iteratively restricted to those substitutions that are compliant with the conditions of step i until we are left with those substitutions for which the entire trajectory postfix can be executed under the jth condition set. When the procedure has been repeated for all $j \leq |C(F)|$, we obtain $\Theta_{poss} = \cup_j \Theta_j$ as the union of all substitution sets that are admissible according to a single condition/substitution pair.

While this condition proof procedure may seem fairly simple, it involves certain intricacies:

- Since we are using dynamic predicates such as $\mathit{effects}(\cdot)$, the proof procedure must iteratively modify the knowledge base after each step i to prove the remaining facts. In F_3, for example, if $X = \mathit{addLink}(a_1, a_2, 0)$, then one of the effects of X is $\mathit{existsLink}(a_1, a_2, 0)$. When using $\mathit{can}(A, Y)@4$ to generate admissible substitutions for Y, we have to consider that $\mathit{existsLink}(a_1, a_2, 0)$ is already in the knowledge base, so that Y cannot be bound to $\mathit{addLink}(a_1, a_2, \cdot)$.

- As in the case of *effects*, frame conditions often contain conditions *precipitated* (rather than presupposed) by certain trajectory steps that have to be performed for those facts to become true. However, these conditions have to already hold at the time of anticipating the trajectory steps that will effect them. In our implementation, we have solved this problem by "mock-modifying" the knowledge base when determining Θ_{poss} for a frame (prior to its actual execution). To this end, we use two dynamic predicates $\mathit{add}(P)$ and $\mathit{remove}(P)$ to add or remove fact P from the knowledge base. These become true after they have performed the insertion/removal action, and all other dynamic predicates like $\mathit{effects}(X)$ are defined using them.

- It must be ensured that when actually *enacting* a frame, all conditions $P@(i+1)$ in the currently applied condition set (which is any of the condition sets $C[i]$ for which $KB \models C[i]\vartheta$ is true under the selected substitution ϑ) are "proven" to ensure that the effects of trajectory steps modify the knowledge base according to frame conditions (to cater for the different kinds of context conditions introduced in section 3.3.1). In particular, this is true of the ultimate step of a frame, since the conditions relevant *after* that step have not been proven in the procedure described above. In frame F_2, for example, $\mathit{effects}(Y)$ should only become true after the entire frame has been completed.

Table 6.5 provides formal definitions for the logical rules that LIESON agents need to dispose of to use the proposal-based frames defined above. They define *agent*, *can*, and *effects* in

$$
\begin{aligned}
&\forall x.self(x) \Rightarrow agent(x)\\
&\forall x.other(x) \Rightarrow agent(x)\\
&\forall x,y,s.agent(x) \wedge agent(y) \wedge x \neq y \wedge number(s) \wedge \neg\exists r.existsLink(x,y,r)\\
&\quad \Rightarrow can(x, addLink(x,y,s))\\
&\forall x,y,s.agent(x) \wedge agent(y) \wedge x \neq y \wedge number(s) \wedge \exists r(r \neq s \wedge existsLink(x,y,r))\\
&\quad \Rightarrow can(x, modifyRating(x,y,s))\\
&\exists r.existsLink(x,y,r) \Rightarrow can(x, deleteLink(x,y))
\end{aligned}
$$

Tab. 6.5: Logical rules for proposal-based frames.

terms of the primitive facts that occur in agents' knowledge bases ($number(x)$ to express that s is a valid numerical rating value, $self(x)$ to inform the agent that his name is x and $other(x)$ to denote that x is the name of some known peer).

Substitutions, condition construction and merging As for occurrence counters and substitutions these are initially empty and will only be filled with values during enactment/update of the respective frames. We should take a minute to explain how this is actually done. Suppose, for example, that the agent experiences the conversation

$$
\begin{aligned}
&\texttt{request}(\texttt{a}_0,\texttt{a}_1,\texttt{addLink}(\texttt{a}_1,\texttt{a}_2,0)) \rightarrow \texttt{accept}(\texttt{a}_1,\texttt{a}_0,\texttt{addLink}(\texttt{a}_1,\texttt{a}_2,0))\\
&\qquad \rightarrow \texttt{confirm}(\texttt{a}_0,\texttt{a}_1,\texttt{addLink}(\texttt{a}_1,\texttt{a}_2,0)) \rightarrow \texttt{do}(\texttt{a}_1,\texttt{addLink}(\texttt{a}_1,\texttt{a}_2,0))
\end{aligned}
$$

while using F_1 as active frame. Apart from the fact that the respective counters are incremented after this and the substitution

$$\vartheta_{new} = \langle [A/\texttt{a}_0], [B/\texttt{a}_1], [X/\texttt{addLink}(\texttt{a}_1,\texttt{a}_2,0))] \rangle$$

is added to $\Theta(F_1)$, two issues remain unresolved:

1. How should we extend the condition set $C(F_1)$ by a new c_{new} for ϑ_{new}?

   - Obviously, for $\sigma(\vartheta, F)$ (cf. equation 4.4, p. 104) to be computed in a reasonable way, the substitutions stored in a frame should only be considered in situations in which *at least* the physical actions that they involve would be executable. Therefore, we have to add *can* and *effects* statements to c_{new} for all physical actions that occurred in the newly experienced encounter.

     For instance, ϑ_{new} should only be relevant for computing the probability of some other substitution if $can(a_1, addLink(a_1,a_2,0))$ holds, and if the effects are made true after executing that action by proving $effects(addLink(a_1,a_2,0))$.
   - Apart from this minimal "condition construction" it is interesting to think about what other aspects of the current encounter should be additionally stored in c_{new}. These additional constraints would determine which past cases are

relevant for similarity calculations under different knowledge base states.

In particular, the interplay between selecting appropriate knowledge base elements for inclusion in c_{new} and defining appropriate encounter state abstractions (section 5.3.2) is very subtle: While encounter states determine the applicability of a frame F as a whole by virtue of the current distribution of Q values over states and frames, frame conditions determine which of the previous cases of F is relevant in the current situation (and with this, the probability distribution over Θ_{poss}).

However, to avoid complicating things further, and since we cannot provide any general guidelines on how to proceed in combining useful condition construction strategies with encounter state abstraction strategies appropriately we refrain from applying such advanced strategies here.

2. The reader may have noticed a subtlety regarding the pre-defined condition sets of F_1, F_2 and F_3, namely that these correspond to empty substitutions.

   With respect to the definition of $\sigma(\vartheta, F)$, this means that they do not play a role in similarity computation, which is quite reasonable as they do *not* represent past cases that should be taken into account when computing substitution probabilities. Looking at the definition of Θ_{poss}, though, which requires for all $\vartheta \in \Theta_{poss}(F, KB, w)$ that at least one of the condition sets of $C(F)$ is fulfilled under ϑ and the current contents of KB, this has further implications. It implies that, once a new ϑ_{new}/c_{new} is stored in the frame, none of the original "frame condition sets" of the frame (those with empty substitutions) need be true anymore, if c_{new} (or any of these "case condition sets") holds and ϑ extends ϑ_{new}. Thus, with time passing, the initial frame conditions will become less and less relevant, which can be a problem.

   To avoid such effects for the moment (which might obscure the results of our experiments), we additionally require that any possible substitution fulfils at least one of the (empty-substitution) "frame condition sets"

   $$\vartheta \in \Theta_{poss}(F, KB, w) \Rightarrow \exists i \leq |C(F)|.(KB \models C(F)[i]\vartheta \wedge h_\Theta[i] = 0)$$

   and ensure that any frame contains at least one such condition.

Encounter state abstraction In section 5.3.2, we explained that using the *theme* of a conversation to describe the state of a conversation is a useful heuristic to derive state abstractions that yield a manageable state space in the presence of a multitude of possible message contents and performative sequences.

In the context of linkage negotiations, we pointed out that appropriate definitions of the theme of a conversation should include (i) information about the linkage actions resulting from or occurring during a conversation, especially with respect to whether these are "positive" or "negative" link modifications and (ii) information about the role the agent has in such a conversation. To turn these intuitions into a concrete definition of encounter states, we use sets of generalised statements of the form

$$[\text{true}|\text{false}]{:}[\uparrow|\downarrow]([I|R], [I|R|T], [+|-|?])$$

to represent the physical actions talked about in an encounter. In such an abstract encounter state,

- true/false denotes whether the reasoning agent initiated the current conversation or not;
- $\uparrow$ and $\downarrow$ stand for a positive/negative link modification (where addition or a modification that will increase its rating value is considered a positive link modification, while link deletion or a rating modification that diminishes the link rating counts as a negative modification);
- I/R for the initiator/responder of the encounter, T for a third party (that might be referred to in an action talked about when an agent requests/offers modification of a link towards an agent who does not take part in the actual conversation);
- $+/-/?$ indicates whether the (reasoning) agent likes/dislikes/doesn't know the target site of the link modification, i.e. it denotes whether the private rating value towards the site that is the target of the respective link modification is rated with a non-negative ($+$), negative ($-$) value or has not been visited yet by the reasoning agent ($?$).

For example, if a_1 and a_2 talk about $\texttt{do}(a_1, \mathit{deleteLink}(a_1, a_3))$ in an encounter initiated by a_1 (while the learning agent a_2 is the responder and likes a_3's site, i.e. $r_2(a_3) \geq 0$) this is abstracted to

$$\{\text{false: } \downarrow (I, T, +)\}$$

If, in the same conversation, a_2 suggests to modify his own link toward a_1 (whom he does not like) from a rating value of 1 to 3, the state (viz *subject*) of the encounter becomes

$$\{\text{false:}\downarrow(I, T, +), \text{false:}\uparrow(R, I, -)\}$$

so that talking about several actions simply results in extending the set by another abstract action description. Note that using sets rather than lists also implies that two states are equal if the same types of actions are talked about in a different order, and that identical elements are collapsed into one, so that, for example, if $\mathit{modifyRating}(a_1, a_3, 2)$ and $\mathit{modifyRating}(a_1, a_3, 3)$ are the (only) two actions talked about in a conversation with ratings and roles as above, the resulting encounter state would simply be $\{\text{false:}\uparrow(I, T, +)\}$.

By applying this state abstraction, we reduce the maximal number of possible encounter states to[10] $2 \cdot (2 \cdot 4 \cdot 3)^2 = 1152$ while retaining enough information that is relevant for frame learning (as we will show below).

Agent ratings and utility benchmarks

Preference-wise, the agent population of ten is split into two groups, each comprising five agents: even-numbered agents $a_0, a_2, \ldots, a_8$ versus odd-numbered agents $a_1, a_3, \ldots, a_9$. Table 6.6 shows the precise distribution of private rating values for all agents in this scenario, but instead of describing the process by which they were designed, we shall simply

[10] At most two actions are talked about in a single encounter according to the frames defined for our basic experiments. Also, the agent is either initiator or responder in all actions in a description, and no agent can request an action from himself.

$r_s(t)$	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
a_0	**3**	0	1	-1	1	-1	2	-2	3	-3
a_1	-3	**3**	-2	1	-1	1	-1	2	0	3
a_2	3	0	**3**	-1	2	-2	2	-2	3	-3
a_3	-3	0	-2	**3**	-2	2	-1	2	0	3
a_4	3	0	1	-1	**3**	-1	2	-2	3	-3
a_5	-3	0	-2	1	-1	**3**	-1	2	0	3
a_6	3	0	1	-1	1	-1	**3**	-2	3	-3
a_7	-3	0	-2	1	-2	1	-1	**3**	0	3
a_8	3	0	1	-1	2	-2	2	-2	**3**	-3
a_9	-3	0	-2	1	-1	2	-1	2	0	**3**

Tab. 6.6: Private agent ratings in a ten-player population. For row s (source) and column t (target), each table entry denotes the opinion $r_s(t)$ agent s has of agent t.

discuss the most interesting aspects of this rating profile that are relevant to the experiments presented here:

- The general idea is that every "even" agent dislikes every "odd" agent and vice versa, while agents like those peers who belong to their own group. Thereby, "to like someone" means that the private rating value towards that agent is larger than zero, while "dislike" is expressed by rating values less than or equal to zero.
- a_0 and a_9 are the representatives of the two groups that are rated most extremely: they are highly popular among in-group peers, but extremely unpopular among out-group agents. These tendencies are also visible with decreasing strength in a_7, a_2 and a_8, while all agents are totally indifferent towards a_1 (except himself). The remaining agents a_3, a_4, a_5 and a_6 are "middle-of-the-road" agents who receive mediocre ratings from both their friends and their enemies.
- Each agent has the highest possible opinion of his own site (indicated by the "3" values along the diagonal of table 6.6 printed in bold face).

What is interesting about this distribution is not only the *heterogeneity* it induces on the agent population (which is a consequence of variations in agent preferences – no two agents have identical private ratings towards all other agents), but the behaviour of the function $score(a_i)$ used to compute the score of agent a_i (and to assess agent performance) under this rating profile.

Of course, it is almost impossible to derive optimal values for this function given the vast amount of possible linkage network configurations (with link ratings ranging from -3 to 3, each link can come in seven different kinds, and there are 90 possible edges in a directed graph with ten nodes).

However, some interesting benchmarks can be derived by looking at special link configurations, such as:

- The empty linkagent network G_0 that contains no edges at all,
- fully connected linkage networks G_{max} and G_{min} in which any two agents a_i and a_j are connected by a link with rating $r(a_i, a_j) = 3$ or $r(a_i, a_j) = -3$, respectively,

score	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	μ	σ
G_0	0.43	0.43	0.40	0.46	0.39	0.45	0.41	0.50	0.44	0.47	0.438	0.031
G_{max}	0.42	0.42	0.39	0.45	0.38	0.44	0.39	0.50	0.42	0.47	0.428	0.036
G_{min}	0.41	0.41	0.38	0.45	0.37	0.44	0.39	0.50	0.42	0.47	0.424	0.039
G_{rat}	0.75	0.58	0.59	0.65	0.65	0.76	0.72	0.72	0.74	0.80	0.696	0.071
$G_{rat,\geq 0}$	0.83	0.58	0.61	0.67	0.72	0.78	0.74	0.75	0.83	0.83	0.743	0.086

Tab. 6.7: Score benchmarks for simple experiments. The first ten columns (from left to right) show $score(a_i)$ values for each agent a_i under the respective linkage network; the two rightmost columns show the mean μ and standard deviation σ for the respective row.

- the fully connected rating-based linkage network G_{rat} in which there exists a link between a_i and a_j with rating $r(a_i, a_j) = r_i(a_j)$ for any two agents a_i and a_j, and

- the non-negative rating-based linkage network $G_{rat,\geq 0}$ in which all links (a_i, a_j) for which $r_i(a_j) < 0$ are omitted from G_{rat}.

Score results for all ten agents under these link graph structures are shown in table 6.7. The score distributions contained therein have very interesting properties which illustrate the complexity of the linkage optimisation problem. First of all, it is remarkable that G_0, G_{max} and G_{min} yield almost identical score results for any given agent. This means that laying maximal or minimal links blindly or not engaging in linkage activity at all will not help improve agent performance at all.

Secondly, and much more importantly, G_{rat} and $G_{rat,\geq 0}$ yield much higher payoffs for *all* agents than the other configurations. In some cases, these can be almost twice as high as those in G_0 (the empty linkage network that agents start out with). This suggests a delicate balance in the global distribution of utility: If all agents lay exactly those links that express their true opinion of others (or, in the case of $G_{rat,\geq 0}$, at least reveal which peers they like while concealing the truth about those they do not like), they are able to achieve an individually and globally much more desirable situation than if the image is blurred by too high or too low link ratings (average score is 0.743/0.696 versus 0.438/0.428/0.424).

So, there is actually an incentive for agents to reveal their true opinions of other members in the society, at least at first glance. However, for two reasons, things are not that simple. The first reason is that there is also an incentive to maximise the ratings *towards* oneself *ceteris paribus*. This is because, in most situations, an immediate increase in one's own score is experienced whenever a link is laid towards oneself. And this may of course conflict with the overall constraint of being as veracious as possible. The second reason is that, as the score values show, it is much better for (all) agents to hide information about which agents they do *not* like, as both average and individual scores are higher under $G_{rat,\geq 0}$ than under G_{rat}. That is, the "politically correct" strategy of only expressing positive opinions publicly dominates the "honest" strategy. This additionally complicates the learning and optimisation problem from an individual agent perspective, because LIESON agents are not endowed with explicit knowledge about these benchmarks, and they must learn how to behave optimally without any guidance other than their experience. The task is additionally aggrevated by the fact that agens are using the simplified score function which will only give them imprecise hints regarding the usefulness of certain actions.

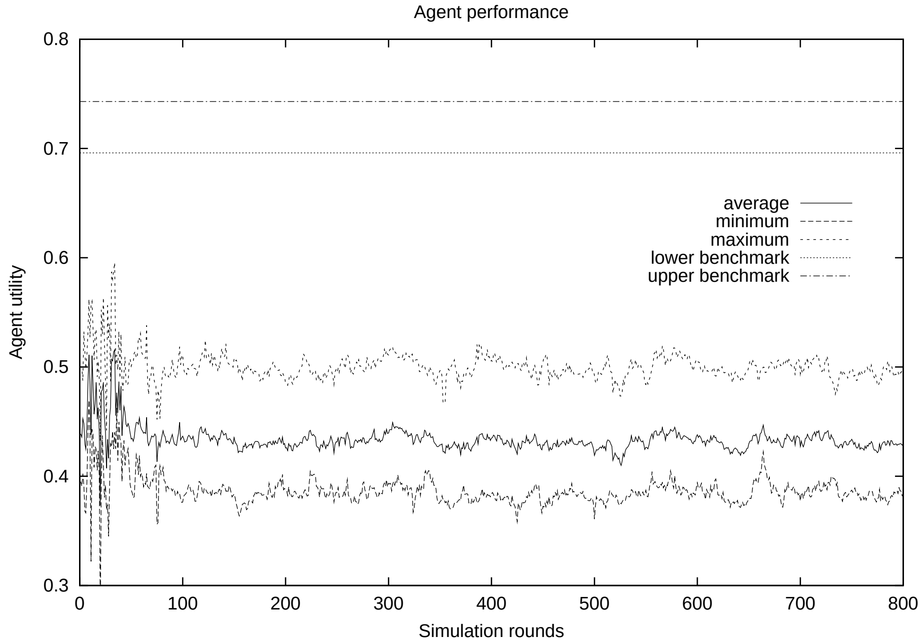


Fig. 6.3: Performance of randomly acting agents (single simulation run)

To sum up, the utility function together with the rating profile used in our experiments provide an interesting point of departure for analysing the behaviour of LIESON agents. In the following, we are going to use the average scores obtained with $G_{rat,\geq 0}$ and G_{rat} as benchmarks in the sense that we expect agent performance in the system to lie in or near the range of these benchmark values.

6.2.2 Results

Results with non-InFFrA agents

Randomly acting agents As a first experiment, we test the performance of randomly acting agents, i.e. agents that neither dispose of a BDI reasoning component, nor of m²InFFrA capabilities. These agents simply explore the environment with `update`- and `explore`-probabilities as shown in table 6.2, calculate their action options in each round and pick one of these actions randomly. No communication between agents takes place in this experiment. Figure 6.3 shows the average performance of these randomly acting agents. As in all subsequent performance plots, agent utility (i.e. their total score) as computed by the omniscient system manager is shown for a certain number of reasoning cycles, where one reasoning cycle is completed when all agents have performed their next action. If they choose to do nothing, they still have to notify the system manager that they have completed another iteration in their decision-making routine.

In all plots to follow, we include the average agent score that results from the benchmark configurations G_{rat} (0.696, labelled "lower benchmark") and $G_{rat,\geq 0}$ (0.743, labelled "upper benchmark") (see table 6.7). The plot in figure 6.3 shows the results of a single simulation run. It is useful to look at such single runs when we want to avoid equilibrating the variation in utility from round to round, as this happens when averaging over multiple

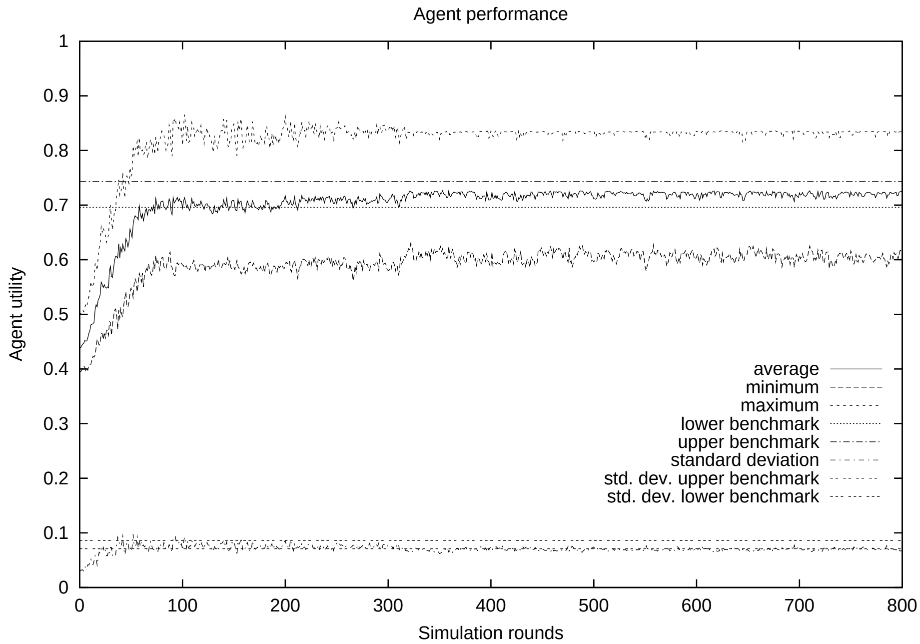


Fig. 6.4: Multi-run average performance of non-communicating BDI agents

runs. In most of the reported experiments below we will also show utility results that are averaged over 50 runs each.

As can be seen from this plot, agent performance is way below the benchmark values, and this is true of the average (calculated as the mean of all individual agents' scores), minimal and maximal agent scores (computed by comparing all agents' scores in each round). This illustrates that random action is of virtually no use in the LIESON scenario, thereby ruling out that coincidental link network configurations may contribute to the performance of BDI and m²InFFrA agents described below. It should also be remarked that even by continuing the simulation for 100000 rounds, not a single link configuration occurs that is even close to the benchmarks. In other words, it is impossible to achieve high scores by sheer coincidence.

Non-communicating BDI agents Next, we look at BDI agents who do not communicate with each other. These agents simply seek to optimise their own utility by performing local actions, and they do not have any means to influence what others do.

As the plot in figure 6.4 shows, these agents fare quite well in that they achieve linkage configurations somewhere between the "honest" and "politcally correct" average payoff. However, the impression one gets at looking at this *multi-run average* performance plot might be misleading.
Such a multi-run average compares 50 simulation runs and shows

- the average utility among all agents,
- the average maximal utility obtained by an agent, and
- the average minimal utility obtained by an agent,

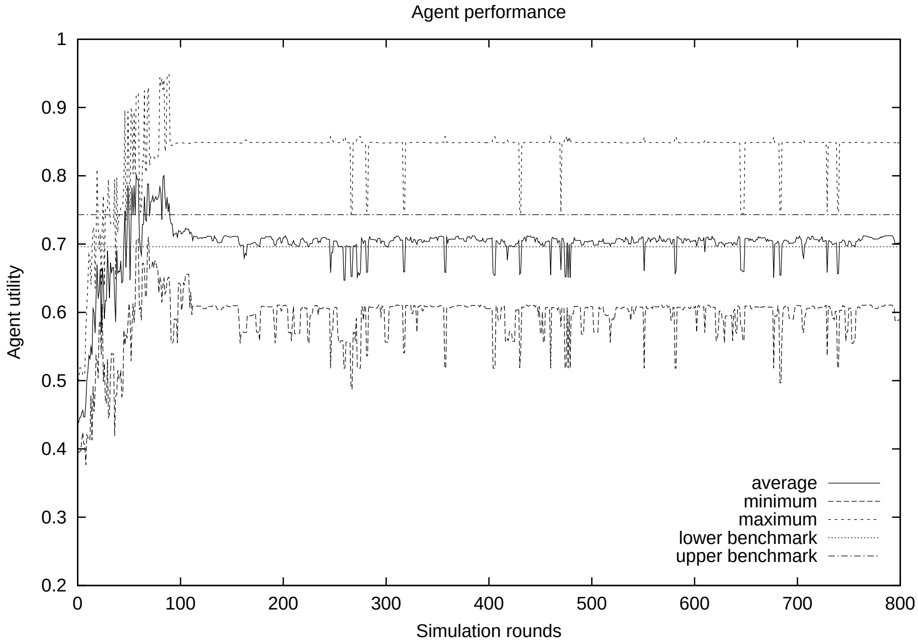


Fig. 6.5: Single-run performance of non-communicating BDI agents

in each round, averaged over 50 runs.[11]

What this averaging entails (which is normally very useful to ensure the reliability of simulation results), is that utility variations that take place at different points in time in different runs are averaged out. So a multi-run average curve will always look smoother than than a typical *single-run average*, which we will therefore also look at in many of the simulation result we discuss. The data used for such a plot is simply picked from one of the 50 runs, and it shows more realistically what happens in a single simulation.

For the case of non-communicating BDI agents, one such single-run plot is shown in figure 6.5. It shows that, in fact, the average agent utility still falls significantly below the benchmark values, and that this occurs quite often. Looking more closely into the simulation data, we can see that there are two cases in which agent utility suddenly drops:

1. Whenever the utility of the best-performing agent increases slightly above the long-term value (which is around 0.848), both the average worst agent's scores drop. This means that the "strongest" agents attempt to further improve their scores at the expense of others, whereupon these other agents react and things get back to normal.

2. When all three values suddenly decrease. This marks an obvious "mistake" of one agent that affects the entire society (it is very improbable that more than one agent performs a flawed action at a time). Agents are prone to make such mistakes from time to time, either (i) because actions become their top priority that were enqueued

[11] Note that this does not imply that these are the "best" or "worst" agents of each run, but that the maximal/minimal value is computed by comparing all agents' scores in each round. What we do to derive the maximum and minimum curves is to compute these values by comparing all agents' scores in a single run, and then to take the mean over 50 different such maximum/minimum curves. This means that the curves show the average performance of the "best" and "worst" agent among all agents across 50 runs.

under different circumstances and are now sub-optimal or (ii) because of the imprecision of the local utility-estimating function agents use (see section 6.1.2).

At the bottom line, this means that agent behaviour (and with it, agent performance) never truly converges, even if variations seem small in the multi-run average. The good news is that agents are capable of recovering from their mistakes and achieving a fairly balanced social distribution of utility. As a final point of analysis we should look at each agent's individual performance. For this purpose, figure 6.6 shows the performance of each agent across the 50 runs in what we will call a *multi-run individual average* set of plots. Each of these plots shows the best, average and worst performance of the same agent across the 50 runs of a simulation with identical parameter settings. The benchmark values for each agent (shown as straight lines that are the plots of constant functions) are taken to be the scores each agent would obtain under G_{rat} and $G_{rat,\geq 0}$, respectively (shown in the last two rows of table 6.7).

As can be seen from these plots, agents a_0, a_5 and a_6 do not attain a utility level that lies between the two benchmarks (on the average). On the whole, agents a_1, a_2, a_3 and a_7 exhibit the best performance (especially given with respect to their "potential" as indicated by the benchmark values). Agents a_4, a_8 and a_9 have a rather mediocre performance, especially in the worst case. No agent manages to stay within the bounds of the two benchmark values in the worst case. We shall compare these results to the performance of m²InFFrA agents further below.

Non-empirical communicating BDI agents Despite the fact that the experiments above are useful to assess the complexity of the LIESON system, they are not really comparable to m²InFFrA simulations for the fact that these experiments are devoid of any form of communication.

To evaluate the performance of m²InFFrA in coping with *stategic communication* (which is the very purpose of the architecture) we have to compare them with other types of communicating agents. Therefore, the third type of non-InFFrA agents whose performance we are going to analyse is that of *non-empirical communicating BDI agents* or "naive" communicating BDI agents. These agents employ the simple kind of communication described in section 6.1.2 (p. 154). This means that they send requests for an action to each other whenever the other's action appears profitable. In turn, when an agent receives such a request, he evaluates the utility gain the requested action would offer to him (as if he had thought of the action himself) and enqueues it in his BDI queue accordingly.

In a way, this is a very reasonable communication strategy, because agents are "fair" in the sense that they consider each action requested by someone as if it were their own. At the same time, agents will not execute any action that does not increase their score, so the strategy also safeguards against self-harming actions.

The reason it is called "naive" lies deeper. The problem with such communication is that agents do not take the *consequences* of their requests into account. They are neither notified about whether the requested action will be executed or not, nor do they check whether the action was executed after a while. Thus, there is no way of telling whether the request was successful. As we will see, this has a dramatic impact on performance.

What happens in societies of naive communicating agents is shown for a single simulation run in figure 6.7: After an initial phase in which agent performance increases fairly quickly, it suddenly freezes and there is no further improvement. In fact, looking at the

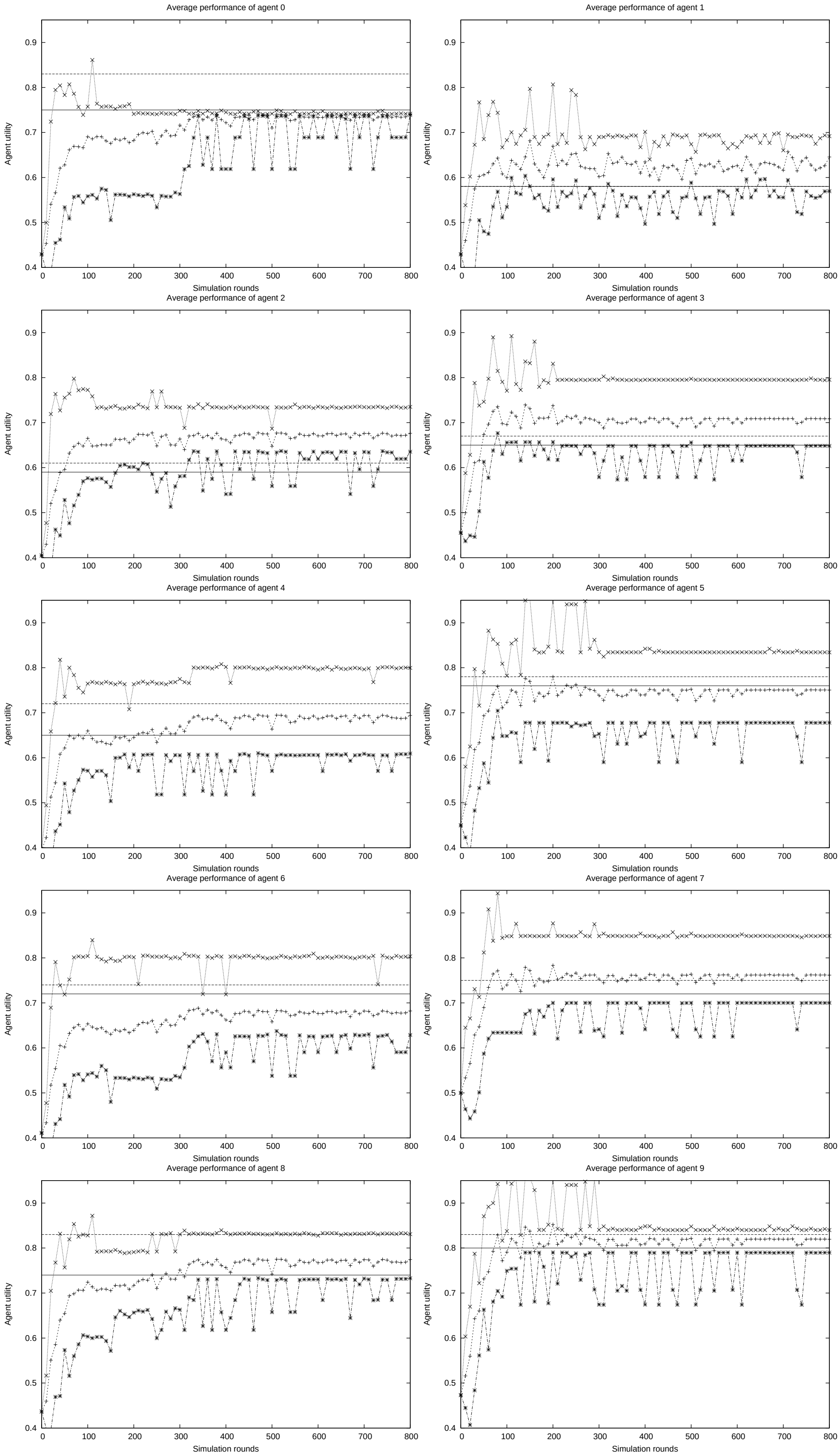


Fig. 6.6: Multi-run average performance of individual agents: Each plot shows average, maximal, and minimal performance of a particular agent for the case of non-communicating BDI agents, with curves and benchmarks as before

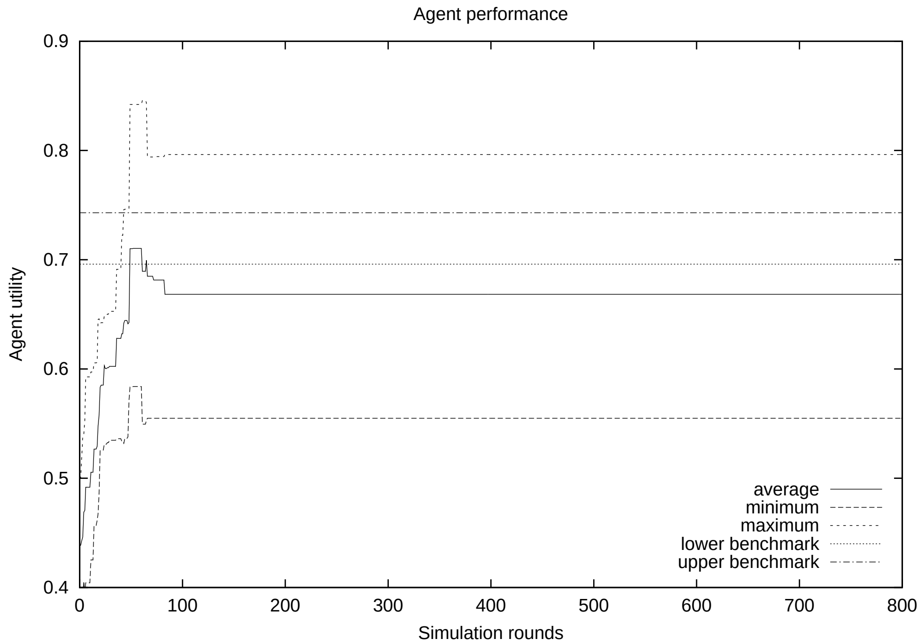


Fig. 6.7: Single-run average performance of non-empirical, communicating BDI agents

actions that are performed after this point, we can notice that link modification activity ceases completely, i.e. agents will not execute any further physical link additions, deletions or modifications after this point. Instead, they only issue requests to others, explore the environment according to the preset exploration probabilities or do nothing. The reason for this is the fact that agents prioritise actions in their BDI reasoning process according to the projected utility gain of these actions. As has been remarked in section 6.1.2 (p. 152), it follows from our definition of the utility function that adding or strengthening outgoing links is less profitable[12] than obtaining ingoing ones. Therefore, when agents estimate the utility of their own or others' possible future actions, it is only natural that there will always be some action another agent might perform which is more desirable than the actions the agent might perform himself. Intuitively speaking, this means nothing but that there will always be things someone else can do for oneself which are deemed more useful than what one can do for others.

During simulations with these naively communicating agents, it is only a question of time until all agents discover this fact and their BDI queue is replete with others' actions. After this, agents will stop executing others' requests or performing own link modification actions as both these types of actions are never rated more highly than the items that are already in the queue. At the same time, they will relentlessly issue requests towards other agents. This kind of behaviour results in completely halting link modification activity, and utility performance freezes at whichever level it happened to be in at the instant in which agents discover the superiority of others' actions over their own. This can also be

[12] This is not entirely true in the general case, as (i) there may be feedback loops that cause a link laid to someone else to increase one's own popularity via third party sites or (ii) agents may weigh the importance of others' popularity higher than their own. From an agent perspective, though, both these aspects are irrelevant; using the simple utility computation method, they only take in-edges from direct predecessors in the linkage network into account rather than general paths.

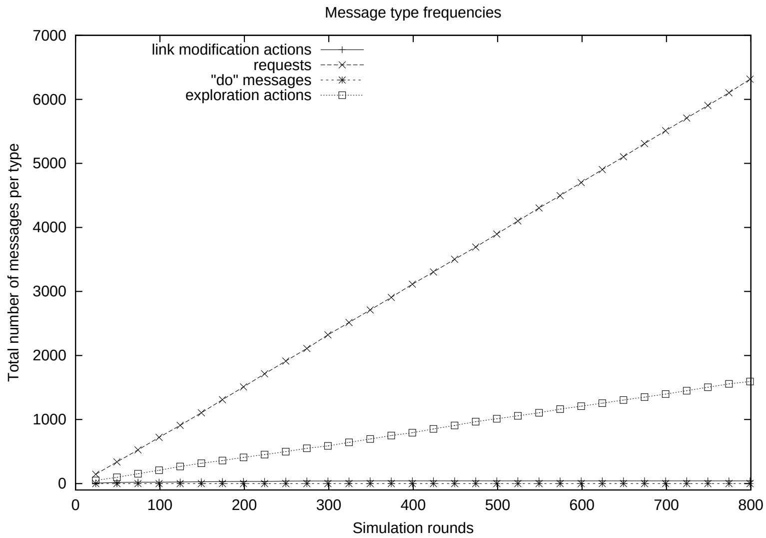


Fig. 6.8: Message/action type statistics for naively communicating BDI agents: cumulative number of linkage modification actions versus cumulative number of requests. The number of link-modifying actions converges to a very low value (43) after about 230 rounds while requests continue to be spawned ("do nothing" messages are not shown).

verified by looking at the different types of messages exchanged among agents over time. Figure 6.8 shows the total number of messages in the system for two different kinds of messages: physical link modifications (*addLink*, *deleteLink* and *modifyRating*) vs. requests for such actions.

As the multi-run average plot (figure 6.9) suggests, the overall performance of non-empirical communicating BDI agents is very poor. They do not prove capable of dealing with each other's autonomy when communicating, since they assume that each request they issue will be honoured by the other party. And they are unable to reason about their experience with previous communication to better estimate the likelihood of successful interactions. In comparison to non-communicating BDI agents, we can see that communication does not always improve the capabilities of a multiagent system. Quite the contrary is the case – communication can even severely limit them if used in a naive way in systems that allow for complete agent autonomy! In the following section, we are going to describe how m²InFFrA agents overcome this problem.

Results with m²inffra agents

The analysis of m²InFFrA agents falls into two parts:

1. A general evaluation of individual and global performance with particular attention to different *desirability test* strategies as these prove to have an enormous impact on performance and also help to understand how and why the approach works.

2. A critical investigation of the contribution of *frame learning* to the overall performance of the system which will elucidate the effects of adaptive frame selection.

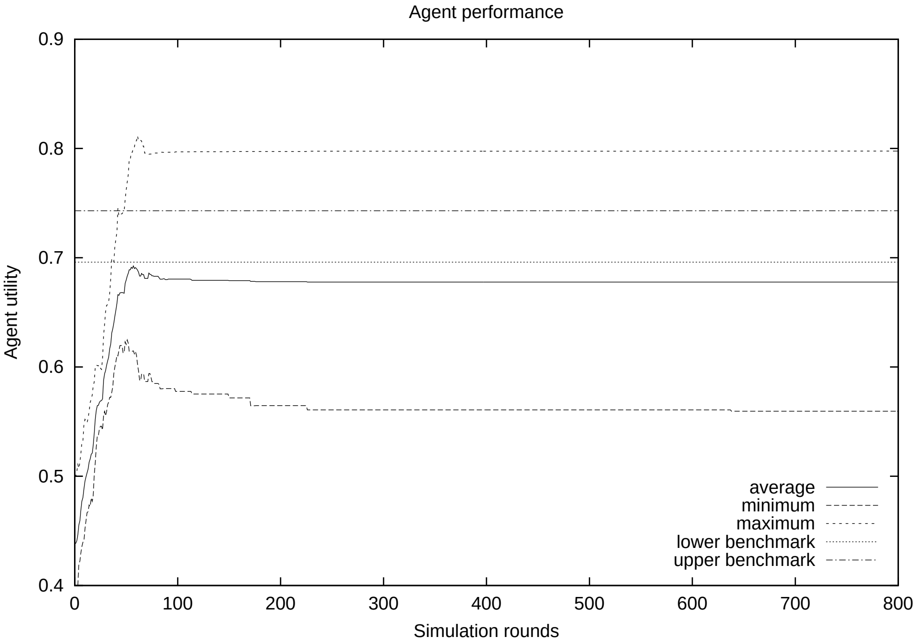


Fig. 6.9: Multi-run average performance of non-empirical, communicating BDI agents

Apart from using an m²InFFrA component inside each agent, the experiments to follow are exactly identical to those reported above, i.e. we use the same rating profiles, internal parameters and evaluation procedure.

The first thing we want to do in our evaluation is to look at the overall behaviour of m²InFFrA agents to see if our algorithms ensure effective interaction management and learning in general terms. After verifying that the basic desired functionality (applying the frames of table 6.4 (p. 164) properly, proving their conditions, selecting optimal frames and substitutions, long-term merging of frames) is in place, it was discovered that the central parameter that affects system performance is the employed *desirability test strategy*.

What we mean by this is the strategy with which an agent determines whether a selected frame and/or individual action is desirable. In m²InFFrA, specifying this strategy amounts to deciding

1. whether to check for (i) *action-level desirability* during application of a frame after an optimal substitution has been selected (step ❺ in figure 5.3, p. 132), (ii) to verify *frame-level desirability* upon Q-based frame choice (transition from step ❻ to step ❾ in the same figure), (iii) none of these (i.e. to simply always pick the best frame and substitution regardless whether they are desirable at all) or (iv) both.

2. which desirability criterion to apply. Here, we distinguish between the *strict* criterion according to which an encounter postfix is only considered desirable if it is expected to increase the agent's total current score, and the *lenient*, entropy-based criterion which we defined in section 5.3.4. For the latter, we can also distinguish between different "entropy corridors" by weighting the right hand side of equation 5.19 (p. 143) with different constant factors, i.e. using the equation

$$b = -\lambda \cdot \Delta\mathcal{E}_{\mathcal{F}}(\varepsilon, postfix(T(F), w)) \tag{6.7}$$

for some $\lambda \in \mathbb{R}$ (where, previously, $\lambda = 1$ held throughout).

Given that the desirability test strategy determines whether an active frame will be complied with or whether the candidate frame that is optimal according to Q-value maximisation will be activated at all, it is clear that the choice of this strategy bears strong implications on agent behaviour.

Figure 6.10 shows a comparison of the four strategies described above for the lenient desirability criterion ($\lambda = 1$). From top to bottom, the plots are arranged in order of increasing "strictness", i.e. no desirability test (NDT), action-level test (ALT), frame-level test (FLT), and desirability test at both levels (A&FLT). In each row, a multi-run average performance plot is depicted in the left column and a single-run average performance plot is shown on the right hand side.

The first thing to observe here is that a clear distinction can be made between the performance of NDT and ALT agents (top two rows in figure 6.10) on the one hand and that of the other two types of agents shown in the diagram on the other. While the latter converge pretty soon to a stable performance level, performance of agents who use the former strategies keeps changing, even if a long-term improvement is visible in the multi-run average. In fact, `do`-actions and BDI-level link modifications cease almost completely (and with them, score change) after a while if agents perform a desirability test at any level. Also, the total number of `do`-actions ranges between 10 and 20 for FLT and A&FLT (for a total of about 2000 requests[13]) while about 300-400 requests lead to some kind of physical `do`-action in the case of NDT or ADT.

NDT agents To begin with, we evaluate the performance of NDT agents who neither perform a desirability test at the action- nor at the frame-level. This kind of m²InFFrA agent selects that frame/substitution which is optimal according to the Q-table and the similarity-based in-frame action predictions, the effect of the lacking desirability test being that he will never deviate from an existing frame or refuse to activate a frame for desirability reasons. Deviance can only occur for adequacy or validity reasons (i.e. if an unexpected peer message is perceived, the remaining actions cannot be executed of frame conditions do not hold). So if asked to do something which they can do, these agents will attempt to respond in an optimal way, but they are forced to either accept the request or make a counter-proposal. Clearly, it can by no means be ensured that a profitable alternative is available even in the case of counter-proposals (let alone in the case of accepting the request as is).

Therefore, these "naive" agents cannot avoid encounters which decrease their score, a fact for which the single-run average provides evidence. However, for those states in which they can make a choice between different frames, agents should be able to learn which frame has been more profitable in the long run. If a certain choice was better in the past it should be chosen more frequently in the future, since the long-term rewards are stored in the Q-table. In particular, making a counter-proposal should mostly be better than accepting the request as issued, as it enables the agent to suggest a more profitable action he can perform instead (F_2 in table 6.4) or even to try to get the other to do something profitable in return (F_3).

In theory, it could be argued that they are also occasionally "lucky" because others will do what they want. However, it would be very disappointing if this is the reason for the

[13] Actually, the total number of requests issued in the whole simulation is around 3000, but about one third of them is directed towards agents who are already busy conversing with someone else.

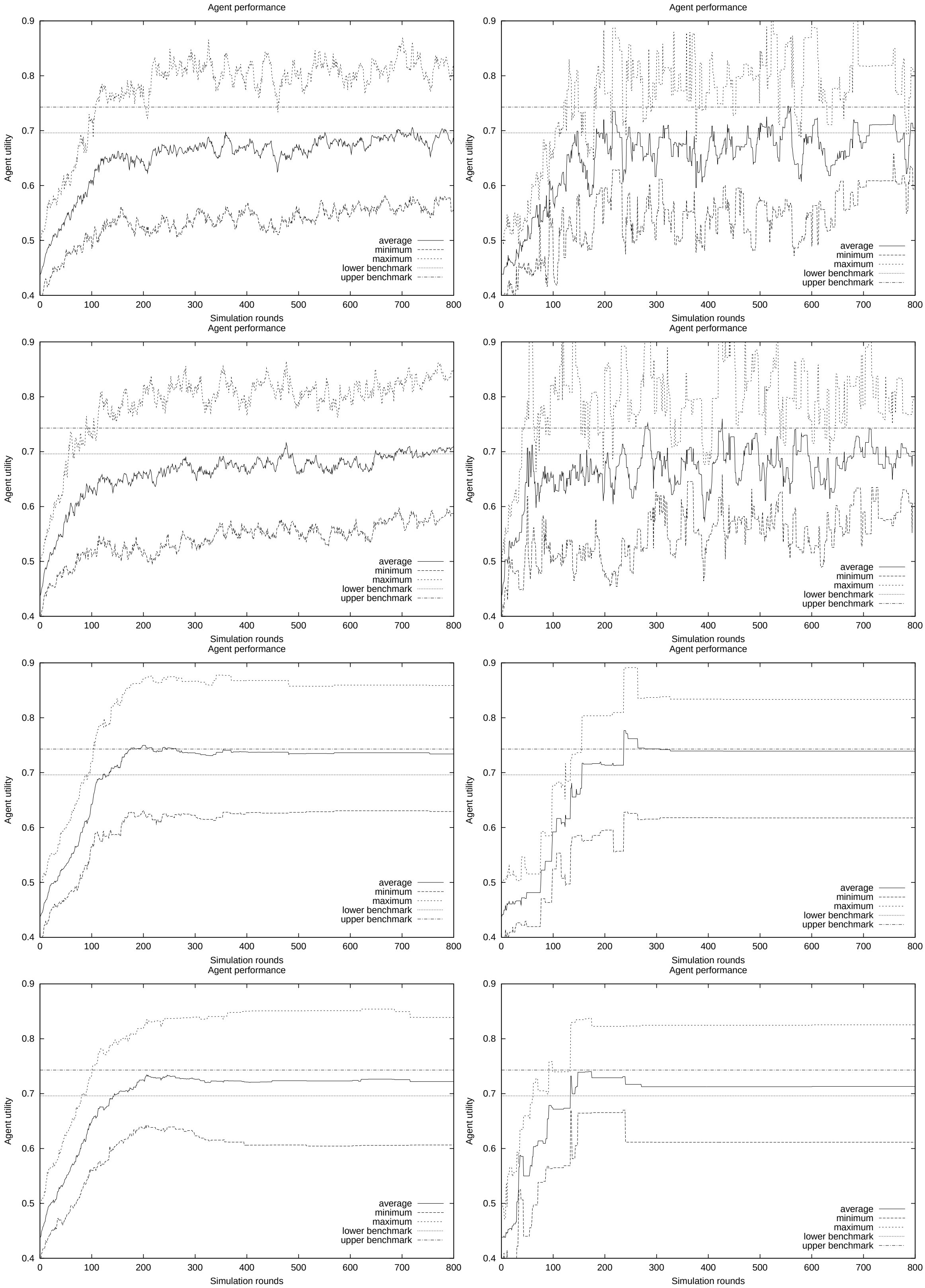


Fig. 6.10: Comparison between different desirability test strategies using the entropy-based desirability criterion. From top to bottom: NDT, ADT, FLT, and A&FLT agents

communicative link modification	*BDI "undo" action*
(00:12) $\texttt{do}(a_0, addLink(a_0, a_5, 0))$	(00:30)$\texttt{do}(a_0, deleteLink(a_0, a_5))$
(00:23) $\texttt{do}(a_0, addLink(a_0, a_3, -3))$	(00:31) $modifyRating(a_0, a_3, -1)$
(00:58) $\texttt{do}(a_0, addLink(a_0, a_5, 1))$	(01:11) $deleteLink(a_0, a_5)$
(01:44) $\texttt{do}(a_0, addLink(a_0, a_8, -1))$	(02:26) $modifyRating(a_0, a_8, 3)$
(01:48) $\texttt{do}(a_0, addLink(a_0, a_9, 1))$	(02:25) $modifyRating(a_0, a_9, -3)$
(02:56) $\texttt{do}(a_0, addLink(a_0, a_2, 2))$	(05:27) $deleteLink(a_0, a_2)$
(04:31) $\texttt{do}(a_0, addLink(a_0, a_1, 0))$	(05:39) $\texttt{do}(a_0, modifyRating(a_0, a_1, 3))$
(04:52) $\texttt{do}(a_0, modifyRating(a_0, a_8, 0))$	(05:13) $modifyRating(a_0, a_8, 1)$
(05:07) $\texttt{do}(a_0, addLink(a_0, a_7, 0))$	(05:18) $modifyRating(a_0, a_7, -2)$
(05:46) $\texttt{do}(a_0, modifyRating(a_0, a_4, 0))$	(06:07) $deleteLink(a_0, a_4)$
(05:54) $\texttt{do}(a_0, modifyRating(a_0, a_6, -3))$	(06:03) $modifyRating(a_0, a_6, 3)$
(06:12) $\texttt{do}(a_0, addLink(a_0, a_9, 0))$	(06:15) $deleteLink(a_0, a_9)$
(06:18) $\texttt{do}(a_0, modifyRating(a_0, a_3, -3))$	(06:19) $modifyRating(a_0, a_3, -1)$
(07:22) $\texttt{do}(a_0, addLink(a_0, a_9, 3))$	(07:37) $modifyRating(a_0, a_9, -3)$
(07:34) $\texttt{do}(a_0, modifyRating(a_0, a_7, 1))$	(07:35) $deleteLink(a_0, a_7)$
(07:50) $\texttt{do}(a_0, addLink(a_0, a_3, 0))$	(07:58) $modifyRating(a_0, a_3, -1)$
(08:14) $\texttt{do}(a_0, deleteLink(a_0, a_3))$	(10:38) $\texttt{do}(a_0, addLink(a_0, a_3, 0))$
(08:19) $\texttt{do}(a_0, modifyRating(a_0, a_2, 2))$	(08:27) $deleteLink(a_0, a_2)$
(08:38) $\texttt{do}(a_0, addLink(a_0, a_5, 1))$	(08:48) $modifyRating(a_0, a_5, -1)$
(11:06) $\texttt{do}(a_0, modifyRating(a_0, a_2, 2))$	(11:09) $modifyRating(a_0, a_2, 1)$
(11:36) $\texttt{do}(a_0, modifyRating(a_0, a_1, 2))$	(11:38) $modifyRating(a_0, a_1, 0)$
(11:40) $\texttt{do}(a_0, addLink(a_0, a_7, -3))$	(14:15) $deleteLink(a_0, a_7)$
(11:48) $\texttt{do}(a_0, modifyRating(a_0, a_4, 0))$	(12:11) $deleteLink(a_0, a_4)$
(11:53) $\texttt{do}(a_0, deleteLink(a_0, a_5))$	(12:43) $\texttt{do}(a_0, addLink(a_0, a_5, 2))$
(13:04) $\texttt{do}(a_0, addLink(a_0, a_1, -1))$	(13:17) $deleteLink(a_0, a_1)$
(14:06) $\texttt{do}(a_0, addLink(a_0, a_1, 0))$	(18:44) $deleteLink(a_0, a_1)$
(14:46) $\texttt{do}(a_0, addLink(a_0, a_7, 0))$	(15:00) $deleteLink(a_0, a_7)$
⋮	⋮

Tab. 6.8: The "undo" effect of BDI-level optimisation. The left column shows the link modification action that occurred in a conversation (do-performative), and the first subsequent action that modified the same link can be seen on the right column. Simulation round numbers are shown in minute:second format.

long-term performance improvement. If other agents in the system are *not* so naive with respect to desirability, they could definitely exploit the naive desirability agents! With this respect, one thing we can show is that recovering from local utility minima is not due to this kind of "luck", but to the *undo effect* of the BDI layer. Table 6.8 illustrates this effect: It shows (an excerpt of) the list of all executed physical actions of the single simulation run in the figure above, while juxtaposing each do-action (that is the result of some social agreement) with the first subsequent action that modified the same link. From the short number of rounds that lies between the communicative link modification and subsequent modifications to the same link and from the fact that most of these subsequent modifications are the result of BDI-level decision-making (as they are not enclosed within a do-performative) we can see that the BDI-level optimisation is trying to make up for the utility loss incurred by the m²InFFrA layer.

But the fact that the BDI layer tries recover from m²InFFrA-caused utility losses does not tell us anything about the effects of learning. Fortunately, we can show that agents

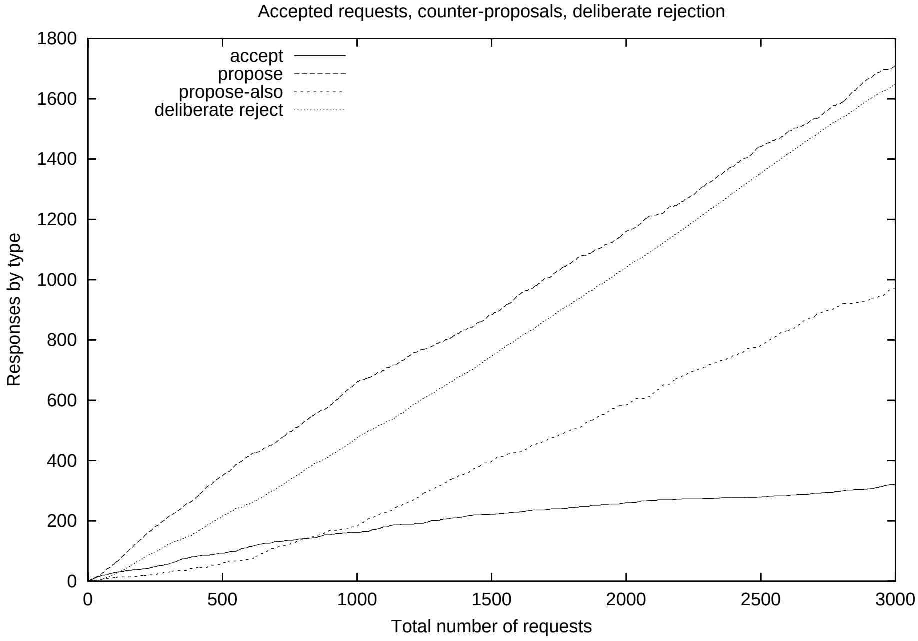


Fig. 6.11: m²InFFrA agents with naive desirability test strategy: Comparison between cumulative number of `accept`, `propose` and `propose-also` messages compared to the total number of satisfiable requests (i.e. all those requests that can be executed by the requestee in physical terms)

do learn that certain frames are more desirable than others. Figure 6.11 compares the cumulative amount of `accept`, `propose` and `propose-also` messages exchanged among all agents in the previous simulation run, which unambiguously stand for activation of F_1, F_2 and F_3, respectively. It can be seen from this figure that the frequency (i.e. the gradient of the cumulative curve) with which F_3 is selected increases while F_1 is decreasing with F_2 remaining fairly stable. This supports our intuition that F_3 and F_2 are more desirable in the long-term if agents *have* to use one of the three frames.

The figure also reveals a more subtle effect of learning, which can be seen from the cumulative number of *deliberate* reject-messages also shown in this plot. These deliberate rejections occur when agents pick a frame that results in a reject-action due to Q-value exploitation, and not because they cannot execute the requested action.

How is it possible that agents deliberately refuse to perform an action, if they are forced to pick a repository frame regardless of whether it is currently desirable for them or not? The reason is that once agents reject a requested action because they are not capable of performing the requested link modification (or perceive such a refusal), this "broken" frame is stored in the repository (see also section 6.1.2, p. 156). Then, this broken frame becomes an ordinary option that is considered in every subsequent framing decision. Eventually, such broken frames will attain a reasonably high Q-value by averaging over all experiences in which they were more useful than F_1, F_2 or $F3$, and with sufficient generalisation through merging over time, they may become the preferable choice in many encounter states. What is obvious from figure 6.11 is that the frequency of these deliberate rejections is increasing, thus adding a second element of adaptiveness to the behaviour of m²InFFrA agents: Apart from learning that making counter-proposals is useful, these agents are also capable of learning that deliberately rejecting an undesirable request is advantageous in many situations. Taken together, both these capacities enable agents to improve their average performance in the long term, even if they still make occasional mistakes.

At the same time, we should not forget that despite the fact that this performance is quite good, the exploitability of these agents is inacceptable from a decision-theoretic perspective. Also, the overall performance levels reached lie around the lower benchmark, which certainly leaves room for further improvement.

ADT agents Quite surprisingly, the plots under the NDT agent performance graphs in figure 6.10 suggest that agents who perform an action-level desirability test (only) are not capable of doing any better than the naive NDT agents. To understand this observation, let us describe this desirability test strategy once more: What ADT agents do is to activate a frame according to the Q-table without testing for desirability. After this, they will initiate a re-framing procedure if the desirability of the remaining actions is too low, and this is checked for in every InFFrA iteration.

Although it may seem that by this process they would avoid performing undesirable actions, what actually happens is that, in fact, most selected frames *fail* this action-level desirability test, thus forcing the agent to select an alternative frame. However, the new frame itself is selected without any assessment of profitability, so that they will continue the encounter with any frame that is considered appropriate according to the long-term utility estimate stored in the Q-table. This process continues in subsequent framing iterations and this results in the fact that ADT agents will actually never stop an encounter unless the same conditions occur that would force an NDT agent to give up. In a typical run, although there are only 24 framing cycles in which the optimal continuation is considered desirable while it is considered undesirable in 284 framing cycles (across all agents), agents still perform a total of 302 `do`-actions[14] which is very similar to the 330 `do`-actions of a comparable NDT simulation with a total of about 2000 requests.

That said, ADT agents *do* perform slightly better than NDT agents as can be seen from the above plots, which is probably due to the fact that at least they are able to spawn a re-framing procedure at the action selection level if even the best continuation seems unprofitable while NDT agents do not have this option and must rely on Q-value based optimisation.

FLT and A&FLT agents The plots shown in the bottom two rows of figure 6.10 depict a behaviour that is fundamentally different from that of NDT and ADT agents just described.

Firstly, as mentioned above, the number of physical actions that result from social agreements decreases dramatically. Especially after a certain performance level has been reached, further actions occur only very rarely. Obviously, the reason for this is that agents never select a frame whose trajectory postfix is not desirable under the optimal substitution. So, as their score increases, there are fewer and fewer proposals that they can accept or make a counter-proposal for that is expected to improve their utility standing.

Secondly, and much more importantly, (A&)FLT agents converge to a much higher global performance level, which lies clearly above the lower utility benchmark. Especially with respect to non-empirical communicating BDI agents and NDT/ADT agents, this is a significant advantage of introducing a frame-level desirability test.

[14] This number is higher than 284, but remember that F_3 involves two `do`-actions per encounter.

A comparison between FLT and A&FLT also reveals that an additional action-level desirability test (that is very time-consuming, anyway) actually does not add to the global performance in combination with frame-level Q-learning. From this we can conclude that omitting the action-level desirability test (step ❺ in figure 5.3, p. 132) from the framing process altogether is a reasonable choice.

Applying different desirability criteria Using the FLT strategy, agents are not forced to execute undesirable actions, and they need not wait for long-term reinforcement learning adaptation to improve their standing. This can be seen as a safe strategy, especially if agents are not going to interact very often in the future or if actions cannot be undone. On the other hand, this is a situation in which agents can opt out of any agreement at any point in time, so that learning leads to global "maximin" behaviour in the sense that agents seek to avoid "worst-case" damage but fail to identify further potentials for cooperation. The fact that the average score of these agents over time is not any higher than that of non-communicating agents as shown in 6.4 supports this claim.[15] Note, however, that m²InFFrA agents clearly outperform non-empirical, communicating BDI agents, cf. figure 6.9 (p. 179).

Sometimes, as shown in the single-run example of the FLT strategy in figure 6.10, the average utility can even significantly exceed the upper benchmark value. Apparently, though, this highly *cooperative* behaviour is not maintained in the long run, which ultimately means that agents fail to recognise a cooperative pattern of behaviour that they have already observed.

An analysis of the application of different desirability *criteria* is useful to identify the reasons for this problem. Figure 6.12 shows results for FLT agents that apply four different desirability criteria determined by different choices of the parameter λ in equation 6.7. From top to bottom, we experiment with

1. a *very lenient* desirability criterion where $\lambda = 2$, according to which the total entropy reduction is considered twice as important as in the experiments so far,

2. the *lenient* desirability criterion $\lambda = 1$ that has been used throughout the above experiments with m²InFFrA agents,

3. a *decreasingly lenient* criterion which uses discounting to progressively reduce an initial value of $\lambda = 2$ to $\lambda = 0$ in the limit (more specifically, in simulation round i, $\lambda = 0.99^i \cdot 2$), and

4. a *strict* criterion by which $\lambda = 0$ and agents will not trade off entropy reduction against immediate utility gain at all.

As can be seen from the plots in figure 6.12, it turns out that the choice of which criterion to apply bears strong implications on global system behaviour: $\lambda = 0$ is clearly inferior to all other criteria, as it precludes the establishment of communication patterns which might ensure an average payoff that lies significantly above the lower benchmark value.

[15] The performance of these m²InFFrA agents *is* actually better if we compare m²InFFrA results to the single-run example of figure 6.5 given that FLT agents don't make the occasional "mistakes" observed there. However, this is only due to the fact that we have imposed no restrictions on the number of suffixes that result from Θ_{poss}, while the BDI layer can only store ten actions in its queue and predict the utility of 23 actions per round. For this reason, we cannot claim that m²InFFrA is superior in this respect.

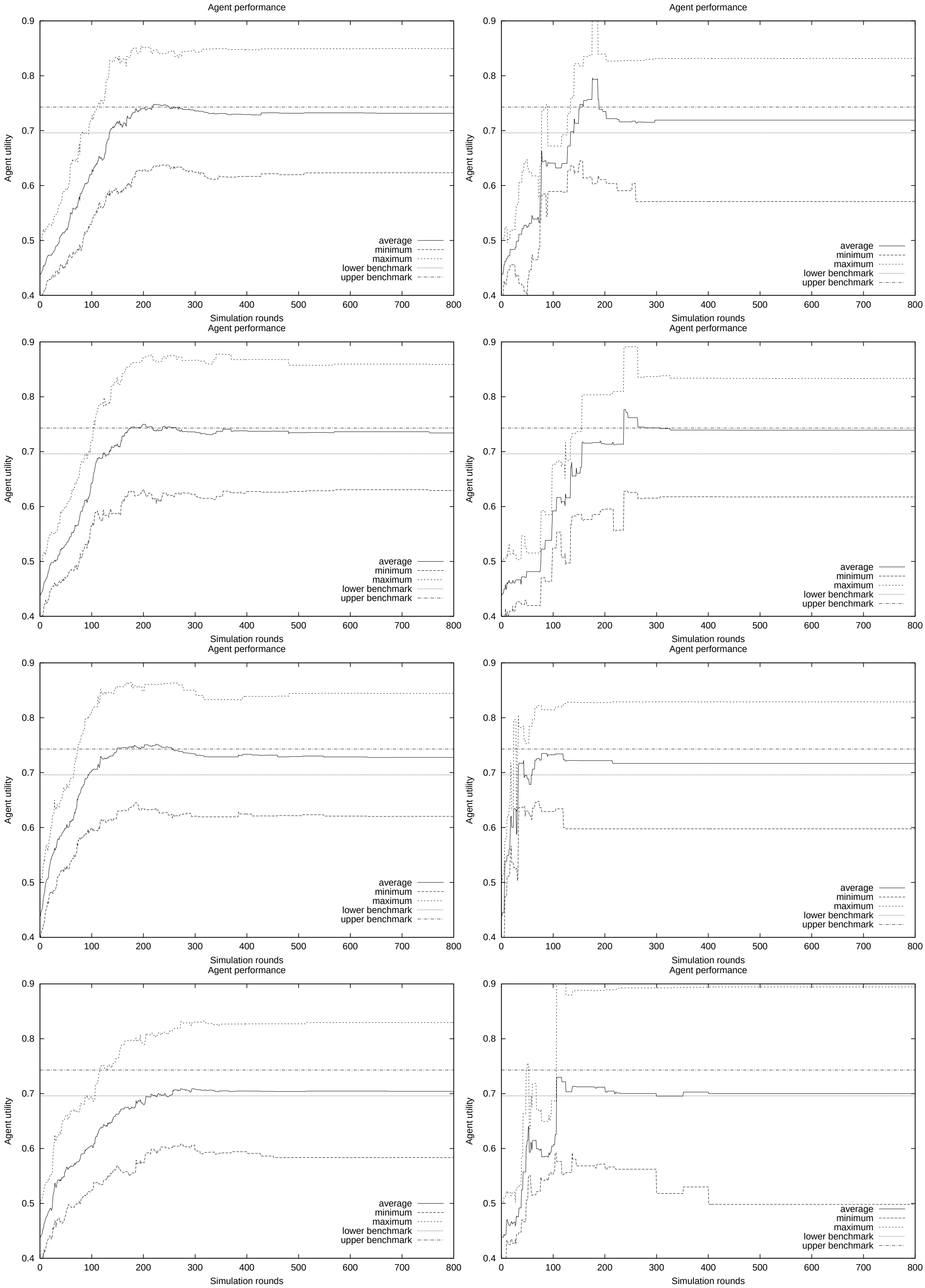


Fig. 6.12: Comparison between different desirability criteria. From top to bottom: very lenient ($\lambda = 2$), lenient ($\lambda = 1$), decreasingly lenient ($\lambda = 0.99^i \cdot 2$) and strict ($\lambda = 0$) criterion

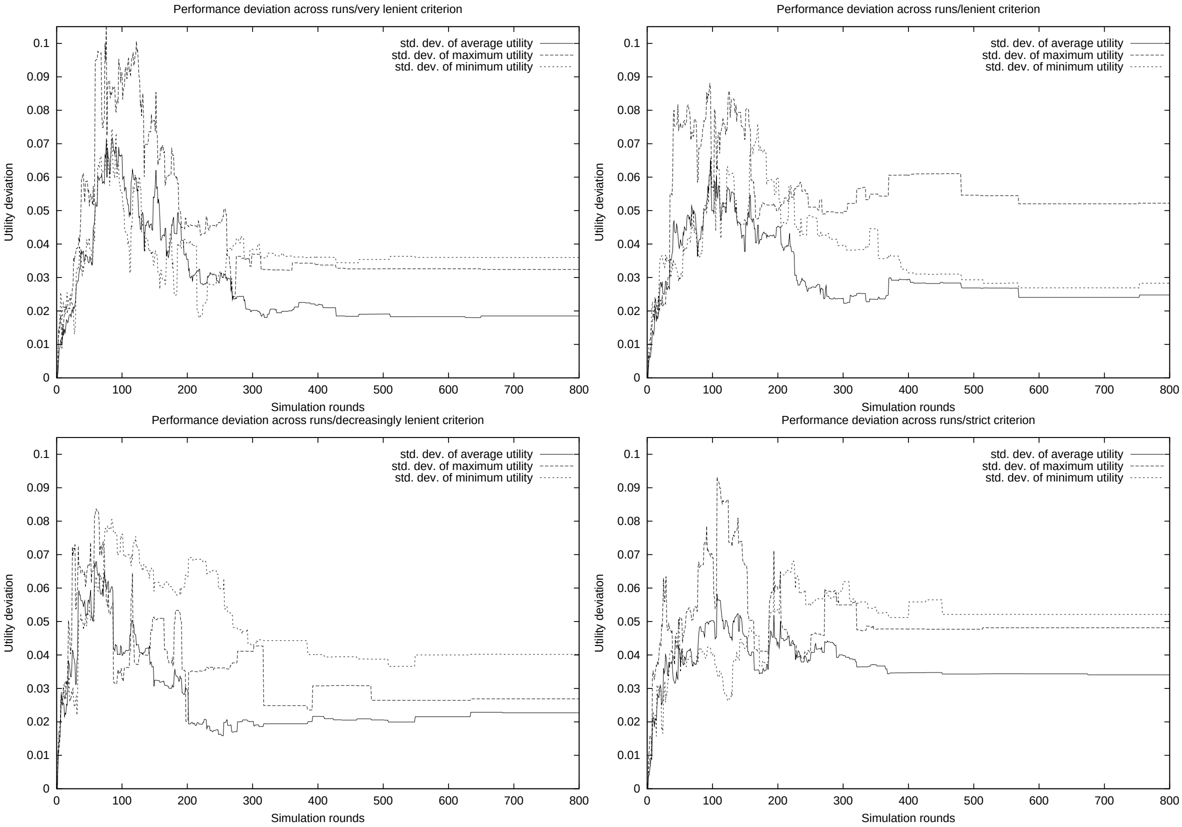


Fig. 6.13: Comparison between multi-run standard deviation for different desirability criteria: very lenient ($\lambda = 2$, top left), lenient ($\lambda = 1$, top right), decreasingly lenient ($\lambda = 0.99^i \cdot 2$, bottom left), and strict ($\lambda = 0$, bottom right) criterion

At the same time, this criterion fosters exploitation, as the difference between best and worst agents is much higher than is the case with more lenient criteria. The single-run performance in the bottom row of figure 6.12, in which the difference between maximum and minimum agent scores reaches almost 0.4 provides a striking example for this phenomenon.

Agents who apply the other three (lenient) criteria perform almost equally well so that, at first glance, it is not obvious where the difference between them lies. To make this difference visible, we need to look at the *standard deviation* of maximum, minimum and average utility between different runs in each multi-run simulation. For this, *multi-run standard deviation* plots are shown for each of the four desirability criteria in figure 6.13. These plots depict the standard deviation of the performance of the best, average, and worst agent across the 50 individual simulations conducted with each desirability criterion. Hence, they allow for an assessment of the variance in maximum, mean, and minimum utility achieved in each of these experiments. In other words, we can use such measurements to evaluate how *certain* the attainment of particular scores is. The relationships between the different quantities of standard deviations are summarised in the following table, where $+$, $\bigcirc$ and $-$ stand for comparatively large, moderately high and small values of the standard deviation, respectively:

Measure	$\lambda = 2$	$\lambda = 1$	$\lambda = 2 \ldots 0$	$\lambda = 0$
σ(minimum)	$\bigcirc$	$-$	$\bigcirc$	$+$
σ(average)	$-$	$\bigcirc$	$\bigcirc$	$+$
σ(maximum)	$-$	$+$	$-$	$+$

The strict criterion obviously does not provide any reliable guarantees, as the standard deviation is high both with respect to worst-case (σ(minimum)), average-case (σ(average)), and best-case (σ(maximum)) performance. This means that neither the best nor the worst or the average agent can rely on a stable performance across different simulation runs.

To ensure a safe *best-case* performance, it seems advisable to start out with a tolerant desirability criterion, as the values for $\lambda = 2$ and the "progressively strict" criterion $\lambda = 2 \ldots 0$ illustrate. This seems natural, as cooperative communication patterns require an initial cooperative stance to be established. However, $\lambda = 2 \ldots 0$ evolves into a more strict desirability criterion over time, and this results in a mediocre level of stability regarding worst-case and average-case performance, as we cannot be entirely sure that those cooperative high-utility patterns will have been established soon enough before the agent becomes more risk-averse. $\lambda = 1$, on the other hand, is only moderately successful in ensuring a decent average payoff, and even less so with respect to maximum payoff, which seems to suggest that $\lambda = 2$ is the optimal choice. Yet, the fact that the standard deviation of worst-case performance is higher as compared to $\lambda = 1$ when using this very lenient desirability criterion reveals that agents run a risk of being overly cooperative and hence potentially exploitable if $\lambda > 1$.

All this taken together provides us with a comprehensive picture of the role desirability criteria play in the balance between avoiding exploitability and fostering cooperation. In any case, what this analysis also shows is that our entropy-based desirability heuristics (section 5.3.4) provide a valuable instrument to fine-tune the social attitude of agents.

Returning to our original observation of the inability of m²InFFrA agents to maintain global utility levels above the upper utility benchmark, we have to state that even the most lenient criterion is not capable to bring about an evolution of cooperation, i.e. we cannot claim that m²InFFrA ensures perfect social coherence. Obviously, this is the price for ensuring strategic, non-exploitable action at the level of the individual. And this effect is not surprising, as achieving such individually rational behaviour is the primary objective of developing an architecture like m²InFFrA.

The contribution of learning As a final point of analysis in this series of experiments, we should address the contribution of frame-level Q-learning to the overall performance of m²InFFrA agents. For this purpose, figure 6.14 shows multi-run averages and single-run examples of simulations in which frames are chosen *randomly* from all matching, executable frames in the repository instead of using Q-values to determine the most suitable frame. The figure summarises the results of experiments with the same configurations as those of figure 6.10 (p. 181), i.e. the NDT, ADT, FLT and A&FLT strategies (in the same order as before). The performance plots confirm our previous observations regarding the learning capabilities of m²InFFrA agents which lie in learning the usefulness of counter-proposals and rejection are valid regardless of desirability test strategy. As depicted in these plots, omitting the frame-learning functionality leads to performance deterioration in all cases. This is a central result that justifies our efforts to add frame-learning capabilities to the m²InFFrA decision-making model.

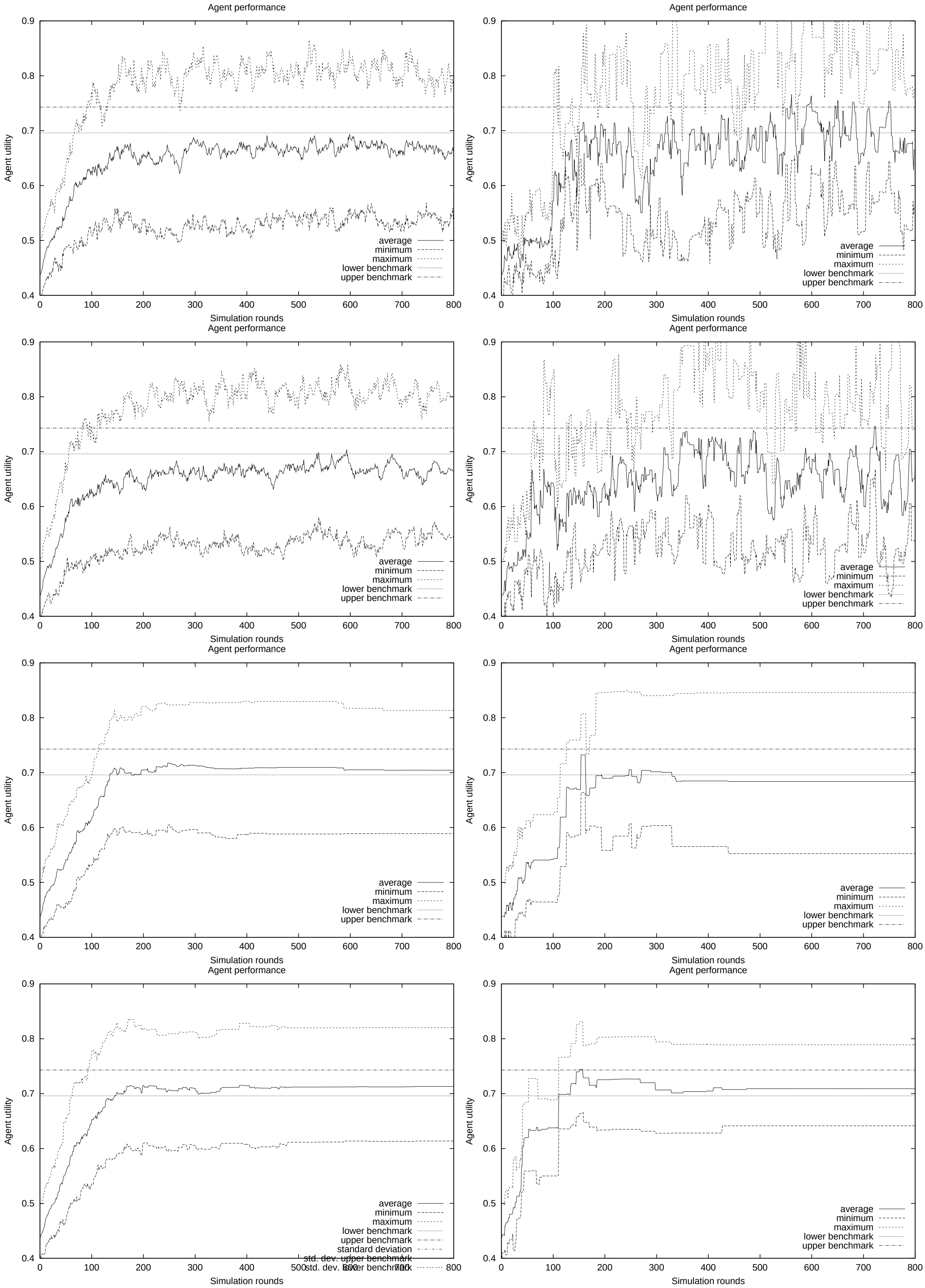


Fig. 6.14: Comparison between different desirability test strategies, no Q-learning. From top to bottom: NDT, ADT, FLT, and A&FLT agents

Of course, the learning capacities observed in the LIESON scenario are different from standard machine learning, in the sense that they are only thought to support the long-term optimisation of interaction strategies that are also influenced by the constantly changing environment and the changing needs of agents. Therefore, we cannot expect convergence to optimal behaviour. But our experiments show that what we *can* expect is *satisficing* learning behaviour and meaningful combination with other, non-InFFrA reasoning components in an integrated architecture. Obviously, the more reasoning is conducted regarding optimal action selection *under current circumstances*, the less important will considerations regarding the long-term usefulness of frames be. This can also be seen from the (smaller) effect learning has on overall system performance when desirability-ensuring methods are applied: In the case of FLT and A&FLT, the "added value" of learning is much smaller than in NDT and ADT (where it is essential to achieve any long-term improvement at all).

Finally, what should not be forgotten is that even if learning does not affect performance decisively in some cases, it does take a lot of weight off agents' "shoulders". This is because frame selection allows for ignoring a large portion of the search space of expectation structures by only considering one frame at a time. Considering this restriction, even identical performance would constitute success. We have shown that m^{2}InFFrA agents do better than this.

6.3 Advanced Experiments

While the results of the previous section are quite impressive and illustrate that the architecture works in practice, they also lack complexity with certain respects:

- The frames used so far contain only the most simple conceivable context information, as their conditions only require that the physical actions involved in a frame are executable. This does not really prove that our reinforcement learning techniques can be combined with context-sensitive conditioning of communication patterns.

- In the experiments above, agents can simply opt out of any conversation at any point in time. This represents only the most voluntaristic mode of social exchange, in which all that agents can learn is making suggestions that are profitable for the other. In contrast to this, many social contexts are characterised by strict rules of social conduct that require more complex reasoning in order to use the underlying norms to one's own advantage.

To transcend this simple level of negotiation, in which agents basically only exchange simple proposals regarding the actions that should be performed next, we will deal with *interest-based negotiation* frames in the second series of simulation experiments.

These advanced experiments involve frames in which agents discuss their goals, point at problems, etc. In the following sections, we first provide a brief introduction to interest-based negotiation, and then report on the experiments conducted within that framework. In that, more space will be devoted to describing the process of designing appropriate negotiation frames than to the experiments conducted with these, the purpose being to provide an extensive case study of the application of m^{2}InFFrA to a given communication scenario.

6.3.1 Interest-based negotiation

Interest-based negotiation (IBN) is a special form of *argumentation-based negotiation* (ABN) (Rahwan et al. 2004). In contrast to proposal-based negotiation methods (such as the contract-net protocol (Smith and Davis 1981), auctions, voting and bargaining (Sandholm 1999), and game-theoretic negotiation models (Raiffa 1982, Rosenschein and Zlotkin 1994)), in which proposals (for the execution of joint actions, the purchase of goods, etc.) are exchanged and an agreement is reached if the proposal is accepted by the negotiating parties, ABN is about exchanging *arguments* to convince each other (of the truthfulness of some fact or theory, of the usefulness of some action, etc.).

The crucial difference between these two types of negotiation is that while only the proposal itself is the issue of negotiation in proposal-based negotiation, ABN allows agents to exchange information beyond the proposal. These pieces of additional information exchanged, called *arguments*, can be used by the agent (i) to justify its negotiation stance or (ii) to influence the other agent's negotiation stance (Jennings et al. 1998a).

IBN is a specific ABN framework proposed by Rahwan et al. (2003) that focuses on exchanging arguments regarding the goal structures and preferences of the negotiating parties. Essentially, it is based on the idea that agents *challenge* each other's proposals or claims to obtain information about the underlying reasons for them (i.e. beliefs, goals, etc.). Using information that is derived from the *justifications* put forward by the other in reply to the challenge, they can then attempt to *attack* those reasons. This can be done by pointing at problems, misconceptions and inconsistencies, but also by suggesting alternative actions or goals. Either way, the objective of these attacks is to change the other's opinion, so that a mutually beneficial agreement can be reached.

What distinguishes IBN from other ABN approaches is that argumentation is not so much about "proving" the other wrong – rather, the purpose of argumentation is to understand the other's internal (mental) state and to alter it to one's own benefit.

Naturally, it is beyond the scope of this thesis to develop a comprehensive frame-based implementation of the theory of IBN, which is very complex in its entirety. Instead, we will present a a simplified view of IBN that is elaborate enough to be turned into interesting InFFrA frames. Apart from employing these "IBN frames" to conduct further simulation experiments, the development of these frames will also serve as a case study on how to build InFFrA agents for a given theory of interaction.

6.3.2 A simple model of IBN

Dialogue model

As mentioned, the basic idea in IBN is to challenge the other's statements so as to attack the justifications he provides for them and thus to persuade him into changing his mind. Figure 6.15 shows the main control loop for IBN dialogues. It consists of two parts:

- *Exchange of proposals:* This part of the dialogue model concerns the exchange of proposals, agreements and rejections. It may also encompass execution and monitoring of courses of action agreed upon.

  To keep things simple, we assume that all dialogues begin with a proposal. This is followed by potential subsequent proposal-rejection loops, in which agents iteratively

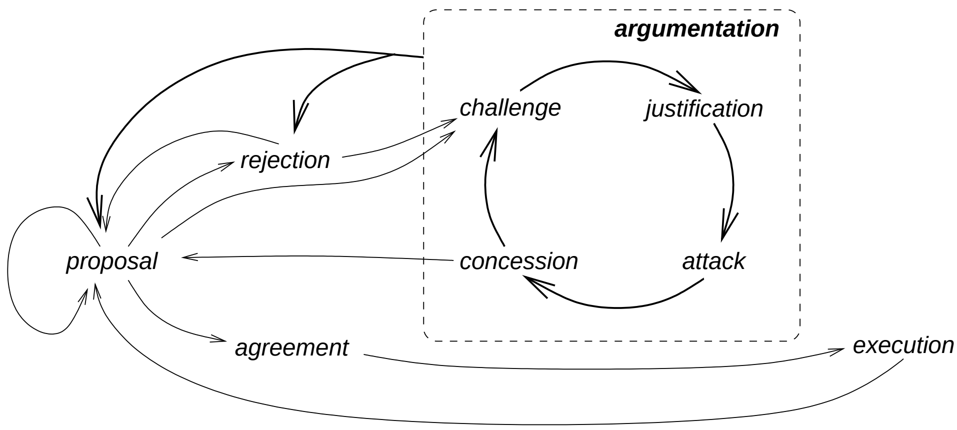


Fig. 6.15: Basic control flow of interest-based negotiation dialogues

make suggestions and counter-suggestions as long as one party does not accept the proposal. Eventually, an agreement is reached or the negotiation is simply terminated by one of the parties at any stage. If the topic of the conversation is not purely "theoretical" (i.e. a discussion about a fact or theory), but also involves an element of action, agreement is followed by execution of these actions. Since action execution may involve a complex flow of control for monitoring purposes (as, for instance, in the execution of joint plans), it may be necessary to interleave the process of execution with additional propose-agree loops to align actions properly, to re-plan in case of unexpected problems, etc. Also, action or plan execution may be followed by *new* proposals after its completion, but this is a case we shall ignore for simplicity (it can be simulated by starting a new dialogue).

- *Argumentation:* This is the part of the communication model in which agents interrupt the proposal exchange process to gather information about each other's mental state.

  If completed, the structure of the entire cycle of each argumentation loop is challenge-justification-attack-concession. After a proposal has been made or rejected, the other party may ask for some kind of justification by challenging the proposal or the rejection. After this justification is provided, it can be attacked, whereupon the attacked party will concede if the attack is successful.

  Of course, either negotiation party may find it appropriate to reject or make a new proposal at any point during this cycle, e.g. if an attack is unsuccessful, if no justification or attack can be found, if the information gathered from a justification is enough to make a new proposal without the necessity of attacking the justification, if the agent simply refuses to concede, etc. This is suggested by the arrows leading back to "propose" and "reject" from the dashed "argumentation" box in figure 6.15.

Essentially, the argumentation part of such negotiations is nothing else but an exchange of "mental proposals" (or, alternatively, "mental requests" – cf. section 4.1) in the form of arguments, with the difference that conceding to an argument is different from normal, proposal-related agreement as it refers to some previous(ly rejected) proposal but aims at transforming one's *stance* towards the original proposal.

Having sketched the process of IBN argumentation, we need to define what kind of information is to be gathered from challenges, how justifications can be generated and attacked, and in which situations attacks lead to concessions. In other words, we have to specify the arguments used in our model of IBN.

Goal graphs

Above, we remarked that IBN focuses on reasoning about each other's internal motives (beliefs, goals, etc.) in order to influence it. *Goal graphs*, as suggested in (Rahwan et al. 2003), enable this kind of reasoning. They do so by facilitating the representation of goal hierarchies, preferences and justifications for goals. As before, we will only discuss those aspects of the theory that are relevant to the development of IBN frames rather than introduce a full-fledged formalism.

Informally speaking, a goal graph depicts the relationships between facts, actions, and goals. It is based on the idea of representing goals and facts as nodes and connecting nodes v and v' by a directed link (v, v') if v *contributes* to the achievement of v'. Edges or sets of edges that end in the same goal are labelled with the identifier of an action to indicate that an action is necessary for a goal to be actually *achieved* if certain other goals are satisfied or certain facts hold. Thus, the ingoing edges of a goal node can be partitioned into subsets, and such a partition $E = E_1 \uplus \ldots \uplus E_k$ in a way "defines" the goal, as the sets of goals/facts that are the source nodes of each E_i have to be achieved *in conjunction* for the target of E to be achieved. E_i and E_j indicate *disjunctive* alternatives of achieving that goal (for $i \neq j$). Some of these "definitions" may require an action, others not. Goals which do not contribute to any other goals are called *supergoals*. For simplicity, we do not consider cycles in these graphs.

As a simple example, consider the goal of "getting to Rome". Obviously, depending on where one currently is located, getting to Rome may require using different means of transportation, making in-between stops, waiting for connections, etc. One definition of "getting to Rome" could be "being in Rome" which does not require any further conditions to be met or actions to be taken, because nothing needs to be done if someone is already in Rome. A second definition may be "getting to Milan and getting from Milan to Rome", where "getting to Milan" is a sub-goal which may, in turn, be defined in its own goal subgraph. If getting to Milan has already been achieved, then there might be further nodes labelled with the goals "no railroad workers' strike", "connection found" and "have at least 60 euro" which are grouped together and labelled with the action identifier "take train" to the node labelled "getting from Milan to Rome".

Note that if the definitions of certain goals involve actions, the goal graph description of goals to be achieved is very similar to conditional planning (Russell and Norvig 2003).

For the linkage scenario, we can build such a goal graph using information about the utility function, where the (single) supergoal of each agent is to increase one's own score. Looking back at equation 6.1 (p. 150), there are different ways to improve one's score:

- by increasing one's popularity directly through the links laid to one's site,
- by increasing the popularity of a positively rated peer ("friend"),
- by decreasing the popularity of a negatively rated peer ("enemy"),
- by decreasing the rating distance between a friend and oneself with respect to third parties, and
- by increasing the rating distance between an enemy and oneself with respect to third parties.

Obviously, depending on the beliefs of the agent (his private rating preferences, existing links, etc.), there may be many different ways of achieving each of these sub-goals. Figure 6.16 shows a possible goal graph for LIESON. Essentially, this goal graph describes how the different sub-goals listed above (called *+ownPopularity*, *-friendPopularity*, *-enemyPopularity*, *-ratingDiffFriend*, *+ratingDiffEnemy* in the graph) that lead to a score increase can be attained. For example, the rating difference in relation to a friend (A) can be reduced if a link from A to B is deleted and $|rating(D, B) - X| > 0 / |rating(D, A) - X| > 0$ holds where

- D is the reasoning agent
- $rating(D, B)$ and $rating(D, A)$ are the private ratings of D for the respective sites,
- X is the public rating of an existing link from A to B

In other words, the LIESON goal graph represents a simplified view of the knowledge we have of the score computation function, which is, of course, much more coarse-grained than the actual score computation function of equation 6.1.

Negotiation moves

Goal graphs are a means of modelling the other's internal state so as to understand his motives and goals and to "massage" him into accepting one's proposals or claims. Using them as an underlying model of reasoning about each other, we can derive the kinds of moves that are possible during an IBN dialogue.

In the "challenge" phase of a negotiation dialogue, the reasoning agent tries to find out why (say) his adversary rejected a proposal. Having only partial knowledge of the other's goals, he may ask for the *reasons* for the other agent's rejection to better understand the other's behaviour (and to find a suitable counter-argument). Such reasons may be that

- the other agent thinks that the requested proposal cannot be carried out, or that
- the proposal threatens the achievement of one of the other agent's goals.

By providing such a *justification*, the peer enables the challenging agent to refine his view of the other's goal graph. If a proposal cannot be implemented, then one of the preconditions (sub-goals) of the proposal must be false according to the peer's belief. If a proposal is a threat to some goal, then it obviously does not contribute to its achievement.

If it is not a rejection that is challenged but the proposal itself, the justification consists of naming the goal the proposal contributes to. The type of justification used, its content

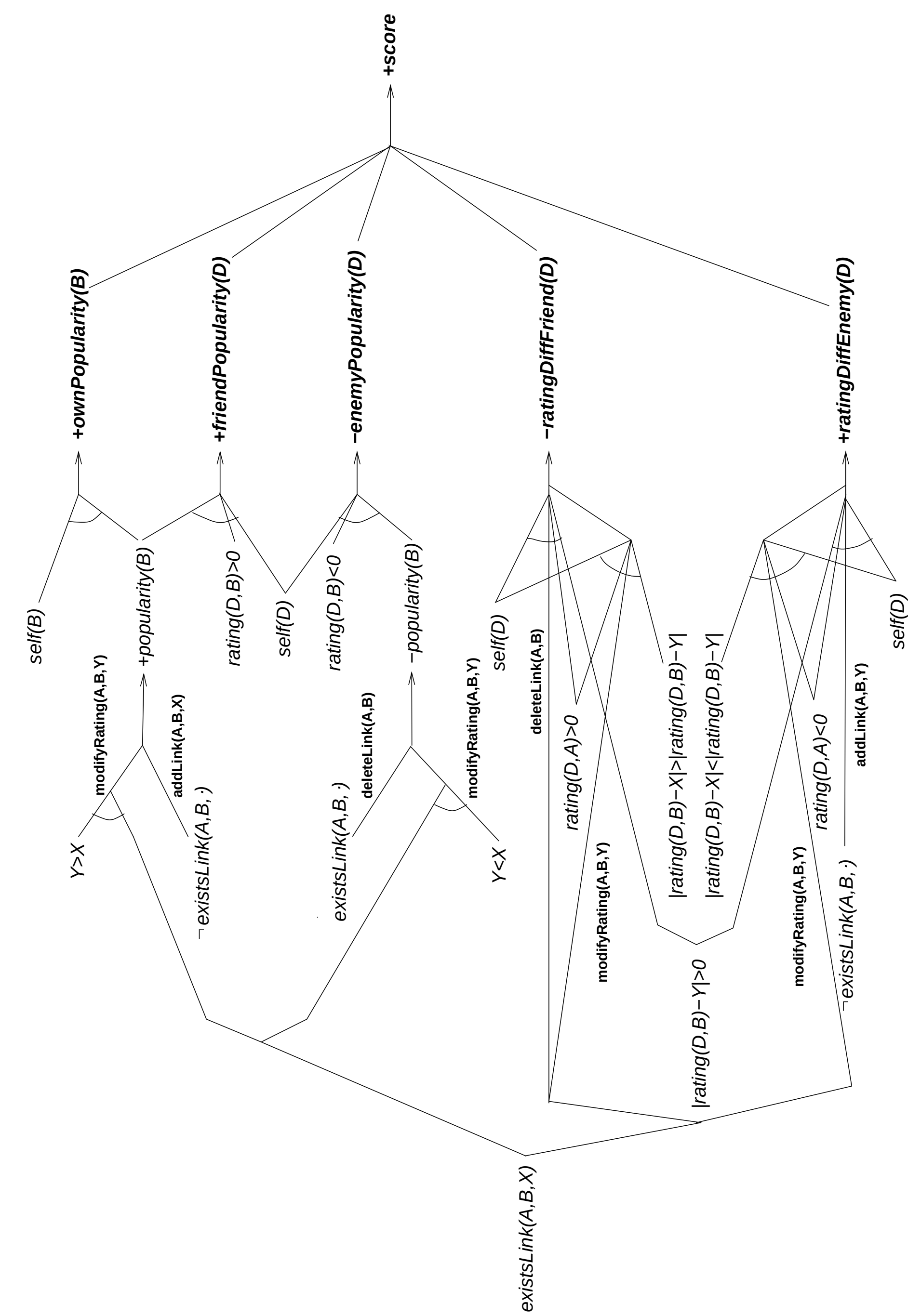


Fig. 6.16: LIESON goal graph: Each goal (in **bold face**) is defined in terms of sets of subgoals or world facts that are connected to the goal they achieve through edges connected with an arc (sub-goal conjunction). If the in-edges of a goal are not connected with each other, they denote alternative ways of achieving the same goal (sub-goal disjunction). If an action is necessary for some (sets of) facts or subgoals to achieve a goal, then the respective edge(s) are labelled with that action. Capital letters are used for variables with "·" as a wildcard symbol for argument values.

and the previous knowledge the challenging peer had of the other's goal graph may determine the kind of *attack* he will use:

- If the other justified his rejection with a potential threat for one of his goals, this can be attacked by proposing an alternative goal.
- If the justification of a rejection lay in claiming the impossibility of implementing the proposal, the other agent may point out to his peer that his knowledge is either obsolete or erroneous.
- If a proposal was justified by a goal, there are several possible attacks:
  - We can argue that the proposal threatens some other goal at the same time.
  - We can argue that the goal put forward cannot be achieved anyway.
  - We can suggest adoption of an alternative goal.

Although there certainly exist more possible kinds of attacks (e.g. appealing to previous commitments, appealing to higher authorities or norms, arguing about the risk involved in trying to achieve a goal, etc.) the above are the most straightforward and generic that can be conceived of.

The type of attack determines whether the other will *concede* or challenge the attack, thus perpetuating the process of argument exchange. Certainly, this description leaves out the details of

- how to evaluate proposals and arguments,
- how to specify which arguments defeat which other arguments,
- how to determine the best possible attack, and
- how to decide on making concessions

which are very important aspects of concrete IBN systems. However, we will show that a generic social reasoning architecture like InFFrA actually enables us to proceed with the implementation of a simple variant of IBN without worrying too much about these issues. The following section shows how this can be done by virtue of "shifting" the semantics of argumentation from the mental (agent) to the social (communication) level.

6.3.3 IBN frames

In constructing frames for IBN among InFFrA-based agents, we adopt a different position from the heavily mentalist approach of "generating arguments to model the other's goal graph and identify optimal proposals/attacks/justifications". Quite differently, we will assume that the *general* structure of goal graphs (up to instances of fact nodes that depend on each agent's private knowledge base, that is) is common knowledge in the sense of a shared communicative convention. In other words, it does not matter whether agents actually *have* the goal graph shown in figure 6.16, but the frames that we provide will force agents to behave "as if" they were acting under the premise that their actions must be in keeping with this goal structure. So when agents communicate, they will be forced to create justifications and attacks in accordance with it.

This means that we regard IBN not so much as an improved method of coordination for the LIESON system, but rather as a complex communication regime that governs the social context within which agents interact. In particular, our agent design so far is quite contradictory to the goal graph with respect to goal generation and prioritisation. Recalling that agents apply the score-predicting function of equation 6.1 (in a boundedly rational fashion using the simple popularity computation method of equation 6.6, p. 152) to project possible link manipulation actions they or their peers might perform and prioritise these in their BDI queue, it is clearly the case that the goal graph plays only a very indirect role in goal generation. More specifically, agents do not consider the importance of all alternative ways of achieving *+score* equally important, as actions that yield a numerically higher potential score are preferred in a greedy fashion. Also, they disregard the fact that some actions previously enqueued might actually not increase their score under the current global link network because they threaten an intermediate goal in the goal graph.

The basic assumptions underlying our definition of IBN frames are:

- All agents share the same goal graph. When talking about relationships between different goals or facts and goals, each agent may argue about the other's goals by referring to the goal graph. Thereby, if certain relationships depend on the satisfiability of certain facts (e.g. existence of certain links), agents reason about goals under the assumption that their own current local knowledge base is correct.

- As far as argumentation is concerned, the existing links are assumed to express the agent's true opinion of some other site, i.e. when reasoning and arguing about someone's goals, agents may assume that $existsLink(a_i, a_j, R) \Rightarrow rating(a_i, a_j) = R$, where $rating(a_i, a_j) = R :\Leftrightarrow r_i(a_j) = R$. So it is assumed that $r_i(a_j) = r(a_i, a_j)$ in the notation of section 6.1.2. This implies that agents are honestly revealing their preferences by the links they lay.[16]

- The communicative conventions require that an agent is able to appeal to a goal he is pursuing if he issues a request, and that he is able to name a threat if he rejects an action requested from him. This means that no requests or refusals can be uttered unless there is a justification for them according to the goal graph. It also implies that attacks directed at a justification must attack a *goal* if the attack is generated by the *responder* in the conversation (and the goal was referred to by the initiator to justify his request), and they must attack a *threat* if the attacking party is the *initiator* of the conversation (who is attacking a justification put forward by the responder for his rejection). The only exception to this rule is when the responder points to a problem with the executability of the action requested from him, i.e. when his beliefs make the action appear impossible.

- An attack may either appeal (i) to an alternative goal, (ii) to an alternative means of achieving the same goal, (iii) to a threat for some goal that is induced by the requested action, or (iv) simply point out that the problem put forward by the other agent does

[16] Keep in mind, though, that agents do not consider others' preferences when they request link modification actions but argue using goal graph structure. For example, if changing a rating from -2 to 3 appears to serve the other's goal, the agent who requests this action does not care whether this would force the other agent to "lie" if his true rating of that site was actually 2. This puts additional pressure on each agent, as he may be forced to change his ratings due to an encounter with one peer, thereby changing his argumentation position for future conversations.

Predicate	*Description*
$possproblem(P, X)$	if P is the case, action X cannot be executed
$problem(P, X)$	X cannot be executed due to fact P (which currently holds)
$goal(A, G)$	G is a goal of agent A
$contrib(G, G')$	goal G contributes to the achievement of goal G'
$achieves_0(X, G)$	action X achieves goal G under current circumstances
$achieves(X, G)$	$achieves_0(X, G) \vee (\exists G'.achieves_0(X, G') \wedge contrib(G', G))$

Tab. 6.9: Auxiliary predicates for IBN frames: *possproblem* and *problem* refer to theoretical/actual problems with executing an action; the *contrib* predicate is equivalent to an edge in the goal graph not labelled with an action, $achieves_0$ stands for an edge that requires an action (edges from facts to goals are implicit since we are only concerned with achievement under current circumstances); *achieves* is true for all super-goals of goals directly achieved

not exist. If the attacking agent finds such a point of attack, the other agent has to concede, and, if he was the one from whom an action was requested, he must accept the original proposal. The same holds true if the justification used by the attacked agent or the problem put forward by him are wrong (or he can't identify a suitable one).

- Finally, only the goals and actions of one agent are discussed during a single conversation. This is always the agent who was challenged, i.e. either the initiator who requested an action or the responder that rejected a proposal.

It should be noted that these requirements are deliberately chosen for our construction of IBN frames, and that different assumptions might be made in other implementations. For example, agents could be allowed to threaten others with certain punishing actions if they fail to comply with what they desire, appeal to habitual practice in a given social context, etc.

The IBN frames we actually initialise agents' repositories with are shown in tables 6.10 (normal execution of request and dealing with execution problems), 6.11 (attacks to asserted threats), and 6.12 (attacks to asserted goals). Table 6.9 (p. 198) describes those auxiliary predicates used for the specification of these frames that were not defined in table 6.5 (p. 167).

In table 6.10, we include a frame F_N of normal execution of a request. Note that in the IBN experiments this is the only non-argumentative frame, i.e. there is no possibility of making counter-proposals, so unless an agent wants to reject immediately (and deviate strongly from everyone's expectations, he *has* to argue to avoid fulfilling and undesirable request. The next frame, F_{AP} (Attack-Problem) depicts the successful attack of a problem: The requestee suggests that P is a problem that hinders execution of X (e.g. $P = \neg\exists R.existsLink(B, C, R)$ and $X = deleteLink(B, C)$) but has to concede and execute X if $\neg P$ is the case and is asserted by the requesting party. The condition sets of this frame are particularly interesting, as they may involve a "change of mind". If the reasoning agent is the one who is requesting the action ($self(A)$), he can use this frame if $possproblem(P, X)$, i.e. if P is a potential problem for X (but need not hold currently). If the reasoning agent is the requestee ($self(B)$), P additionally needs to hold before it is claimed

$$
\begin{aligned}
F_N = \ & \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{do}(B,X) \rangle, \\
& \langle \{ \mathit{can}(B,X)@1, \mathit{effects}(X)@2 \} \rangle, \\
& \langle \xrightarrow{0} \langle\rangle \rangle \Big\rangle \\
F_{AP} = \ & \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{reject}(B,A,X) \xrightarrow{0} \texttt{ask-reason}(A,B,\texttt{reject}(X)) \\
& \quad \xrightarrow{0} \texttt{inform-problem}(B,A,P) \xrightarrow{0} \texttt{attack-problem}(A,B,\neg P) \\
& \quad \xrightarrow{0} \texttt{concede}(B,A,\neg P) \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{do}(B,X) \rangle, \\
& \langle \{ \mathit{self}(A), \mathit{possproblem}(P,X), \mathit{check}(P)@4, \mathit{can}(B,X)@7, \mathit{effects}(X)@8 \}, \\
& \quad \{ \mathit{self}(B), \mathit{problem}(P,X)@1, \mathit{check}(P)@5, \mathit{can}(B,X)@7, \mathit{effects}(X)@8 \} \rangle, \\
& \langle \xrightarrow{0} \langle\rangle, \xrightarrow{0} \langle\rangle \rangle \Big\rangle \\
F_{CP} = \ & \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{reject}(B,A,X) \xrightarrow{0} \texttt{ask-reason}(A,B,\texttt{reject}(X)) \\
& \quad \xrightarrow{0} \texttt{inform-problem}(B,A,P) \xrightarrow{0} \texttt{concede}(A,B,P) \rangle, \\
& \langle \{ \mathit{self}(A), \mathit{problem}(P,X)@4 \}, \{ \mathit{self}(B), \mathit{problem}(P,X) \} \rangle, \\
& \langle \xrightarrow{0} \langle\rangle, \xrightarrow{0} \langle\rangle \rangle \Big\rangle \\
F^R_{CC} = \ & \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{reject}(B,A,X) \xrightarrow{0} \texttt{ask-reason}(A,B,\texttt{reject}(X)) \\
& \quad \xrightarrow{0} \texttt{concede}(B,A,\mathit{no\text{-}reason}()) \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{do}(B,X) \rangle, \\
& \langle \{ \mathit{can}(B,X)@5, \mathit{effects}(X)@6 \} \rangle, \\
& \langle \xrightarrow{0} \langle\rangle \rangle \Big\rangle \\
F^I_{CC} = \ & \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{ask-reason}(B,A,X) \\
& \quad \xrightarrow{0} \texttt{concede}(A,B,\mathit{no\text{-}reason}()) \rangle, \\
& \langle \{\} \rangle, \\
& \langle \xrightarrow{0} \langle\rangle \rangle \Big\rangle
\end{aligned}
$$

Tab. 6.10: IBN frames for the LIESON scenario: Normal execution of a request, pointing out a problem that hinders execution of the requested action, conceding to an asserted problem and admitting a lack of justification to be a problem for X for the frame to be used (@1). The $check(P)$ predicates force agents to actually interrupt the encounter and check (by observation of the environment) whether P is the case, and concession is mandatory if B actually can perform the action in the end.

The semantics of $check(P)@i$ are that it is true for both P and $\neg P$ if time-step i has not been reached yet, i.e. the outcome of proving P is not determined yet. Thus, before time-step 4, it doesn't matter whether P is the case or not for A – he can plan to attack by asserting $\neg P$ anyway (also, as P is not yet bound to a value, it would make no sense to prove it at this point in time anyway). When time-step i is reached, an update-action has to be spawned to check whether P is the case. After this, only P *or* $\neg P$ can be true. In F_{AP} this means that once `inform-problem`(B, A, P) is uttered, A steps back from the conversation to check whether P holds. If he discovers P to be true, the *can*-predicate will become false, and he cannot attack P. Conversely, B must check whether P holds after receiving `attack-problem`$(A, B, \neg P)$ to make sure he was right about P.

Thus, the difference for the two parties is that B must concede if he thought it is a problem (otherwise he must not even use it as a justification), whereas A will accept the other's justification (temporarily) even if P is only a potential (but non-existent) problem (this is necessary because A might be initially unable to prove $problem(P, X)$ if he thinks there is no problem regarding X). This nicely illustrates how the condition-proving mechanism must be aligned with frame design to result in appropriate framing. If P is a real problem A must concede if he can prove $problem(P, X)$, and this is captured by the third frame F_{CP} (Concede-Problem). Frames F^R_{CC} (Concede-Challenge-Responder) and F^I_{CC} (Concede-Challenge-Initiator), finally, can be used to concede that one has no suitable reason to provide for his rejection or proposal. F^R_{CC} amounts to having to execute the original proposal, while F^I_{CC} leads to encounter termination without any action consequences.

The second category of IBN frames (table 6.11) deals with attacks to claimed threats, i.e. to situations in which agent B refuses to execute X and justifies this with a threat to a goal T that X does not achieve. Note that the interpretation of "threat" used here is a very broad one, as it considers every action a threat for a goal that does not achieve the goal immediately. A much stricter interpretation of the concept would be to only call X a threat for T if, for example, after execution of X there is no way of achieving T. F_{ATG} (Attack-Threat-Goal) describes an attack to a threat by appealing to an alternative goal which X achieves instead and F_{ATM} (Attack-Threat-Means) specifies how a threat can be attacked by pointing at an alternative action for achieving T (assuming that T is important to B since he mentioned it). F_{CT} (Concession-Threat), finally, has to be used if the attacking party has to concede to the asserted threat, and no physical action is taken. The frames contained in table 6.12 are quite orthogonal to those of the previous category, as they cater for attacks to goals asserted by the requesting party A. In F_{AGM} (Attack-Goal-Means), B finds an alternative action Y that would achieve the goal G put forward by A, and AGT (Attack-Goal-Threat) points at X being a threat to some other pursued goal. Finally, F_{CG} (Concession-Goal) marks the situation in which B has to give in because no threat or alternative action could be identified as a suitable attack. As opposed to the the second category, concession entails that the requestee must fulfil the original request in these frames, while attack using a threat implies that A must give up his original demand, which is in accordance with the fact that the attacking and justifying parties have been swapped.

To summarise, we have managed to develop m²InFFrA frames for the IBN framework in such a way that agents can argue about each other's goals and beliefs. These frames require

$$F_{ATG} = \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{reject}(B,A,X) \xrightarrow{0} \texttt{ask-reason}(A,B,\texttt{reject}(X)) \xrightarrow{0} \texttt{inform-threat}(B,A,T) \xrightarrow{0} \texttt{attack-threat}(A,B,\textit{alternative-goal}(G)) \xrightarrow{0} \texttt{concede}(B,A,G) \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{do}(B,X)\rangle, \langle\{\textit{can}(B,X),\textit{goal}(B,T),\neg\textit{achieves}(X,T),\textit{goal}(B,G),\textit{achieves}(X,G),G \neq T\ \textit{can}(B,X)@7,\textit{effects}(X)@8\}\rangle, \langle \xrightarrow{0} \langle\rangle\rangle \Big\rangle$$

$$F_{ATM} = \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{reject}(B,A,X) \xrightarrow{0} \texttt{ask-reason}(A,B,\texttt{reject}(X)) \xrightarrow{0} \texttt{inform-threat}(B,A,T) \xrightarrow{0} \texttt{attack-threat}(A,B,\textit{alternative-action}(Y)) \xrightarrow{0} \texttt{concede}(B,A,Y) \xrightarrow{0} \texttt{request}(A,B,Y) \xrightarrow{0} \texttt{do}(B,Y)\rangle, \langle\{\textit{can}(B,X),\textit{goal}(B,T),\neg\textit{achieves}(X,T),\textit{achieves}(Y,T),Y \neq X\ \textit{can}(B,Y)@7,\textit{effects}(Y)@8\}\rangle, \langle \xrightarrow{0} \langle\rangle\rangle \Big\rangle$$

$$F_{CT} = \Big\langle \langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{reject}(B,A,X) \xrightarrow{0} \texttt{ask-reason}(A,B,\texttt{reject}(X)) \xrightarrow{0} \texttt{inform-threat}(B,A,T) \xrightarrow{0} \texttt{concede}(A,B,\texttt{reject}(X))\rangle, \langle\{\textit{can}(B,X),\textit{goal}(B,T),\neg\textit{achieves}(X,T)\}\rangle, \langle \xrightarrow{0} \langle\rangle\rangle \Big\rangle$$

Tab. 6.11: IBN frames for the LIESON scenario: Rejection is only justified if a threat can be appealed to; the threat can be attacked by suggesting an alternative goal or an action that does not threaten the goal. If no attack can be found, the argumentation initiator has to concede.

agents to be able to give reasons for suggesting or rejecting an action. Furthermore, agents must accept any alternative action that leads to achievement of the same goal and any alternative goal that can be achieved instead if these are used in attacks by the respective other party. There is no prioritisation among goals, and arguments only refer to the goal graph of that agent whose stance is being challenged.

6.3.4 Results

We have conducted extensive simulation experiments with m²InFFrA agents who employ IBN frames using the same configuration of the agent society and identical parameter settings as in section 6.2.2. While the purpose of this was to ensure maximal comparability between the results of the two series of experiments, we choose to make the IBN scenario even more difficult for agents by forcing them to use the *naive* desirability test strategy NDT (cf. the discussion in section 6.2.2, p. 179). This means that we prohibit agents to simply opt out of conversations or to cancel agreements just because they do not seem profitable to them.

Why is this reasonable? IBN frames provide methods for dealing with rejection explicitly, as the different frames represent *methods of conflict resolution* which can be used

$$\begin{aligned}
F_{AGM} = \; & \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{ask-reason}(B,A,\texttt{request}(X)) \xrightarrow{0} \\
& \quad \texttt{inform-goal}(A,B,G) \xrightarrow{0} \texttt{attack-goal}(B,A,\textit{alternative-action}(Y)) \\
& \quad \xrightarrow{0} \texttt{concede}(A,B,Y) \xrightarrow{0} \texttt{do}(B,Y) \big\rangle, \\
& \quad \big\langle \{ \textit{can}(B,X), \textit{goal}(A,G), \textit{achieves}(X,G), \textit{achieves}(Y,G), \\
& \quad X \neq Y, \textit{can}(B,Y)@5, \textit{effects}(Y)@6 \} \big\rangle, \big\langle \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle \\
F_{AGT} = \; & \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{ask-reason}(B,A,\texttt{request}(X)) \\
& \quad \xrightarrow{0} \texttt{inform-goal}(A,B,G) \xrightarrow{0} \texttt{attack-goal}(B,A,\textit{threat}(X,T)) \\
& \quad \xrightarrow{0} \texttt{concede}(B,A,\textit{threat}(X,T)) \big\rangle, \\
& \quad \big\langle \{ \textit{can}(B,X), \textit{goal}(A,G), \textit{achieves}(X,G), \textit{goal}(A,T), \neg \textit{achieves}(Y,T) \} \big\rangle, \\
& \quad \big\langle \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle \\
F_{CG} = \; & \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{ask-reason}(B,A,\texttt{request}(X)) \\
& \quad \xrightarrow{0} \texttt{inform-goal}(A,B,G) \xrightarrow{0} \texttt{concede}(B,A,\textit{goal}(G)) \\
& \quad \xrightarrow{0} \texttt{request}(A,B,X) \xrightarrow{0} \texttt{do}(B,X) \big\rangle, \\
& \quad \big\langle \{ \textit{goal}(A,G), \textit{achieves}(X,G), \textit{can}(B,X)@5, \textit{effects}(X)@6 \} \big\rangle, \\
& \quad \big\langle \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle
\end{aligned}$$

Tab. 6.12: IBN frames for the LIESON scenario: A request must be justified by a goal; this justification can be attacked by suggesting an alternative that achieves the same goal or pointing out a threat. If no attack can be found, the argumentation initiator (responder in the conversation) has to concede and the initial request is honoured.

whenever a goal or belief conflict occurs among agents (see section 4.1.6). By defining appropriate reactions to rejection, IBN frames allow agents to *distinguish* whether a rejection is due to framing failure (in which agents still utter a default reject message and terminate the encounter) or whether it is deliberate. In the latter case, the rejecting party has to justify its position by referring to goals and beliefs while adhering to the communicative "rules of the game". As our intention is to analyse how well agents can cope with these rules, it does not make sense to allow them to break frames, unless there is real framing failure. Note that by using Q-based frame selection and substitution optimisation, agents will still attempt to find the best possible solution that is in accordance with the communicative regime, so the naive desirability strategy does not imply that agents will accept any proposal or argument.

Practically speaking, this amounts to a different meaning of "reject" than before: If this performative occurs as the second message in a conversation right after a request, it is a "normal" utterance that may be followed by challenge, attack, etc. Else, it marks a real misunderstanding. Since agents are only allowed to depart from the defined frames for validity

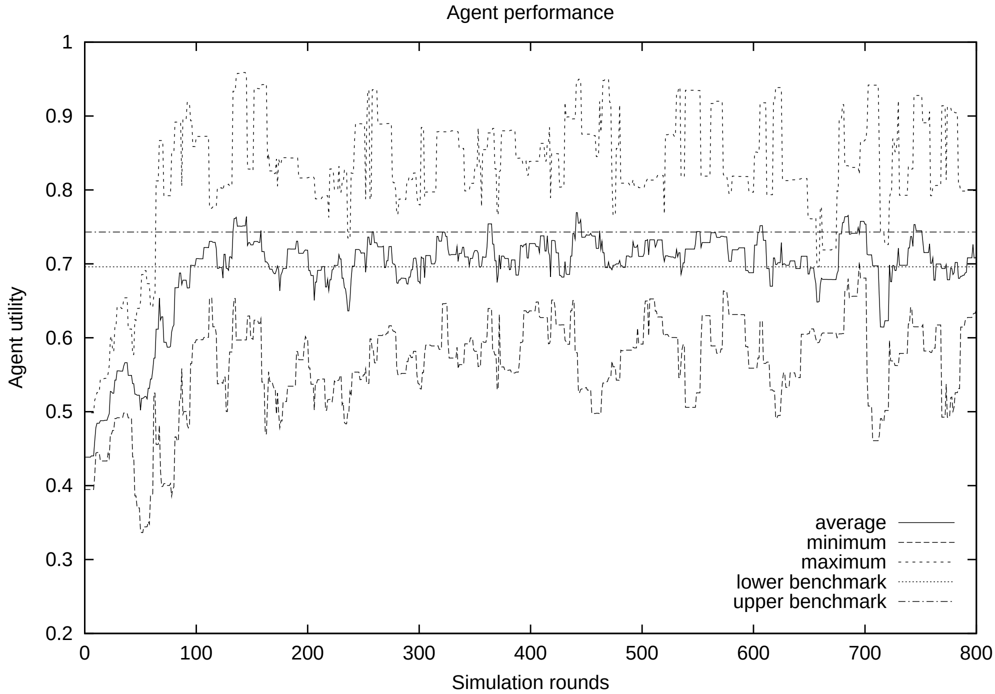


Fig. 6.17: Single-run average performance of IBN agents

or adequacy (and not for desirability) reasons in this scenario, we can infer that the reason for any "broken" frame must be that one of the agents is unable to match the perceived frame with one of the IBN frames or to prove the conditions of any matching frame. This can occur, for example, if agent A suggests `attack-threat`$(A, B, \mathit{alternative\text{-}goal}(G))$ in F_{ATG} (p. 201) when he believes that X has threatened some other goal T of B but B cannot prove $\mathit{achieves}(X, G)$ with his knowledge base. In this case he would have to reject and quit the encounter.

Performance results

With these preliminary considerations in mind, let us now turn to the performance analysis of "IBN agents". Figure 6.17 shows a typical single-run average for these agents. Although the IBN scenario requires much more intricate reasoning and agents are only able to achieve the desired actions if they either manage to win an argument or are lucky enough not to be challenged by their interaction partner, agent performance is impressive, as it reaches almost the same level as with simple proposal-based agents. In fact, compared to the NDT case in our basic experiments (cf. 6.10), agents do significantly better. The reason for this is that by reasoning over the other's goals, the alternatives that agents suggest are desirable for the other quite often (even if the other does not have the right to reject for desirability reasons, he might consider a challenge or attack rather than accept a (counter-)proposal). In other words, argumentation-based negotiation creates a tendency towards *mutually beneficial counter-proposals*, because the counter-proposals have to take the goals of the agent whose initial proposal (or rejection) is challenged into account.

Furthermore, interest-based negotiation seems to have an *equilibratory* effect. As can be seen from the standard deviation curve in figure 6.18, the variance between the per-

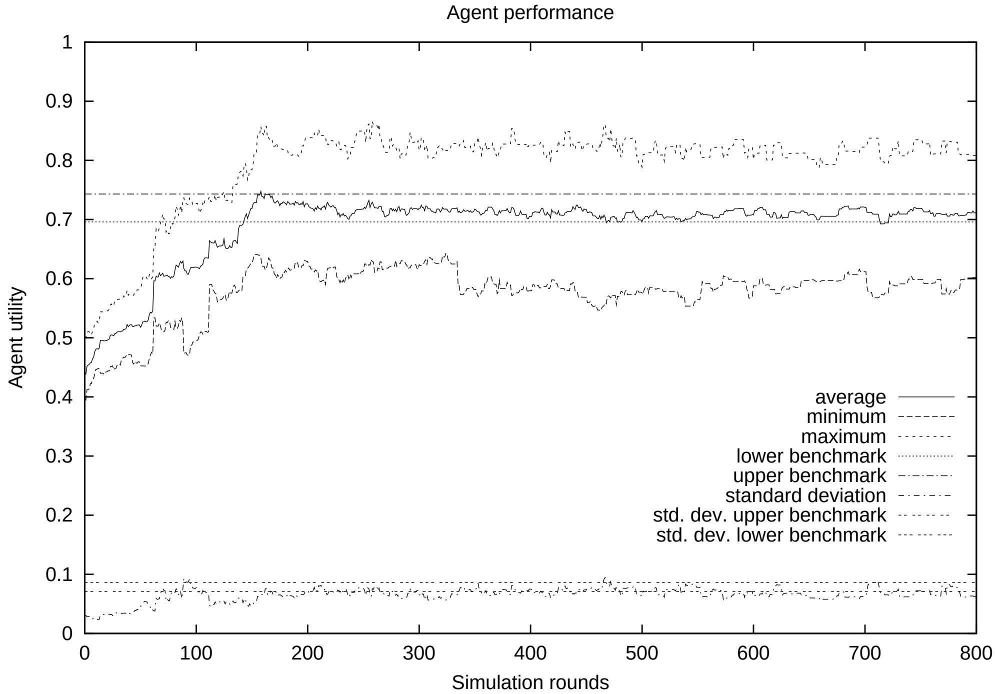


Fig. 6.18: Multi-run average performance of IBN agents

formance of different agents is similarly low as in the non-communicating BDI case. This seemingly contradicts the observation that the distance between best and worst agents is equally high in the IBN single-run example of 6.17 and in the single-run plots for simple proposal-based agents (plots on the right hand side columns of figures 6.10 and 6.14). A closer analysis of each individual agent's performance in all 50 runs (figure 6.19) sheds some light on this matter. As depicted there, IBN enables *every* agent to win an argument now and then regardless of his "power" in terms of popularity. This is true even of the "weakest" agents a_0, a_5 and a_6 whose score suddenly increases way above its average level every now and then. Of course, all agents often also lose arguments so that they are forced to execute undesirable actions. Although we do not intend to draw any philosophical conclusions from this observation, it does suggest that adhering to a strict regime of rational communication rules may help reduce the chasm between most and least powerful agents.

Example conversation

To give a feel for the kind of conversations that can be observed in such IBN simulations, tracing an example negotiation is useful. In this example, a initiates a conversation by asking b to add a link to a's site with rating 3, i.e. the first message is

$$\texttt{request}(a, b, \texttt{do}(b, \mathit{addLink}(b, a, 3))).$$

Note that, as in all above m²InFFrA simulations, this request is spawned because $\mathit{addLink}(b, a, 3)$ appears at the top of a's BDI queue, i.e. no framing activity has been started yet by a. Upon receipt of this message, b picks F_{CT}[17] (p. 201) because this frame has a higher

[17] In practice, F_{CT} has changed over time (like all other repository frames) and differs significantly from the original repository frame in terms of previous stored cases and/or frame merging. We refer to the original

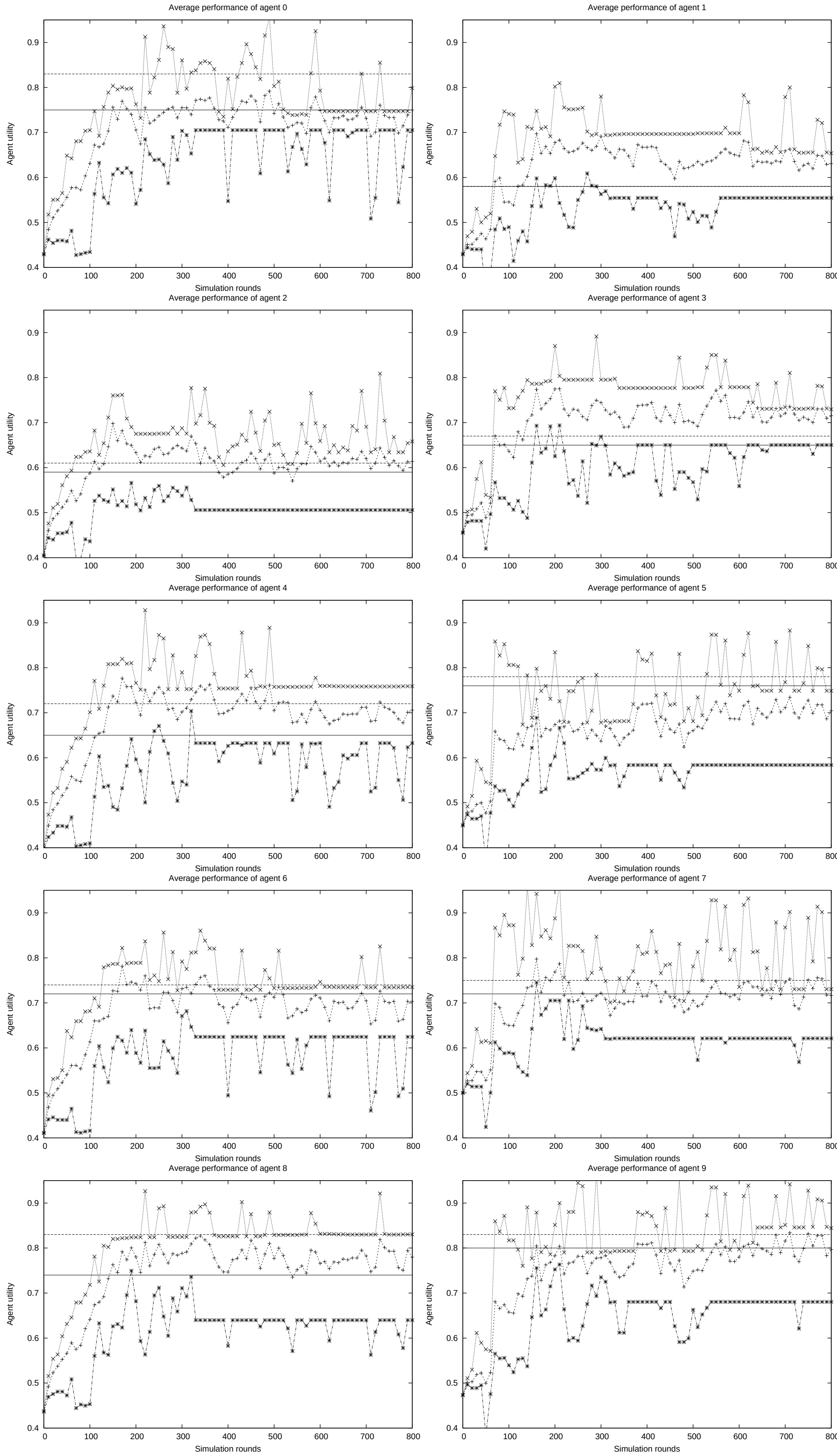


Fig. 6.19: Multi-run average performance of each individual agent: Each plot shows average, maximal, and minimal performance of a particular agent for the case of IBN agents, with curves and benchmarks as before

Q-value than F_N in the current state $\{\text{false: } \uparrow (R, I, -)\}$ (assuming that b does not like a, see also the explanation of LIESON encounter states in section 6.2.1, p. 168). This is reasonable, because a has experienced a decrease in utility when employing F_N and laying a link or strengthening an existing link to an "enemy". Hoping that a will concede after hearing that $addLink(b, a, 3)$ is a threat for b's goal $+score$, b replies with

$$\texttt{reject}(b, a, \texttt{do}(b, addLink(b, a, 3)))$$

and it is important to remember that activation of F_{CT} requires b to prove $goal(b, +score(b)) \wedge \neg achieves(addLink(b, a, 3), +score(b))$ ***prior to*** rejecting, which illustrates that in our argumentation scenario rejection is only permitted if one has a good reason for it. Unfortunately, b's reply does not allow a to activate F_N which would have been the best choice in state $\{\text{true: } \uparrow (R, I, -)\}$, but he can resort to F_{CC}^{R} (p. 199) and hope that b will not find a suitable justification for his rejection. So, the next message is

$$\texttt{ask-reason}(a, b, \texttt{reject}(\texttt{do}(b, addLink(b, a, 3))))$$

but – sadly for b – a has an argument up his sleeve, and justifies his stance by sending an

$$\texttt{inform-threat}(b, a, +score(b))$$

message according to F_{CT}. This forces a to re-frame once more, and although attacking with the proposal of some alternative goal or action (F_{ATG}/F_{ATM}, p. 199) would yield the highest utility according to the Q-table, a finds himself unable to prove $achieves(\texttt{do}(b, addLink(b, a, 3), G)$ and $achieves(Y, +score(b))$ so that both attack frames fail for lack of condition satisfiability. Therefore, a has to activate F_{CT}, and the encounter ends after giving in with

$$\texttt{concede}(a, b, \texttt{reject}(\texttt{do}(b, addLink(b, a, 3))))$$

whereupon the original request is never fulfilled.

This shows how a sensible flow of communication can be ensured by specifying content constraints (cf. section 4.1.5) on the applicability of different frames under different circumstances. Of course, these constraints cannot avoid misunderstandings that result in "broken" frames if participants' knowledge base contents are incompatible. In the above conversation, for instance, if a had (additionally) not been able to prove $\neg achieves(\texttt{do}(b, addLink(b, a, 3)), +score(b))$ after he received the `inform-problem` message, he would not have understood the argument put forward by b. In the absence of any matching, executable frame the only thing he could do in this case would be to send a `reject` message and cancel the dialogue.

Outlook: Frame construction

The negotiation frames presented above utilise a wide range of different justifications and attacks, but they do not allow for exchanging a series of subsequent arguments in a single conversation. To sketch the process of modularising the frames we have used so far and combining them appropriately to enable *iterated* arguments, we shall give an outlook on how this can be achieved.

repository frames here only for ease of presentation.

Ideally, we would like to have one challenge-justification-attack-concession cycle frame for each particular justification-attack type listed in tables 6.10, 6.11 and 6.12. These frames could then be added to perceived encounter prefixes in such a way that they enable agents to continue a discussion to exchange different arguments (and hopefully reach an agreement). Obviously, introducing one such frame for each of the five justification-attack combinations is a tedious business and also suggests that we are overlooking possibilities for generalisation.

In fact, the eleven frames we have used so far can be reduced to a much more concise set of six frames by introducing predicates and rules that capture all the relationships between justifications and possible attacks at knowledge-base (rather than frame condition) level:

$$\begin{aligned}
goal(A,G) \wedge achieves(X,G) &\Rightarrow justification(means(X,G), \mathtt{do}(A,X)) \\
goal(A,G) \wedge \neg achieves(X,G) &\Rightarrow justification(threat(X,G), \neg\mathtt{do}(A,X)) \\
problem(P,X) &\Rightarrow justification(problem(P,X), \neg\mathtt{do}(A,X)) \\
\neg P &\Rightarrow attack(\neg P, problem(P,X)) \\
goal(A,T) \wedge \neg achieves(X,T) &\Rightarrow attack(threat(X,T), means(X,G)) \\
achieves(Y,G) \wedge X \neq Y &\Rightarrow attack(alt\text{-}means(Y), means(X,G)) \\
goal(A,G) \wedge goal(A,T) \wedge achieves(X,G) &\Rightarrow attack(alt\text{-}goal(G), threat(X,T))
\end{aligned}$$

Essentially, these rules express the same constraints on goals, means, and problems as the frame conditions in the set of IBN frames previously introduced yet in a way that allows for generalising over different justification-attack combinations.

Based on the *justification* and *attack* predicates, we can define the core element of controlling the argumentation process which is a logically reified notion of agent intention. By adding the rules

$$\begin{aligned}
&\forall A \forall X. intends(A,X) \Leftrightarrow \exists J.\, (justification(J,X) \wedge \neg\exists C.\, attack(C,J)) \\
&\forall A \forall X. intends(A,X) \Leftrightarrow \neg intends(A, \neg X)
\end{aligned}$$

to each agent's knowledge base we make explicit that an agent is supposed to intend (from a social perspective) an action X iff there exists a justification J for X that cannot be attacked. Also, for any action X, the agent must either intend X or $\neg X$, i.e. object to perform X. In other words, unless an "invulnerable" justification can be provided for not doing X, the agent must accept to do X and vice versa. It has to be emphasised that this constitutes a "social" notion of what one should be willing to do and has nothing to do with what the agent actually wants to do from a subjective perspective.

What we also need to track the current issue that is being discussed (and which potentially changes during consecutive argumentation cycles) is a predicate $topic(D)$ which denotes that D is the point at issue. In our scenario, D is always an action, but in the general case it might also be a statement about the physical world or a mental state that is being argued about. The following rules govern the changes to this topic that occur after a certain justification has been provided:

$$\begin{array}{l} \textit{new-topic}(D, \textit{alt-means}(Y), Y) \\ \textit{new-topic}(D, \textit{alt-goal}(H), D) \\ \textit{new-topic}(D, \textit{threat}(X, H), \neg D) \\ \textit{new-topic}(D, \textit{problem}(P, X), D) \end{array}$$

The role of this change of topic in iterated argument exchange will become clear in the examples below. The same holds true of the predicate *defied*(J, D) which is used to denote that justification J for issue D has been defied before by the other party.

With this, we are ready to define the set of frames that can be used for *iterative IBN*. They are shown in table 6.13. The frames F_{RD} and F_{RR} stand for execution and rejection of the requested action X. According to F_{RD}, B has to accept unless it has previously been shown (through an exchange of arguments) that *intends*$(B, \neg X)$ is the case. After the request, *topic*$(B, \texttt{do}(X))$ is added to the repository (to make sure the topic is stored in case an unexpected reject follows) which is removed is $\texttt{do}(B, X)$ is observed. F_{RR} indicates that rejection is certain if *intends*$(B, \neg\texttt{do}(B, X))$ has been proven from A's perspective. B, on the other hand, can activate this frame unless he has been proven to intend $\texttt{do}(B, X)$.

The three actual justification-attack-concede frames require the existence of a *topic*(D) fact in the knowledge base and stand for successful justification (F_{AJC}), successful attack (F_{AJAC}), and successful challenge (F_{AC}), respectively. Using the *justification* and *attack* predicates, they control the generation of new justifications and attacks. Once a justification has been defeated by some attack, a *defied*-fact is stored in the knowledge base, so that this justification cannot be used again. Concession implies that the topic (and all *defied*-facts associated with it) is deleted, and may entail the addition of *intends*-facts depending on whether the conditions for this rule are met. More specifically,

- if an attack is successful, the challenged party has to concede that it intends the opposite of what it argued for;

- if a justification cannot be attacked, the attacking party has to give in because there exists at least one unvulnerable attack;

- if a challenge is successful no justification could be found and the attacking party wins.

Thereby, the *new-topic* predicate is used to “shift the topic” of discussion. For example, if Y is an alternative action for a goal that was named as a reason for not wanting to execute some action and F_{AJAC} is performed, then the new topic will become Y so that the requesting agent can demand this action from the peer in the next iteration.

The purpose of the last frame F_{RA}, finally, is to enable swapping roles so as to be able to challenge the original request (rather than a rejection). This happens by requesting a discussion which the other *has* to accept (as no other frame matches and the frame conditions hold trivially). Returning to the example conversation we analysed above, this could

$$F_{RD} = \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A, B, \texttt{do}(B, X)) \xrightarrow{0} \texttt{do}(B, X) \big\rangle,$$
$$\big\langle \{\neg intends(A, \neg\texttt{do}(B, X)), \neg intends(B, \neg\texttt{do}(B, X)), add(topic(\texttt{do}(B, X)))@1,$$
$$remove(topic(\texttt{do}(B, X)))@2, can(B, X)@1, effects(X)@2\} \big\rangle, \big\langle \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle$$

$$F_{RR} = \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A, B, \texttt{do}(B, X)) \xrightarrow{0} \texttt{reject}(B, A, \texttt{do}(B, X)) \big\rangle,$$
$$\big\langle \{self(A), intends(B, \neg\texttt{do}(B, X)), add(topic(\neg\texttt{do}(B, X)))@2\},$$
$$\{self(B), \neg intends(B, \texttt{do}(B, X)), add(topic(\neg\texttt{do}(B, X)))@2\} \big\rangle,$$
$$\big\langle \xrightarrow{0} \langle\rangle, \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle$$

$$F_{AJC} = \Big\langle \big\langle \xrightarrow{0} \texttt{ask-reason}(A, B, D) \xrightarrow{0} \texttt{justify}(B, A, J) \xrightarrow{0} \texttt{concede}(A, B, D) \big\rangle,$$
$$\big\langle \{self(A), topic(D), justification(J, D), \neg\exists V1.attack(V1, J),$$
$$add(intends(B, D))@3, add(intends(A, D))@3,$$
$$remove(defied(\cdot, D))@3, remove(topic(D))@3\},$$
$$\{self(B), justification(J, D), add(intends(B, D))@3, add(intends(A, D))@3,$$
$$remove(defied(\cdot, D))@3, remove(topic(D))@3\} \big\rangle, \big\langle \xrightarrow{0} \langle\rangle, \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle$$

$$F_{AJAC} = \Big\langle \big\langle \xrightarrow{0} \texttt{ask-reason}(A, B, D) \xrightarrow{0} \texttt{justify}(B, A, J) \xrightarrow{0} \texttt{attack}(A, B, C)$$
$$\xrightarrow{0} \texttt{concede}(A, B, C) \big\rangle,$$
$$\big\langle \{topic(D), justification(J, D), \neg defied(J, D)@1, attack(C, J),$$
$$add(defied(J, D))@4, new\text{-}topic(D, C, E), remove(topic(D))@4,$$
$$add(topic(E))@4\} \big\rangle, \big\langle \xrightarrow{0} \langle\rangle, \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle$$

$$F_{AC} = \Big\langle \big\langle \xrightarrow{0} \texttt{ask-reason}(A, B, D) \xrightarrow{0} \texttt{concede}(B, A, \neg D) \big\rangle,$$
$$\big\langle \{self(A), topic(D), add(intends(B, \neg D))@2, remove(topic(D))@2,$$
$$remove(defied(\cdot, D))@2\},$$
$$\{self(B), topic(D), \neg\exists V2.justification(V2, D), add(intends(B, \neg D))@2,$$
$$remove(topic(D))@2, remove(defied(\cdot, D))@2\} \big\rangle, \big\langle \xrightarrow{0} \langle\rangle, \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle$$

$$F_{RA} = \Big\langle \big\langle \xrightarrow{0} \texttt{request}(A, B, \texttt{do}(B, X)) \xrightarrow{0} \texttt{req-discuss}(B, A, \texttt{do}(B, X))$$
$$\xrightarrow{0} \texttt{acc-discuss}(A, B, \texttt{do}(B, X)) \big\rangle,$$
$$\big\langle \{add(topic(\texttt{do}(B, X)))@3\} \big\rangle, \big\langle \xrightarrow{0} \langle\rangle, \xrightarrow{0} \langle\rangle \big\rangle \Big\rangle$$

Tab. 6.13: Frame set for iterative interest-based negotiation with m²InFFrA.

be re-enacted by the sequence

$$\begin{aligned}&\texttt{request}(a,b,\texttt{do}(b,\mathit{addLink}(b,a,3)))\rightarrow\texttt{reject}(b,a,\texttt{do}(b,\mathit{addLink}(b,a,3)))\\&\rightarrow\texttt{ask-reason}(a,b,\neg\texttt{do}(b,\mathit{addLink}(b,a,3)))\\&\rightarrow\texttt{justify}(b,a,\mathit{threat}(\texttt{do}(b,\mathit{addLink}(b,a,3)),+\mathit{score}(b)))\\&\rightarrow\texttt{concede}(a,b,\mathit{threat}(\texttt{do}(b,\mathit{addLink}(b,a,3)),+\mathit{score}(b)))\end{aligned}$$

using the iterative IBN frames just introduced. But the difference is now, that although a would know that $\mathit{intends}(b,\neg\texttt{do}(b,\mathit{addLink}(b,a,3))$ is the case and is not allowed to activate F_{RD} for this reason again, he can request a different action in the same encounter (by using the procedure for trajectory concatenation described in section 5.2.2), e.g. by saying

$$\texttt{request}(a,b,\texttt{do}(b,\mathit{deleteLink}(b,a))).$$

Let us assume b wants to challenge this request, and after swapping roles by performing the sequence

$$\begin{aligned}&\texttt{request}(a,b,\texttt{do}(b,\mathit{deleteLink}(b,a)))\\&\texttt{req-discuss}(b,a,\texttt{do}(b,\mathit{deleteLink}(b,a)))\\&\texttt{acc-discuss}(a,b,\texttt{do}(b,\mathit{deleteLink}(b,a)))\end{aligned}$$

b is able to place his challenge

$$\texttt{ask-reason}(b,a,\texttt{do}(b,\mathit{deleteLink}(b,a))),$$

which is responded to by

$$\texttt{justify}(a,b,+\mathit{score}(a))$$

on the side of a. If the next move is

$$\texttt{attack}(b,a,\mathit{alt\text{-}means}(\texttt{do}(b,\mathit{modifyRating}(b,a,3)))$$

then a might not be able to find a suitable justification J that has not been defied yet, and has to concede with

$$\texttt{concede}(a,b,\mathit{alt\text{-}means}(\texttt{do}(b,\mathit{modifyRating}(b,a,3))).$$

At the same time, the topic has changed to $\mathit{modifyRating}(b,a,3)$ by applying the respective *new-topic* rule, i.e. the agents are talking about a different action now. Finally, if a desires to do so, he can start a new encounter sequence with

$$\texttt{request}(a,b,\texttt{do}(b,\mathit{modifyRating}(b,a,3))),$$

and he can be sure that b will perform the action as he has implicitly admitted that $\mathit{intends}(b,\texttt{do}(b,\mathit{modifyRating}(b,a,3))$ holds.

Although we have not compared iterative IBN frames to the previous "one-shot" IBN frame in real experiments, these examples give a vivid illustration of the levels of elaboration that can be achieved by combining different m²InFFrA frames. A comparison between the one-shot and the iterative IBN frames reveals an interesting property: In the one-shot frames, we had to explicitly list a number of different trajectories for the different types of arguments which is fairly awkward, but results in fairly simple frame condition sets. More particularly, the one shot frames allowed to discern what *kind* of justification or attack was intended by observing the performatives uttered by the other party. In contrast to this, the relationships between attacks and justifications were "hidden" in complex logical rules that cannot be directly observed in communication in the case of iterative IBN frames.

Generally speaking (i.e. not only with respect to the m^{2}InFFrA architecture), this can have a big impact on agents' ability to deal with mis-framing, as iterative IBN frames disclose no more information about the intended justification or attack than is necessary for the control flow of the interaction. At the same time, of course, they enable much more complex forms of negotiation.

Thus, it seems that the more modular and general frames we construct, the more homogeneity assumptions have to be made regarding the internal reasoning mechanisms of interacting agents. This means that a trade-off has to be achieved between generalised, elegant frame representations on the one hand and verbalising internal agent level details at the level of message surface structure on the other.

6.4 Summary

This chapter proved the adequacy of our approach by means of a thorough empirical validation of m^{2}InFFrA. First, we introduced the LIESON system, a complex software simulator that provides a suitable testbed for evaluating the interaction management and learning architecture we devised. LIESON is characterised by all the properties of a realistic application domain that matter for our purposes: complete agent *autonomy*, *openness* of the environment, incomplete knowledge, volatility of environment conditions, and the necessity for *strategic interaction* to achieve a mutually beneficial situation.

Subsequently, we discussed the experimental setup, i.e. internal parameters, utility benchmarks, frame condition handling and other practical issues that arise during the implementation of a concrete, m^{2}InFFrA-based software system.

This was followed by an extensive report on so-called basic experiments conducted with simple, proposal-based frames. The central conclusion that can be drawn from these experiments is that agents are able to learn to the strategic use a set of given frames (and also of generalised versions of these that result from frame merging as well as "broken" frames which are generated after framing failures). Furthermore, an analysis of the effects of different desirability strategies showed that, depending on the requirements of the particular application, we can trade off the importance of safeguarding agents against exploitation versus a socially cooperative attitude.

This setup was adopted for the advanced experiments without any further modification. A substantial portion of the discussion of this second series of experiments was devoted to introducing *interest-based negotiation*, a framework that allows agents to reason and argue about their goals and beliefs during negotiation. The application of m^{2}InFFrA to IBN demonstrates how frames can be developed from scratch to suit a given regime of communicative rules and conventions. The effectiveness of our approach was most strikingly illustrated by experiments in which m^{2}InFFrA agents manage to cope even with this very complex social setting. Also, these advanced experiments prove that employing the proposed social reasoning architecture in different scenarios requires only high-level (i.e. frame specification level) modifications to the architecture.

Quite naturally, we have not been able to evaluate system performance for just about *any* possible configuration, and many issues were not touched upon that would have been interesting to analyse: changing agent populations and scalability issues, different levels of communication cost, and prior frame knowledge, to name but the most obvious. However, since this is the first account of a formalised and implemented frame-based reasoning architecture, we believe that the results we have discussed are highly reassuring as concerns the potential for future improvements to the architecture.

Most importantly, this is the case because we have managed to *integrate* knowledge-level reasoning about communication with experience-based communication learning, thereby bridging the gap between agent cognition and the evolution of social interaction and communication processes.

Departing from the pure "interaction reasoning and learning" view of this (and the previous two) chapters, the following chapter shows how InFFrA can be applied to other kinds of applications.

7. Further Applications

Because of their generic character, interaction frames and InFFrA can be used as a foundation for many concrete methods and systems that lie beyond the scope of the m^{2}InFFrA-LIESON combination that our analysis so far was based on.

As examples of areas to which the framework has been or might be applied, we discuss *opponent classification in games*, *integration with supra-individual communication system approaches*, and *formal autonomy specification methods*.

After this, we list a number of more general ideas for miscellaneous possible applications of frame-based approaches to elucidate the potential of our approach from a more global perspective.

7.1 Opponent Classification

Opponent modelling is one of the core areas of multiagent learning. In large-scale MAS in which interactions among particular agents are only occasional, the ability to model some other agent so as to be able to develop an optimal interaction strategy towards him is severely confined by the fact that only very little information is available about each and every *particular* peer.

Obviously, using a classification approach to categorise other agents and build models of *opponent classes* rather than individual adversaries is one possible solution to this problem. In (Rovatsos and Wolf 2002), we proposed the ADHOC heuristic (ADaptive Heuristic for Opponent Classification) as an InFFrA-based method that aims at providing a classification mechanism independent of the concrete method applied to modelling opponent behaviour itself.

7.1.1 AdHoc

The target application domain of ADHOC is that of iterated multiagent games. We assume that agents move around on a toroidal grid, and whenever two agents meet in the same caret, they play a fixed number of l *Prisoner's Dilemma* (PD) (Luce and Raiffa 1957) games (the payoff matrix of the one-shot game is shown in table 7.1). If more than two agents meet, the pairs of agents that are going to play against each other are randomly determined. Neither of the two agents knows what strategy his opponent is pursuing, and only perceives the actions performed by both parties and the payoff he receives after each game himself. The goal of the game is, of course, to maximise one's own long-term cumulative payoff.

ADHOC assumes that an opponent modelling method OM is available which can be used to learn an adequate model of an opponent in the long run by observing the behaviour of that opponent. In our implementation, for example, we use the $US-L^*$ algo-

a_i \ a_j	C	D
C	(3,3)	(0,5)
D	(5,0)	(1,1)

Tab. 7.1: Prisoners' Dilemma payoff matrix. Matrix entries (u_i, u_j) contain the payoff values for agents a_i and a_j for a given combination of row/column action choices, respectively.

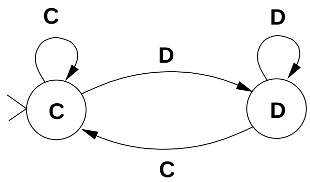


Fig. 7.1: DFA for the TIT-FOR-TAT strategy in the Prisoner's Dilemma game. Edge labels represent "own" action choices and state labels the other's reactions to these actions.

rithm proposed by Carmel and Markovitch (1998). This algorithm models every opponent as a deterministic finite automaton (DFA) that is consistent with a number of observed action sequences. Basically, this is achieved by introducing additional (hypothetical, as they are not observed themselves) internal states of the peer's decision model whenever the action sequences generated by the peer indicate a choice point at some step (i.e. if sequences differ after this step). Figure 7.1 shows a sample DFA that would be constructed by $US - L^*$ if the other was behaving according to the famous TIT-FOR-TAT strategy.

What ADHOC does is to build and maintain a (variably-sized, bounded) set of opponent classes

$$\mathcal{C} = \{c_i = \langle A_i, Q_i, S_i\rangle | i = 1, \ldots k\}$$

where each class consists of

1. a DFA A_i that models the behaviour of opponents in c_i,
2. a Q-table Q_i to learn optimal strategies against A_i (the state space of the Q-table is the state space of A_i, and its entries are updated using the rewards obtained after each of the l rounds),
3. a set of samples (recent fixed-length sequences of game moves of both players) S_i with which A_i is trained (these are collected whenever the modelling agent plays against class c_i).

Further, a similarity measure $\sigma : Agents \times \mathcal{C} \rightarrow [0;1]$ between adversaries and classes is maintained, as well as a (crisp) membership function $m : Agents \rightarrow \mathcal{C}$ that describes which opponent pertains to which class.

After a sequence e of games has been played with opponent a during an "encounter", the modelling agent updates the sample set S_i for $c_i = m(a)$ and adapts A_i if it fails to predict e correctly ($m(a)$ is initially undefined for all a). Also, the values in Q_i are updated with payoffs received during e, so that an optimal strategy is learned over time.

The classification procedure relies on the definition of a function *BestClass*$(a, e, \mathcal{C}, k, \rho)$ which retrieves the most appropriate class for opponent a given the current encounter e. The function attempts to find a class in $\mathcal{C}$ that matches e with at least similarity ρ. If no such class can be found, a new class is created, unless $|\mathcal{C}| = k$, i.e. the upper bound on the number of admissible opponent classes has been reached. If $|\mathcal{C}| = k$, the constraint on ρ is dropped and the most similar class is returned as the best candidate for classification of a. Roughly speaking, the top-level classification procedure (which is called after an encounter e is finished) is based on the following principles:

- First, all $\sigma(a, c)$-values are updated for the current opponent a and each class $c \in \mathcal{C}$. Similarities are always computed as the ratio between encounters with a correctly predicted by c and the total number of encounters with a.

- If a is an agent encountered for the first time, he is classified to any c that correctly predicts the current encounter e (this is achieved by applying $\rho = 1$ in the *BestClass* function).

- If a is a known agent, nothing needs to be done unless the opponent model (class) $m(a)$ has been modified because of e, which only happens if $m(a)$ did not correctly predict e.

- If e caused modifications to $m(a)$, the σ-values for all agents not in $m(a)$ are reset to 0 since nothing can be said about their similarity with $m(a)$ if the model for $m(a)$ has just changed; also, empty classes are erased from $\mathcal{C}$.

- If $\sigma(a, m(a))$ falls below similarity threshold δ, or if $m(a)$ has been stable for a long time, this means that $m(a)$ changed since a was classified to it and/or that $m(a)$ is a very useful class since it correctly classified many opponents. Therefore, a is re-classified to the maximally similar class, which has to be at least as similar as some threshold ρ_1 (using *BestClass*).

- Even if similarity is larger than δ we still re-classify a unless $m(a)$ is a highly stable model, but only to highly similar, highly stable classes (using a threshold $\rho_2 \gg \rho_1$ when calling *BestClass*), so that similar (or identical) classes are merged in the long run.

As concerns action choice *during* encounters, the Q-table belonging to $m(a)$ is used (with additional Boltzmann exploration) if a has been classified before. If a is encountered for the first time, the most similar class is determined *after each move* using σ-values, and the action suggested by the respective Q-table is played.

ADHOC has proved quite effective in simulation experiments: When playing against n fixed strategies, it converges to n classes that can play optimally against arbitrary numbers of adversaries as long as these play one of the n strategies. If all agents use ADHOC, on the other hand, no stable global behaviour emerges, agents make random action choices. Only if additional assumptions are made (e.g. if agents play TIT FOR TAT for a while themselves whenever the other exhibits random behaviour) can cooperation be established (we refer the interested reader to a detailed account of these results in (Rovatsos and Wolf 2002)).

7.1.2 InFFrA-based analysis

The development of ADHOC was guided by the general idea of abstracting from individual opponents in favour of looking at different types of opponents relying on the intuition that human social reasoning also uses stereotypes, especially if little information is available regarding a particular person one is interacting with.

Since ADHOC is a method to record, organise and exploit regularities in interaction processes in a socially intelligent way, we can analyse and characterise it in terms of InFFrA terminology. For this purpose, we look at different elements of InFFrA and identify the corresponding ingredients of ADHOC one by one.

Interaction Frames

The opponent classes in ADHOC can be conceptualised as interaction frames with a *trajectory model* (the automaton, a causal model of reaction to one's own actions) defined in terms of two roles whose relationship is one of mutual interdependence: One role is always fixed – the modelling agent itself; the other role is described by keeping track of all opponents who match it (this is done by the m-function). The *context model* is largely trivial: Activation and de-activation conditions are simply "being in the same caret with an opponent" and "having finished l IPD games", pre- and sustainment conditions are empty. Post-conditions are represented by means of the Q-table which captures reward expectations. As for *beliefs*, these are implicit to the architecture: Both agents know their action choices (capabilities), both know the game has a fixed length, both know that the other's choices matter. *Links* exist implicitly between all frames since they are all exclusive alternatives to each other: they share role sets, belief models and context (apart from post-conditions) and are tailored for the same kind of interaction (since there is only one type of interaction). So what makes one frame different from the other are DFAs, Q-value tables, and S_i-/σ-data stores.

The *history* of a frame is stored through the entries of the Q-table and the samples in S_i-sets that reflect past experiences with that frame. As for *status*, a role assignment takes place whenever encountering an agent by using A_i to predict its behaviour, and trajectory status is updated by state changes in the DFA during play. Tracking context status is trivial except for the Q-update, but the update of the σ-function as well as re-setting their values in case of DFA modification track framing experience across frames, and they actually manipulate *all* frames' private attributes simultaneously. More particularly, the *difference model* that is represented by σ-values is constantly computed for *all* frames with respect to the current interaction situation (cf. below).

Framing

As concerns framing, InFFrA cannot only be used to model the processing steps of ADHOC sketched above, but also to identify *weaknesses* and *advantages* of the heuristic and this underlines the usefulness of InFFrA for improving existing systems.

To achieve this, we need to distinguish between (i) the case in which the current opponent a has been encountered before and (ii) the case in which we are confronted with an unknown adversary.

Case (i): The *matching process* occurs at the start of every encounter; $m(a)$ is chosen (blindly) as the most appropriate frame from the repository and is activated. This choice is then never altered during the encounter and this is a first disadvantage of the system, because no frame assessment and adjustment occurs *during* encounters, thus limiting the adaptability of the modelling agents severely within the current interaction.

In *situation interpretation*, the current sequence of moves (the *perceived frame*) is recorded, stored in $S_{m(a)}$ and the entries in the Q-table are updated according to recent payoffs. *Frame matching* consists of updating similarity values for *all* frames with respect to a. Compared to the InFFrA intuition, this is a much more complex matching activity, since it compares the difference model with all classes, so that the lack of framing assessment is partly made up for by adding complexity at the frame matching stage.

As mentioned, *frame assessment* and *re-framing* occur only *after* the encounter: Frame validity is assessed according to whether the current sequence of opponent moves is understood by the DFA in $m(a)$ or not. Here, we observe a second drawback of the prototype: Adequacy and desirability assessment is clearly under-developed, since neither consistency of Q-values nor the expected usefulness (e.g. expected future payoff with that class) of frames is taken into account. For example, the classification heuristic would not be able to cope with types of opponents whose actions have different utility outcomes for the modelling agent. If, e.g., own payoff matrix entries differed across opponents, those opponents would still end up being classified identically if they perform identical actions. Also, since the agent has no choices regarding partner selection, it does not make sense to weigh the desirability of entire Q-tables against each other.

As a consequence, the *framing decision* itself has only effects on future encounters with the same agent. It depends on the simple criterion of whether the DFA of c has just been modified or not. If so, the *frame adjustment module* comes into play: It potentially reclassifies a, creates a new class for it and resets similarity values for non-members of c. At the same time, it seeks to retain highly stable classes and to merge similar classes in the long run. This is undoubtedly the most elaborate component in the opponent classification MAS, and it nicely illustrates the possibilities of a long-term organisation of interaction experience, especially because, in a boundedly rational manner, it tries to distinguish between frames only where necessary. *Trial instantiation*, on the other hand, is very simple: It can be trivially reduced to using the maximally (and highly) similar candidate frame as the new value for $m(a)$, because similarities between a and all classes are constantly tracked.

Frame enactment is performed by tracking opponents' actions in the current DFA and by selecting the next move according to the Q-table. Then, since there are no other reasoning levels to compete with, *behaviour generation* is straightforward. In this enactment stage, a third shortcoming can be identified that probably explains the lack of structure in interaction among ADHOC agents described above. It lies in the fact that the frames impose no restrictions on the action selection mode of the modelling agent himself – in fact, the trajectory represented by the DFA prescribes only the actions of the opponent, and the modelling agent is merely optimising its behaviour towards that opponent. Therefore, since no agent feels it should comply with some more specific pattern of behaviour, no recurring efficient interaction patterns can emerge unless the ADHOC agent is playing against fixed-strategy opponents.

Case (ii): In the case of unknown opponents, frame assessment and re-framing is implemented as in the previous case. The differences lie in matching and in making framing decisions, which occurs after each round of the encounter (and not only after the entire encounter). After each round, the modelling agent activates the most similar class with respect to the current sequence of moves and uses this class for enactment decisions (according to the respective Q-table).

Again, this illustrates the implementation of the bounded rationality principle: The framing effort is in this case much greater (and adheres much more to InFFrA requirements), given that interactions with unknown agents are much riskier than those with known adversaries. This suggests that an extension of the opponent classification heuristic that also allows for re-framing *during* encounters in case (i) might increase agent performance, yet at a greater computational cost.

This evaluation provides evidence for the practical use of InFFrA as a conceptual framework that supports the analysis of existing MAS by decomposing social reasoning algorithms into different functional components. Starting from the functionality of these components, we can identify advantages and shortcomings of the analysed systems and suggest improvements.

Of course, our framework is not applicable to just about any social reasoning architecture. ADHOC is an approach which, like InFFrA, focuses on managing different types of interaction patterns and on how to exploit these in a strategic way. Also, like InFFrA, it strongly relies on social abstraction and transient social optimality (section 5.1.1) itself. Our analysis shows that InFFrA provides the appropriate modelling tools *for this kind* of social reasoning methods.

7.2 Integration with Communication Systems

In section 4.1, we already discussed *communication systems* (CSs) informally and explained how they can be seen as a foundation for the empirical semantics of m²InFFrA developed in 4. As has been mentioned there, we have developed a full-fledged formal model of CSs elsewhere (Nickles and Rovatsos 2004, Nickles et al. 2004b, Nickles et al. 2004a) that is based on defining how a knowledge-based entity that observes communication in a MAS can use an *expectation network* (EN) to model the empirical semantics of communication and its evolution in that MAS.

In the formal model suggested in (Nickles and Rovatsos 2004) and then further adapted in (Nickles et al. 2004a), ENs are modelled as probabilistic trees that capture continuation probabilities between subsequent message patterns by labelling edges with transition probabilities, while edges or sets of edges can additionally be conditioned with logical constraints and/or variable substitutions. Apart from so-called *cognitive* edges that are derived from observation and are thought to represent the actual communicative behaviour of agents, ENs may also contain *normative* edges which may not directly relate to observation but express an additional "normative force" that influences certain continuation probabilities without empirical evidence. This may be useful when the CS has certain expectations that have not been fulfilled yet and it is reasonable to make some *a priori* assumptions regarding the effects of certain messages. For example, it may be realistic to assume that an agent who has been mislead by some other agent is unlikely to communicate with that agent again.

Figure 7.2 gives an example of an EN in the CS framework with paths for different kinds of conversations, partially conditioned by certain logical conditions and/or variable sub-

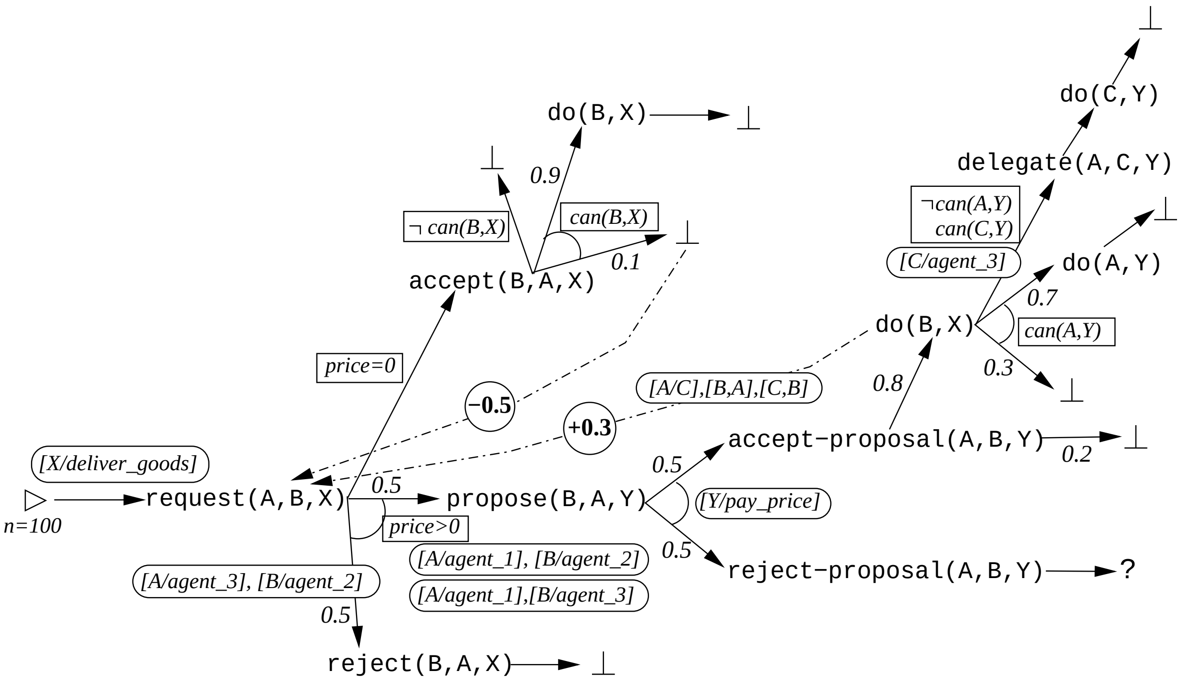


Fig. 7.2: An expectation network. Nodes are labelled with message templates and the special symbols "▷", "⊥" and "?" that indicate conversation initiation, termination and "don't know" semantics, respectively. Nodes are connected by (solid) cognitive edges labelled with numerical probabilities in *italic* font, or (dashed) normative edges with round circles containing a numerical "force" value in **bold** face. Substitution lists/conditions belonging to edges appear in rounded/edged boxes near the edge. If neighbouring edges share a condition this is indicated by a drawn angle between these edges.

stitutions (whose semantics is that the paths are only relevant if the respective conditions are fulfilled at present and/or if the variables have the respective values).

Normative (dashed) edges are labelled with a numerical "force" that determines whether and to which degree they increase or decrease the probability of their target node when a conversation is observed that has lead to their source node. Note that an EN augmented by normative edges need not be a tree, as these edges may point to ancestor nodes of certain messages (in the example above, the "start" node of a conversation that leads to the same path would become less likely after the utterance that lead the agent to realise he is being betrayed and this node is obviously a descendant of the "start" node). Also, normative edges may have an impact across different paths on the tree, if, e.g., one conversation makes another one more likely (the betrayed agent can be expected to file a complaint against the fraudulent one to a third party, for instance).

Special labels are used to denote encounter initiation, encounter termination and "don't know" semantics. Messages have these "don't know" semantics if they (and their consequences) have never been observed before, or if there is such high variation in their potential consequences that no clear semantics can be discerned.

The complete CS formalism, which we will not go into here for lack of space, provides a framework for defining all aspects that are relevant to processing such ENs:

- It allows the designer to specify how the EN is updated upon observation of a new message. This involves:
  - Initialising the EN at the start of the observation process, where it need not necessarily be empty, but may contain prior expectations of the observer.
  - Identifying whether the new message is a continuation of an existing path (and, if so, how to determine that path) or whether it marks the start of a new encounter.
  - Defining update rules for edge weights. These weights might simply measure frequencies of certain observations, but there might also be some bias towards certain expectations, e.g. if certain continuations are ignored as "misunderstandings" or particularly highlighted as "desirable" courses of interaction.
- It can be used to project expectations about future communication using the current EN, and to weigh the importance of normative edges appropriately, depending on how the information the EN provides will be used in the actual system.
- It enables the designer to define rules of long-term EN management, which may involve (for example) pruning paths that occur very rarely, coercing existing paths into more abstract ones whenever appropriate, splitting overly long paths when a certain modularity of "parts of a dialogue" is discernible, etc.
- It provides means of incorporating the world knowledge of the observer into the process of updating and using the EN.

What is important to understand is that CS can be used to construct *active* observers of communication which evolve their own model of communicative expectations over time and in doing so take their own beliefs and goals into account. This means that the observer does not necessarily maintain a one-to-one picture of the correlations between communicative actions in a system, but that he may derive a biased, incomplete or even "wrong" view of the ongoing communication if this suits his own purposes.

Using the CS approach we can endow one ore more system components with knowledge about the regularities that govern communication in a MAS, which can then be used for different purposes:

- A CS can be used to develop a *social system mirror* (Lorentzen and Nickles 2001) that records these regularities and provides agents with information about social structures by virtue of "reflecting" what has been observed globally to the individual agent. Each agent can use this information to guide his own communication behaviour.
- At an agent-oriented software engineering level, such mirrors can be utilised as a CASE tool by human software designers to manage the evolving behaviour of open systems, as has been suggested in (Brauer et al. 2001). The EXPAND (EXPectation-oriented ANalysis and Design) method proposed there is based on using a CS view of a MAS to monitor how agents communicate while the system is in operation. This is claimed to be an adequate method for systems in which the internal design of the agents cannot be assumed to be known by the designer of the MAS (different agents

may represent different stakeholders, they may have been designed by one person but changed their behaviour due to adaptiveness, etc.) With the information derived from the respective CS, the designer can take appropriate measures without restricting agent autonomy if the MAS exhibits undesirable behaviour. For example, it can manipulate the EN reflected back to the agents, so that they take different communication norms into account when making communicative decisions.

- Expanding upon the mirror concept, *mirror-holons* were proposed in (Nickles and Weiss 2004) to act as middle-agents that may even restrict other agents' autonomy and reduce them to normal distributed procedures whenever appropriate. The idea here is that once an efficient and stable global pattern of behaviour has emerged and has been identified by the mirror, the whole MAS can switch to a "normal distributed system with centralised control" mode, and the mirror (which now embodies, in a sense, all agents' local functionalities and can therefore be regarded as a holon) can simply "execute" this pattern. When the environment changes and decentralised control makes sense again, agents may re-gain their autonomy, and this process can be iterated as necessary.

A variety of other applications have been sketched in (Nickles and Rovatsos 2004), (Nickles et al. 2004b), and (Nickles and Weiß 2003). What is common to most of them is that even if observation takes place at a supra-individual level, the knowledge that is derived for them should eventually be used by agents. So, the question arises of *how to get EN knowledge of a CS into agents' "heads"?* – our frame-based approach is an answer to this question.

As we have shown in (Nickles and Rovatsos 2004), it is not possible to convert any given EN into a frame repository with identical continuation-predicting semantics, while the converse mapping can be easily achieved (by simply turning each frame trajectory into a path, labelling edges with appropriate conditions and substitutions and deriving edge weights from trajectory and substitution counters). This is also what we would expect as CSs constitute a framework that is much more general than that of interaction frames. However, we can approximate a given EN (or a sub-portion of it, since the global system of communication regularities will neither be manageable nor interesting for a single agent) by a set of m^{2}InFFrA frames such that the continuation probabilities derived from the respective frame repository correspond to those that the original EN would have predicted.

This paves the way for employing m^{2}InFFrA in the CS applications listed above, and it supports the claim that InFFrA has the capacity of *mediating* between global social knowledge (potentially gathered using a much more global perspective than that of a single agent) on the one hand and its cognitive processing on the other.

7.3 Integration with RNS

When describing a social context, InFFrA mostly relies on defining this context in terms of the surface structure of communication patterns, i.e. it is primarily trajectory-centred.

The *roles, norms & sanctions* (RNS) framework (Nickles et al. 2002, Weiß et al. 2003) takes an alternative approach by providing a formal schema for specifying actors in a MAS in terms of the roles these agents may fill and what activities are associated with these roles. In that, the central aim of RNS is to enable a precise and fine-grained specification of computational *autonomy* by defining each activity in terms of *norms* and *sanctions* associated with it. In other words, RNS is a formalism for the role-based definition of behavioural expectations.

In RNS, the social frame of a MAS is defined in terms of a *role space* that is a collection of *roles*. Each role, in turn, is defined through a set of *activities*, and specifying such an activity requires defining the *norms* and *sanctions* that the activity is subject to. To be a bit more specific, each activity is associated with a set of norm-sanction pairs (so-called *status statements*) which define its *status range*. The syntax for these statements is

<*status_type*>: NORM <*norm_type*> <*condition*> + SANC <*sanction_type*> <*sanction*>

where

- *status_type* can be used to indicate whether the validity of the respective norm-sanction pair DEPends on someone who will request the activity or holds INDependently of any request,
- *norm_type* explains whether the statement expresses a **P**ermission, **O**bligation or **I**nterdiction to perform the activity and *condition* describes under which conditions the statement becomes relevant, and
- *sanction_type* describes whether the sanction *sanction* exerted on someone failing to comply with the norm is a **RE**ward or a **PU**nishment (i.e. a positive or negative sanction).

Informally speaking, such a status statement has the following semantics: If it is attached to activity A and activity A makes part of role R, then any agent who fills role R has to adhere to the norm specified in this statement. If the sanction is of type **RE**, the agent will be rewarded by the "sanction" specified in the statement for fulfilling the norm. If it is of type **PU**, he will have to face the specified punishment for not adhering to the norm.

RNS supports four different kinds of activities:

1. Basic activities concerned with resource and event handling in the environment,
2. request activities, i.e. requests for execution of activities by others,
3. sanctioning activities that result in punishing or rewarding other agents, and
4. change activities which change the norms and sanctions associated with certain activities.

This makes RNS a highly expressive tool for specifying what agents are obliged, permitted and prohibited to do in terms of executing actions in the environment, requesting things from others, reacting to norm fulfilment and violation, and even altering the normative frame of a MAS.

Instead of going into the details of all four types, table 7.2 provides two examples for a basic activity and a request activity. In the first activity specification, three statements specify who is entitled to deliver a material at a certain quantity (this activity might make part of, say, a "producer" role). The first status statement says that without being asked for it, any producer is permitted to deliver, and no sanction or reward is associated with not delivering something at one's own initiative (keyword NO). The second and third statements depend on whether the producer is asked by someone else to deliver. The second

```
ACT deliver ( material,quantity )
    { STATUS RANGE
        <IND> : NORM <P> <NO> + SANC <NO> <NO>
        <DEP EACH> : NORM <O> <quantity ≤ 100> + SANC <PU> <withdraw_role>
        <DEP AssemblyMg> : NORM <I> <material = steel> + SANC <PU> <pay_fine>
    }

ACT REQUEST ( EACH ; USsupplier ; NOT deliver ( material, quantity ) )
    { STATUS RANGE
        <IND> : NORM <P> < (material = steel) AND (rating(material) = poor)> +
                  SANC <NO> <NO>
      NORMATIVE IMPACT
        NORM <I> <material = steel>
    }
```

Tab. 7.2: Examples of activity definitions in the RNS schema

statement expresses that if *any* (keyword EACH) agent requests a delivery, the producer is obliged to deliver if the quantity is less than 100. Otherwise, the producer role will be withdrawn from the agent. The third norm-sanction pair is even more specific: If an assembly manager agent (role AssemblyMg) requests the delivery of steel, the producer must not deliver otherwise he will have to pay a fine.[1]

More complex still, the request activity specification in the lower part of the table defines rules for any (keyword EACH) agent filling the USsupplier role under which he may ask the producer NOT to deliver. In addition to their status range, such request actions also have a *normative impact*[2] which describes the norms induced by a request. In our example, the request not to deliver would result in an interdiction to deliver steel. The status range of the activity contains an independent statement according to which the request may be issued if the material is steel and is of poor quality. No sanctions apply if such a request is not issued.

How does all this relate to InFFrA? As mentioned before, the central modelling primitive of RNS are roles, i.e. it is an actor-centric approach, while InFFrA is more process-oriented as models of frames are built around patterns of communication. Although these two approaches seem quite orthogonal, there are two reasons why it makes sense to integrate them:

1. Both aspects (the specification of *agents* and that of *interaction mechanisms*) play a crucial role in the practical process of engineering agent-based systems. It is quite impossible to design a MAS without thinking about the (types of) agents involved *and* the languages and protocols they will use to interact and communicate.

2. Both RNS and InFFrA pin down their typification of social knowledge on *behavioural expectations*, whether these be activities attached to roles (in the case of RNS) or frame trajectories (in the case of InFFrA).

[1] Note that in a complete RNS specification, we would have to define roles and activities for these sanctions elsewhere.

[2] Similarly, sanctioning activities have a *sanctioning impact* (i.e. the execution of some concrete sanctioning action), and change activities have a *status impact* that effects a change in the status range of some activity (by deleting, adding or replacing norms/sanctions).

To sketch what this process of integration might look like, let us assume a role space defined in RNS describes the desired behaviour of agents who fill certain roles. Typically, such a specification is part of the (early) analysis and design phase of MAS development, as it describes *what* agents should be allowed and disallowed to do, but not *how* this is going to be achieved.

More specifically, once the rules are available that define the autonomy status of each role, we need some mechanism to confer them on agent cognition in a way such that they can be reasoned about. Note that this does *not* mean that agents should be made to blindly execute the norms associated with activities. In fact, it is one of the strengths of RNS that it specifies the consequences of not adhering to normative rules rather than excluding this possibility *a priori* and that, in this sense, it constitutes an *autonomy-respecting* approach.

This is where InFFrA comes in: If we are able to generate interaction frames from an existing RNS specification, we can use InFFrA to strategically reason about the normative context in a social system. This is because InFFrA agents are able to take this social context into account, without necessarily abiding by existing normative rules. If normative knowledge is available in the form of a set of pre-specified frames, agents can autonomously decide when to use these frames based on their own goals and on the social consequences that can be expected from certain actions.

So the question is: How can we construct frames that correspond to a given RNS role space? Essentially, we have to look at the actual communicative processes that result from the norms and sanctions specified for each activity and associate them with the respective roles. For this, we can proceed as follows:

1. The communication language must be augmented by symbols for those physical actions that need to be referred to in communication because they appear in norm and sanction specifications. These actions should be used as content values of `do`-messages (or messages marked as "physical actions" in some other way).[3] This must be done for
   (a) basic activities, as these constitute the primitives of environment-manipulating actions and
   (b) all punishing and rewarding actions that appear in the SANC parts of status statements.
2. Statements of the form $role(X, R)$ must be introduced for all roles in the role space and all agents in the system, and the logical language used by agents must be extended to reason about these. Likewise, it must be ensured that all conditions occurring in the NORM parts of all status statements can be handled properly.
3. In RNS, the basic communicative actions are requests and sanctions, so frames have to be devised for each request-action and request-sanction pair. This can be achieved by performing the following procedure:
   (a) For each SANC part of a status statement of type IND related to a basic activity A, a frame is created with the trajectory
$$\langle \mathtt{do}(X, A) \rightarrow \mathtt{do}(Y, S) \rangle$$

[3] Apart from these actions that occur in the RNS specification there may be, of course, other physical actions that manipulate the environment. However, as these are not subject to normative expectations, they need not be used in the language in which we express frame trajectories.

where S is the sanction and X, Y are variables for the respective agents.

(b) For each `SANC` part of a status statement of type `DEP` related to a basic activity A, a frame is created with the trajectory

- $\langle \texttt{request}(Y, A) \rightarrow \texttt{reject}(X, A) \rightarrow \texttt{do}(Y, S)\rangle$ if the norm type is **O**
- $\langle \texttt{request}(Y, \neg A) \rightarrow \texttt{do}(X, A) \rightarrow \texttt{do}(Y, S)\rangle$ if the norm type is **I**, and
- $\langle \texttt{request}(Y, A) \rightarrow \texttt{do}(X, A) \rightarrow \texttt{do}(Y, S)\rangle$ if the norm type is **P**

with S, X, and Y as before.

(c) The roles & relationships slot of each of these frames must contain conditions which describe that X fills role R if R is the role to the status range of which the original status statement pertains. Constraints on agents Y and their roles must be derived from the corresponding request (`ACT REQUEST`-marked) definitions in RNS.[4]

(d) The conditions attached to the `NORM` part of each statement must be included in the context slot (or beliefs slot, if they concern epistemic states) of the corresponding frame.

We should take a minute to illustrate the frame construction procedure using the RNS specification of table 7.2 as an example. If we employ m²InFFrA notation, converting the activity specification to frames would result in the set of frames shown in table 7.3. F_1 and F_2 correspond to the first `DEP`endent status statement which obliges the producer (we assume that the activity specifications of "deliver" pertain to the definition of a *Producer* role) to deliver if the quantity is less than 100. To express norm-compliant and deviant behaviour, we obviously need two frames. Note also the use of a *sanction*(X, Y, S) symbol to express that a sanction is executed. As we do not have any further information about who will execute the sanction, we assume that Y can perform it himself.

Frame F_3 refers to the status statement that prohibits delivery of steel to any *AssemblyMg* agent. This time, for the sake of the example, we shall assume that Z is a third party (here: a *Supervisor* agent) that has to implement the sanction.[5] This frame nicely illustrates the indeterminacy of agent behaviour in open systems, as it does contain the sanction but not a possibility to *preclude* the undesired behaviour.

The last frame F_4 captures the request activity of table 7.2. If a *USsupplier* agent requests the producer X not to deliver steel of poor quality and X still does so, Y may sanction by filing a formal complaint. It is noteworthy that the RNS request activity specification does not mention this sanction but only the normative impact that follows from the request activity (i.e. the interdiction to deliver). Therefore, we have included a sanction in F_4 that would correspond to the following RNS sanctioning activity specification:

[4] Note that RNS allows for a multi-perspective specification of status range for a single activity. As in the example of table 7.2, the norms associated with the requesting party can be specified independently from those of the requestee. If the specification of the request activity (or of a sanctioning activity) is omitted, then obviously no role constraints apply to the agents who can issue the request or perform the sanction.

[5] This implies we have to give up the strict turn-taking principles of m²InFFrA which would necessitate minor modifications to the algorithms we have presented in previous chapters (all "other" parties can actually be regarded as a single conversation party, so this will not make a big difference). Another alternative would be to split the frame into two parts such that the sanctioning dialogue happens outside the context of the initial (deviant) action.

$$
\begin{aligned}
F_1 = \; & \Big\langle \langle \xrightarrow{0} \texttt{request}(Y, X, \mathit{deliver}(M, Q)) \xrightarrow{0} \texttt{do}(X, \mathit{deliver}(M, Q)) \rangle, \\
& \langle \{\mathit{role}(X, \mathit{Producer}), Q \leq 100\} \rangle, \langle \xrightarrow{0} \langle\rangle \rangle \Big\rangle \\
F_2 = \; & \Big\langle \langle \xrightarrow{0} \texttt{request}(Y, X, \mathit{deliver}(M, Q)) \xrightarrow{0} \texttt{reject}(X, Y, \mathit{deliver}(M, Q)) \\
& \quad \xrightarrow{0} \texttt{do}(Y, \mathit{sanction}(Y, X, \mathit{withdraw\_role})) \rangle, \\
& \langle \{\mathit{role}(X, \mathit{Producer}), Q \leq 100\} \rangle, \langle \xrightarrow{0} \langle\rangle \rangle \Big\rangle \\
F_3 = \; & \Big\langle \langle \xrightarrow{0} \texttt{request}(Y, X, \mathit{deliver}(M, Q)) \xrightarrow{0} \texttt{do}(X, \mathit{deliver}(M, Q)) \\
& \quad \xrightarrow{0} \texttt{do}(Z, \mathit{sanction}(Z, X, \mathit{pay\_fine})) \rangle, \\
& \langle \{\mathit{role}(X, \mathit{Producer}), \mathit{role}(Y, \mathit{AssemblyMg}), \mathit{role}(Z, \mathit{Supervisor}), M = \mathit{steel}\} \rangle, \\
& \quad \langle \xrightarrow{0} \langle\rangle \rangle \Big\rangle \\
F_4 = \; & \Big\langle \langle \xrightarrow{0} \texttt{request}(Y, X, \neg \mathit{deliver}(M, Q)) \xrightarrow{0} \texttt{do}(X, \mathit{deliver}(M, Q)) \\
& \quad \xrightarrow{0} \texttt{do}(Y, \mathit{sanction}(Y, X, \mathit{formal\_complaint})) \rangle, \\
& \langle \{\mathit{role}(X, \mathit{Producer}), \mathit{role}(Y, \mathit{USsupplier}), M = \mathit{steel}, \mathit{rating}(M) = \mathit{poor}\} \rangle, \\
& \quad \langle \xrightarrow{0} \langle\rangle \rangle \Big\rangle
\end{aligned}
$$

Tab. 7.3: Frames derived from the RNS specification in table 7.2

```
ACT SANCTION ( EACH ; EACH ; NOT deliver ; NORM <I> <material=steel> )
    { STATUS RANGE
        <IND> : NORM <P> <NO> + SANC <NO>
      SANCTIONING IMPACT
        SANC <PU> <formal_complaint>
    }
```

It states that EACH agent who fills EACH (i.e. any) role may punish someone who violates an interdiction to deliver steel by filing a formal complaint. This example also shows how constructing interaction frames can aid the process of filling the gaps that may exist in an RNS specification.

While these examples illustrate how the static specifications of normative rules in RNS can be turned into specifications of communication processes by using InFFrA, this transformation is of course by no means complete or devoid of problem issues:

- One problem is how to deal with change activities. These are different from basic, request and sanctioning activities in the sense that they do not involve overt action. Therefore, the question arises how these changes of normative activities can be brought into the realm of communication. It turns out that this is actually a very important question, because if one agent is entitled to alter the normative context of a system, other agents can only successfully adapt to these changes if they have been made explicit in communication.

In general terms, this can be achieved by either (i) introducing generic frames for the public announcement of changes in the status range of activities (which can be used at the time of these changes) or (ii) by informing others about the current expectations associated with an activity *during* a conversation. We will return to these issues in section 8.2.

- Another issue is how to handle the combination of **O** and IND in the status specification of an activity. If the norm type is **O** it hardly makes sense to think of IND statements because this would force the agent to execute the action all the time (therefore IND usually only appears in statements with norm types **I** and **P**). However, a specification of what to do in this case would be necessary if we were to provide a complete procedure for transforming RNS schemata into frames.

- So far, InFFrA and m²InFFrA frames were assumed to describe the structure of communication processes within the temporal scope of a single conversation. In contrast to this, RNS norms are supposed to be valid for an extended period of time. If, for example, one agent is supposed to refrain from doing something that was forbidden to him by someone else, then the sanction should follow if he ever performs the prohibited action again, even after the current conversation.

  This necessitates a different form of conversation management from that implemented in m²InFFrA such that agents can monitor actions performed with the context of requests and sanctions in the long term and not de-activate that frame in the meantime. What this entails is that the agent must be able to activate several frames at a time (one for each norm-inducing conversation that has not been completed by a sanction yet).

- As mentioned in footnote 4, RNS allows for the specification of norms and sanctions from several different perspectives that concern a single activity. For example, a norm that DEPends on someone's request for a basic activity should be included both in the definition of that basic activity as in the specification of the request activity.

  With this respect, we should emphasise that InFFrA can by no means ensure that all norms are defined precisely enough or that no conflicts between different norms will occur (although it may occasionally aid this process, as in the example of F_4). In other words, if the original RNS specification is flawed, the resulting frames might be unreasonable or not usable for the agents.

Still, we believe that both RNS and InFFrA can profit from combining the two approaches, and that together, they provide very powerful tools both for the deontic specification of a social context as for its realisation in terms of communicative conventions. Most importantly, InFFrA enables us to transform the *semantics* of expectations defined in an RNS specification into socio-cognitive patterns rather than to merely support the derivation of syntactical aspects interaction mechanisms (messages, protocols) that are needed to implement what has been previously defined in RNS.

7.4 InFFrA in different application scenarios

The previous sections have described three approaches that can be analysed, extended or combined with InFFrA so as to obtain more powerful methods for social modelling and reasoning. Next, we are going to look into practical application scenarios that are well-suited for InFFrA *as such* to give a feeling for the wide range of possibilities to use the framework in practice.

7.4.1 Markets and organisations

The experiments we have conducted so far dealt exclusively with negotiation among loosely coupled agents where occasional interaction is the rule. Also, the negotiations were conducted in a fairly *voluntaristic* vein in the sense that social commitment was very fragile as agents may opt in and out of the agreements they make by violating their previous commitments after communicating, and because there is no need to compromise one's private goals for the sake of social coherence.

Two very important "arenas" of social interaction in which these assumptions do not hold and which play a very important role in human society and in a variety of artificial multiagent systems are *markets* and *organisations*.

Markets enable complex forms of interaction by means of *economic exchange*. By using money as a universal medium of exchanging utility, agents are no more restricted to merely performing those actions that are either profitable for themselves (and such options are rare in many applications as most physical actions involve some effort) or to actions for which direct reciprocal exchange ("I will do X for you if you do Y for me") that is mutually beneficial is available. This is because economic exchange enables agents to trade monetary value for their own efforts or loss of resources since they can use money to achieve other goals when entering other interactions in the future. Quite naturally, market-based systems and the game-theoretic and decision-theoretic principles that are used to model them constitute one of the major areas of research in distributed AI (see, e.g. (Sandholm 1999)). Most of these approaches seek to devise interaction mechanisms that fulfil certain optimality criteria and are safe against fraudulent behaviour in the sense that such behaviour would be either irrational or eventually uncovered. Quite surprisingly, the issue of how to learn the *regularities* of agent behaviour in markets (i.e. the market *culture*) has received comparatively little attention in the literature. Although there exist a few approaches (such as, for example, the work of Tesauro and Bredin (2002) in which agents learn optimal bidding strategies in auctions using reinforcement learning), there is still a lot of work to be done in this area.

InFFrA would be very suitable for this purpose, and there are several reasons for this:

- As we have shown in the m²InFFrA model, it is possible to incorporate utility-based considerations that are backed by decision-theoretic principles in the framing process. In this way, we can combine rational reasoning with empirical reasoning about the regularities that govern the behaviour of actors in a market.

- The generalisation mechanisms used in m²InFFrA enable agents to make "as many distinctions as necessary" in the long run. Most market-based scenarios allow for different degrees of *volatility* of interactions (ranging from such occasional interactions with the same set of interaction partners as in auctions to recurring negotiations with long-lived business partners). With this respect, it would be possible for frame-based agents to learn very general regularities for interactions that occur only occasionally while being specific about those types of interactions whose instances resemble each other very much.

- Frames enable agents to combine the economic transaction itself with knowledge about the whole communicative setting. They can correlate the observed behavioural patterns with attributes of the situation other than "who payed which price and what they got for it" or "who made which bid at which stage of the auction". This can be very important in complex markets, because at the bottom line, any economic transaction takes place within a wider societal context.

Interestingly, the shortcoming of overlooking the importance of actual behaviour and experience with interaction that can be observed in the area of market-based systems is paralleled by a similar phenomenon in the field of computational organisation theory (Carley and Gasser 1999). There, researchers are mostly concerned with the *structural* aspects of organisations (relationships, organisational rules, etc.) and only to a lesser extent with the *evolution of interaction processes* among members of an organisation (and their organisation-external interaction partners). More specifically, once the organisational structure has been designed, the problem of how agents can deal with it remains largely unresolved.

Organisations are probably one of the most complex social entities in human society that a single individual can handle.[6] In our view, interaction frames provide an ideal method to facilitate this process for artificial agents and artificial organisations. First of all, the framework laid out in chapter 3 and the complex examples given there demonstrate that abstract InFFrA provides a variety of modelling tools that cover all relevant aspects to model complex organisational processes. Secondly, InFFrA constitutes the missing link between organisational structure and its cognitive processing. The designer can model the organisation, define the respective communicative processes, and then endow InFFrA agents with knowledge about these processes so that they can properly interact with the organisation. Thirdly, the idea of accepting existing social procedures but employing them strategically to further one's goals (rather than constantly modifying them) is probably more adequate for organisational settings than for any other level of sociality. This is because organisations typically constitute that level of social exchange at which a set of rules has to be accepted and complied with (and one usually commits to doing so when signing the work contract) but must be used in whichever way is most useful for the rationally reasoning stakeholder who, after all, is pursuing his own agenda apart from that of the organisation.

Finally, organisations possess the richest collection of complex forms of interaction (workflow & business processes, different kinds of meetings, organised teamwork, negotiation and contracting, delegation and reporting procedures, lobbying, informal discussions, instruction and training, cooperative problem solving and knowledge management to mention but a few). So, without doubt, they constitute one of the most challenging fields of application for designing interaction frames and framing mechanisms.

7.4.2 Human-computer interaction

Until now, our analysis has been restricted to communication and interaction among artificial agents. Alternatively, it is possible to use InFFrA to provide agents with knowledge

[6] In fact, it is often the case that even the complexity of the portion of an organisation that is relevant to a particular person either belonging to or interacting with the organisation exceeds the cognitive capacities of that person.

regarding communicative conventions that exist in human societies so as to facilitate natural and socially appropriate communication between artificial agents (digital assistants, user interface agents, etc.) and human users.

The advantage of using a frame-based approach for this kind of application is twofold:

1. The surface structure of dialogues which encapsulates the social rules of interaction among humans can be turned into frames without agents actually having to "understand" human culture. It suffices if the context conditions and agents' internal goals and beliefs link the use of these conventions to core agent activities in an appropriate way.

   For the human user, this might also add to the "lifelike" qualities of the agent, since it suffices for an agent to merely reproduce existing social practice in order to appear socially competent.

2. By its modular design, InFFrA enables the designer to exchange different sets of frames at will without having to modify any other part of the agent's functionality (if the purpose of the agent is otherwise the same, that is).

   This can be very valuable if the agent is to be integrated in a new social context (for example when being used with adults instead of children).

The central challenge for adapting our framework to fit human-computer communication lies, of course, in augmenting the simple speech-act based communication language we have used by elements of natural language.

Even the most general ideas about how this can be achieved lie certainly beyond the scope of this thesis. All we can say at this point is that we would profit from the modularity of InFFrA if we were to include elements of natural language in interaction frames, as this process would only affect the surface structure of communication and bear only very little effects on the remaining parts of InFFrA frames.

7.4.3 Semantic Web technologies

The final application area that we would like to mention (and for which we are only going to give the most general of ideas) is the Semantic Web (SW) (Berners-Lee et al. 2001) and SW-related technologies and, more particularly, the prospect of interaction-oriented meta-data.

From a conventional point of view, the World Wide Web is a huge collection of digitally stored information that can be globally accessed – so-called *content* – mostly presented in the form of text, images and videos or animations. The basic goal of the SW endeavour is to make this huge amount of content more accessible, manageable and easily searchable by providing machine-readable meta-data, i.e. data that provides information *about* the content while being suitable for computer processing.

Currently, the cornerstones of SW technologies that are expected to contribute to this goal are: ontologies (that provide the shared vocabulary in which content is described), protocols (for accessing data and services in open system) and reasoning mechanisms (which allow computational agents to process the information they find on Web sites or obtain from other agents).

Seen from a different angle, however, the "semantics" of the Web is not given by the content that is provided on Web sites and related Internet services but lies in the *use* that is made of this content by those who access it. In other words, what matters is the *interaction* between human users and owners of Web sites, service providers, artificial agents crawling the Web, etc.

And this is where interaction frames come in: They can be used to model observed interaction processes, and the knowledge about these interaction processes can then be presented in a machine-readable format. For example, an observer agent who uses InFFrA could record the interactions that take place between clients accessing a service on a Web site and the service provider. The resulting interaction patterns can be made available as a frame repository on that site so that any agent can inspect them to understand the prevalent interaction practice and communicative conventions.

Such applications would strongly support the evolution of a Semantic Web, as they would aid in capturing the meaning of interaction processes. With the advent of such technologies as Web services, application service provision, enterprise application integration and enterprise application cooperation, the importance of interaction in Web-based technologies is constantly increasing, and the ability to model these interaction processes has the potential to significantly contribute to a transparent and well-structured *interaction Web*.

7.5 Summary

This completes our account of actual and potential future applications of our method. The broad spectrum of different applications we have covered, ranging from very concrete ones that have already been implemented to more abstract ideas for envisioned uses of our approach demonstrates its wide applicability and underlines the importance of the research conducted on the subject.

The next, final chapter presents a summary of the entire thesis, an outlook on future work and some concluding remarks of a more general nature. Especially section on future work nicely contrasts the list of applications we have presented in this chapter, as it discusses the most important improvements that could be made to our methods, thereby ensuring that a critical view is not omitted altogether.

8. Conclusion

To round up, we are first going to provide a summary of our presentation so far and then to review our main results and contributions in this chapter. Then, an outlook on possible directions for future research on the subject will be given. The chapter is concluded with some more general closing remarks.

8.1 Thesis Summary

In chapter 1 (p. 3) it was claimed that the subject of this thesis is

> *The learning and strategic use of categories of interaction patterns by socially intelligent agents in open systems, in which no information is available about agents' future behaviour other than what is known from interaction experience.*

To contribute to this subject, we set out to develop an abstract social reasoning architecture for dealing with these categories of interaction patterns. We defined a formal model for a particular instance of this architecture together with suitable learning and decision-making algorithms, and evaluated it using an implementation that was validated in a complex application scenario.

In doing so, we also touched upon a variety of other topics, such as the sociological foundations of our approach, the theory of communication systems and empirical semantics, hierarchical reinforcement learning methods, the Web linkage domain, different approaches to negotiation, etc. In the light of the variety of issues dealt with, we should take a minute to look at the methodological implications of what was called the iterative "narrowing down" (p. 7) of the scope of this thesis from chapter to chapter.

Essentially, our main aim was to explore the different dimensions of computational interaction frames and their potential to be used to improve agents' strategic communication capabilities in open MAS. Obviously, this involves looking at

- the foundations of communication and its semantics,
- issues related to learning communication patterns, and
- the link between social context and its cognitive processing

to gain a full understanding of research issues that arise when trying to build adequate architectures and algorithms for dealing with communication.

At the bottom line, what ties all these different aspects together is the quest for methods that enable agents to *deal with the means of communication provided in a multiagent society*. In a very pragmatic sense, we could say that if the inter-agent communication layer is specified – as is common in the area of agent-based and multiagent systems – in terms of an agent communication language and a set of interaction protocols defined using this language, the methods we have developed enable agents to *reason about* and to *strategically use* this communication layer to their own benefit. Practically speaking, the agent designer who uses InFFrA/m^{2}InFFrA is required to perform the following tasks:

- To break down the protocols into meaningful communication and action sequences represented in a generalised fashion through the use of message patterns and to turn these sequences into frame trajectories.
- To build frame condition sets by combining the semantic rules (usually given in the form of pre- and postconditions) of speech acts that occur along a trajectory and to endow agents with rules for logical inference that will render them capable of reasoning about these conditions.
- To adapt the concrete InFFrA reasoning mechanism of agents appropriately to the constructed frames so that a goal-oriented (viz utility-maximising) use of the existing means of communication can be ensured.

It is by looking at our approach from this perspective that it reveals its true strengths. If the agent designer performs the above steps, what he gets "for free" by using InFFrA/m^{2}InFFrA is:

1. That agents will be capable of storing their interaction experience inside frames and to learn the long-term utility of each frame. This means that they can use the existing protocols optimally considering the actual *behaviour* of other agents (and themselves) towards the pre-specified protocols and ACL semantics.
2. That they will integrate (their own and others') *deviations* from the pre-specified normative rules of communication in their experience whenever they realise that frame trajectories or frame conditions have been violated. Moreover, they will be able to learn whether deviant behaviour itself is profitable in certain situations.
3. Agents will flexibly generalise over similar interaction experiences, thus managing their frame repositories under computationally scarce resources. Also, by virtue of "reasoning by similarity", they will be able to estimate the probability of achieving a communication goal even when faced with unknown situations.
4. Whenever possible, they will seek to re-combine existing trajectories and potentially discover new ways of coordination in this way or at least build meaningful "compound" frames if it makes sense to join previously separated message sequences together.

All this taken together nicely illustrates that the capabilities of frame-based agents clearly exceed those of agents built to simply apply existing interaction protocols and ACL semantics under the assumption that these will be adhered to by everyone in the system. And this underlines the importance of our approach for communication in *open* systems.

To illustrate the process of focusing on the aspects that have lead to this core functionality throughout our presentation in the previous chapters we depict it graphically in figure 8.1. The figure explains which elements of each chapter played an important role in

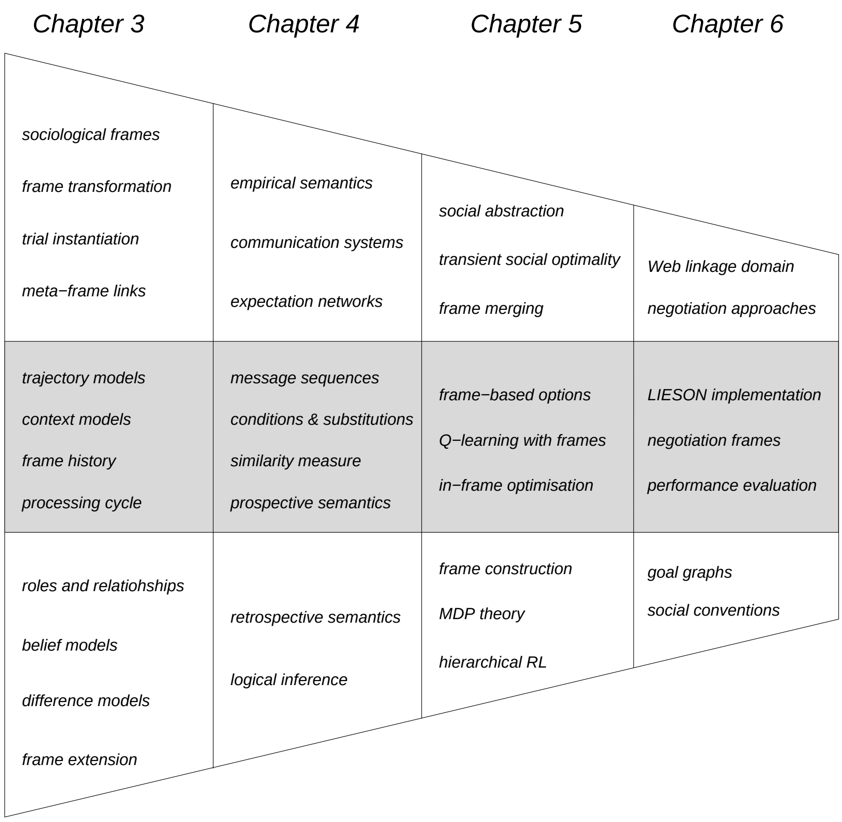


Fig. 8.1: Thematic structure of thesis chapters: The main contributions of each chapter are grouped together in the shaded box in the middle of the figure. Foundational aspects or concepts used to a lesser degree in subsequent chapters are arranged around the core issue area.

the respective next step of concretion and thus constitute the *core* of our work that resulted in the improved social reasoning functionality.

That said, it is a natural consequence of the pursued methodology that not all the possibilities for building frame-based systems which result from the theories and concepts we talked about have been exploited to the limit. The differences between what would have been possible in theory and what was actually realised are most obvious in two cases:

1. The pragmatic character of the formal model used to define m²InFFrA in comparison to the vast range of possibilities for specifying frames and framing procedures introduced in the abstract InFFrA architecture.

2. The focus on specific negotiation scenarios in the implementation of m²InFFrA within LIESON as compared to the many different existing coordination mechanisms and protocols that the implemented system could have been tested with.

As concerns 1., the process of designing a concrete architecture based on the abstract framework was primarily guided by the idea of designing m²InFFrA as a *minimal* model of InFFrA that still realises the most important features of the framework. It uses trajectory models that describe message pattern sequences $T(F)$ in a content-rich agent communication language $\mathcal{M}$, propositional frame conditions $C(F)$ that can be used to represent roles and relationships, contexts, and beliefs. Frame history is represented by past substitutions stored in $\Theta(F)$, substitution counters $h_\Theta(F)$, and by the Q-values that store a long-term average of the rewards obtained using a frame. Frame status is captured by the current substitution ϑ_{fixed} (p. 100). Links between frames exist by virtue of the framing Q-table. This table implicitly describes a preference ordering between different frames in different states. Also, links are realised through trajectory occurrence counters $h(F)$ that track frame matching across frames. Extension is not modelled explicitly, all agents are assumed to share the same set of frames. The history of frame modification is not explicitly stored inside a frame.

Regarding the InFFrA inference cycle, matching and interpretation takes place in the form of matching repository frame prefixes with the perceived frame trajectory. The difference model consists of a statement about whether the perceived frame matches a prefix of some repository frame or not. Framing assessment is based on validity and adequacy checking (prefix matching, computation of Θ_{poss}, proving context conditions), and on determining the desirability of the optimal substitution ϑ^* (p. 123). Frame selection and re-framing is realised through probabilistic Q-choice with $P(s, F)$ (p. 125), and frame adjustment through frame merging/frame construction (sections 5.2.3/5.2.2). Determining the optimal action m^* (p. 123) within a frame constitutes the frame enactment and behaviour generation strategy of m²InFFrA.

The way these processing steps are realised is heavily influenced by the principles of empirical communication semantics laid out in section 4.1: desirability heuristics depends on entropy measurements of the frame repository (or an expectation network-like interpretation of it), re-use of existing expectations is applied by concatenating frame trajectories in frame construction in a goal-directed way. Most importantly, the probabilistic (prospective) frame semantics that are based on computing the likelihood of continuation probabilities $P(w'|w)$ (p. 109) capture the very essence of empirical semantics.

It is important to understand that m²InFFrA represents an example of InFFrA that is *computationally tractable* and allows for applying decision-making and learning methods, and thus to transcend the level of merely specifying abstract interaction frames and framing procedures. And this is of course only possible by controlling some of the complexity of InFFrA.

As for 2., our choice to apply m²InFFrA to negotiation was guided by several considerations. First of all, negotiation is one of the most general, interesting and complex forms of interaction. Additionally, as a coordination mechanism, it is particularly suitable for *open* systems, because it focuses on *how to reach agreement in the presence of a conflict of interest.* In other words, to look at negotiation means to investigate the "worst case" of interaction, in which self-interested agents with divergent motives, preferences and goals have to either (i) opt for a compromise that will usually incur some cost to ensure the other's willingness to cooperate or (ii) resort to communicative conflict. Therefore, we have reason to believe that our results carry over to more cooperative settings with ease.

Also, negotiation (and, in particular, argumentation-based and interest-based negotiation) requires a close interplay between local reasoning about mental attitudes on the one hand, and social reasoning about communicative conventions and the adequacy of different negotiation strategies on the other. Thus, although some InFFrA aspects (such as roles and relationships) did not play a crucial role in our experimental validation, the combination of complex frame conditions with a variety of different negotiation moves provides more than sufficient complexity to illustrate the effectiveness of our approach.

To sum up, the concrete formal models and implementation we provided constitute a typical and simple but powerful instance of the abstract InFFrA framework. What is more, they illustrate its practical value in complex application scenarios. In the following paragraphs, we are going to discuss how they might be further improved considering the possibilities that the abstract framework and the theoretical foundations of our work offer.

8.2 Future Work

Despite our extensive treatment of computational interaction frames, of abstract and concrete architectures devised for dealing with them, and of a number of related issues, a variety of open problems remain to be investigated in the future.

In order to describe the most important and interesting of these issues at a concrete level, we are going to take a very practical view. This view consists of analysing the most important elements of what would have been possible by exploiting the full complexity of InFFrA and was not realised in the systems actually developed using the abstract architecture.[1]

8.2.1 Complex trajectory models

The most obvious shortcoming of our formal model of InFFrA is that it only allows for linear message pattern sequences as trajectory models. Although this does not render the capturing of complex interaction protocols impossible in principle, breaking down these protocols into the set of message sequences that they allow may result in a prohibitively large number of m^{2}InFFrA frames and does not appear an elegant solution.

Part of this problem can be alleviated by modularising frames such that their trajectories capture those parts of protocols that occur repeatedly (e.g. bidding rounds in auctions). These "protocol bits" can be linked with each other through appropriate frame conditions (such that certain paths can only occur when others have been completed) but this obviously violates central assumptions about the atomic character of frame execution. In particular, the probabilistic semantics of m^{2}InFFrA does not allow for projecting message sequences beyond completion of the current frame, and thus consequences of the conversation that would occur after (repeated) application of subsequent frames cannot be taken into consideration in the decision-making phase.

Yet the only adequate solution would be to extend the semantics of m^{2}InFFrA in such a way that they can deal with more complex trajectory models (e.g. finite-state machines or Petri nets). This would involve major modifications to the empirical semantics model since we have not developed a model for cyclic expectation networks yet. The obvious area that we would have to look into for models of such stateful stochastic processes is that of *dynamic Bayesian networks* (Nicholson and Brady 1994).

[1] Note that we deliberately restrict this discussion to future work on improving the formal framework and concrete algorithms proposed in previous chapters. An outlook on future *applications* has been given in chapter 7.

8.2.2 Condition construction and condition mining

An issue that is maybe less obvious is that of condition generation in new cases that are stored in existing frames. In the implementation described in chapter 6, we used a fairly simple mechanism to construct conditions for new cases. There, as described in section 6.2.1 (p. 167) we distinguished between "frame conditions" and "case conditions". The former represent general conditions that have to hold during each enactment of the frame, while the latter are inserted when a repository frame is extended by a new case and are thought to represent the circumstances of that specific instance. These case conditions need not hold in future enactments, but their validity is checked when computing the (similarity-based) likelihood of possible substitutions in the future.

The way in which these case conditions are constructed after a new encounter is rather simplistic: we generate pre- and postconditions (using the *can* and *effects* predicates) for each physical action that occurred along the perceived trajectory and include them in the condition set that corresponds to the new substitution. This ensures that the constraints which need to hold for future re-enactments of a previous case are respected, and that the consequences of physical actions are taken into account when estimating the utility of a ground trajectory.

The fact that all non-physical actions are ignored in this process of case condition construction has several implications:

- All frames that do not contain physical actions have empty case condition sets. In particular, if no frame conditions were supplied for these frames (which is always the case if the frames in question are not frames that the repository was initialised with at the beginning), *all* condition sets of the frames will be empty. So, in fact, no real contextualisation takes place (these frames can be used anytime and anywhere).

- This entails that, as no conditions have to be proven for the remaining actions on the trajectory suffix while using such a frame, *any* content can be inserted into remaining performatives. This is always the case if the running substitution leaves certain degrees of freedom with respect to content variables that occur in messages still to be uttered during the conversation.

  As an example, consider a variant of F_2 in table 6.4 (p. 164) in which a `reject` occurs instead of `do` so that the frame has no physical effects. If stored and used in later framing cycles, this frame will allow B to use any content in `propose`(B, A, Y), as Y is no more restricted to the set of physical actions that can currently be executed.

  This not only means that the set of possible substitutions Θ_{poss} becomes huge (or even infinite, as in the language $\mathcal{M}$ we use[2]) but also that the resulting conversations can be nonsensical.

All this insinuates that we should think about more complex ways of generating case conditions for new cases of existing frames and for completely new frames.

With this respect, one approach that we find particularly appealing is that of *condition mining*, i.e. determining which conditions are relevant from one's current beliefs using inductive learning methods.

[2] In practice, instead of generating all possible content symbols, agents simply use a single variable that stands for "arbitrary content" as the choice of content makes no difference utility-wise.

Roughly speaking, this approach is based on the following idea: at any point in time at which an encounter is completed, this marks a "success" case for either (i) an existing repository frame that was applied during the encounter or (ii) a completely new, previously unexpected message sequence. In case (ii), the new case can also be seen as a "failure" case for the repository frame that was originally intended (or frames, in the case of additional re-framing procedures during the encounter). For any such positive or negative sample, the agent disposes of a set of beliefs (in the form of knowledge base facts) that held while the respective encounter took place. This enables us to look for commonalities between the different situations in which a frame succeeded or failed, and with this we are able to learn decision rules by "mining" through the different knowledge base instances while using heuristic rules to prune all information that is completely irrelevant to the current encounter.

Eventually, the result of such inductive learning of conditions under which a frame is likely to work out and conditions under which it will probably fail would be that we can correlate those aspects of a general set of world beliefs to frame enactment that are relevant for it. Additionally, the distinctions that can be made using the results of this "condition learning" with respect to the applicability of frames under changing circumstances might be used to define powerful encounter state abstractions (see section 5.3.2) in a bottom-up fashion.

8.2.3 Frame modification operators

In previous chapters, we described two operations that transform entire frames:

1. concatenation of frame trajectories that enables agents to generate new frames on the fly and to utilise them in a planning sense described in section 5.2.2, and
2. frame merging and generalisation (section 5.2.3) which is useful to ensure the manageability of frame repositories and to evolve useful abstractions of concrete encounters.

Both these methods demonstrate how existing frames can be combined to yield new ones. Quite naturally, many other useful operations on frames can be thought of, as has been suggested in section 3.3.2 on frame history (p. 60).

Without going into details, we list some of the operators that might be defined to obtain more complex methods of automated frame construction and which appear most interesting:

- *Frame splitting*: If different parts of a single frame are identified that would make sense to be executed independently, it should be possible to split one frame into several sub-frames. This would be particularly useful if used in combination with the "join" method we have proposed for frame concatenation, as it would allow for a modular re-combination of all meaningful constituents of existing frames.
- *Frame consolidation*: After many enactments, the frames that evolve during m^{2}InFFrA simulations may contain a large number of very similar substitutions and conditions. To prevent the generation of almost trivial (but huge) collections of frame attributes for particular instances, a large number of past cases could be coerced into a smaller list by omitting details that do not seem to affect the applicability of the frame both in terms of conditions as in terms of substitutions.

- *Frame pruning*: Since frames are often supposed to contain a normative picture of what certain types of interaction should be like, it seems desirable to see to their internal consistency. For this purpose, it might be useful or even necessary to prune exceptional cases under which a frame occurred (but which do not really fit into the semantic category of interaction processes that the frame is thought to represent) from that frame.

This is by no means an exhaustive list of useful frame modification operators, and eventually it will depend on the respective application which of them will be reasonable to implement.

8.2.4 Evolution of language and frame dissemination

A central characteristic of the formal model and implementation of InFFrA that has been developed in this thesis is that it (only) enables agents to *apply* and *re-combine* existing interaction patterns, but that it does not provide methods to generate completely new forms of interaction.

Actually, the question of how to conceive of new methods of communication boils down to asking how new communicative *symbols* can emerge in a society. This is because, for existing symbols, it suffices to modify existing frames to construct new communicative patterns. Considering that the capacity of humans to invent and spread new symbols in communication is of crucial importance to the evolution of communication and language in a dynamically evolving society (we have already touched upon this issue in sections 2.3.2 (p. 35) and 4.1.4 (p. 85)), it seems reasonable to ask how this process might be realised in frame-based systems and whether it would be useful for such systems.

From a pragmatic perspective, what we would have to do to enable the introduction of new symbols would be:

- *To specify when new symbols are needed.*
  If we follow common-sense intuition and consider the theory of communication put forward in chapter 4.1, new symbols are necessary whenever an agent is not able to express his expectations using existing symbols. From a frame-based point of view, this can happen whenever
  - an agent wants to deviate from an existing pattern but would like to establish the deviant pattern as a possible procedure that can be re-used[3] in the future;
  - an agent has constructed a new frame that should be re-used in the future and would like to dispose of a distinct symbol to indicate use of that frame;
  - an agent wants to indicate to his adversary that expectations regarding the continuation of the current conversation are invalid and that a new communication path will be followed.

  That is, a new symbol is necessary whenever no symbols are available which capture the meaning (which has been defined as a set of mutual expectations in our framework) of the current situation and the agent wants to convey this meaning to others he is interacting with.

[3] If it is only a one-time rejection, there is no need for capturing the associated expectations or generating a symbolic "encoding" for them.

- *How to convey the meaning of these symbols.*
  A "chicken-and-egg" relationship of mutual dependence between symbols and expectations creates major problems when trying to introduce a new symbol in a social system. On the one hand, whoever invents the new symbol would like others to understand its meaning, i.e. to be aware of the expectations associated with it. On the other, the new symbol is being introduced precisely for the reason that no other known symbol captures the expectation the agent wants to express.

  Although there is no solution to the general problem, there are ways to convey the meaning of new symbols, at least if an appropriate set of commonly accepted symbols already exists for which agents have shared expectations:

  – *Deictic demonstration* is a method by which agents explain what they mean by enacting the desired consequences themselves. This is very common among humans, for example when pointing at an object and saying the word that stands for it to explain something to someone who does not speak the same language. However, it is quite complex to apply this method to expectations regarding *actions*, as this may necessitate imitation of what the other party is supposed to do upon hearing the new utterance.

    In frame-based agents, this would would require defining specific interaction frames within which such demonstrations can take place in an organised fashion.

  – *Reasoning by analogy* can be used if the expectations associated with a new symbol only differ in some details from existing ones, and the agent can use a symbol that bears some similarity to other existing symbols. This gives other parties the opportunity to behave as before, while the agent who created the new symbol can provide feedback on those parts of the expectation he has for the new symbol that are not fulfilled by others' reactions.

    Of course, this is a very fragile process in which misunderstandings can occur in many situations. Moreover, it requires a mutually accepted frame-based apparatus for making corrections whenever the other agent is not behaving according to the intended meaning of the new symbol.

  – *Talking about expectations themselves*, finally, is the most generic and most powerful method of conveying meaning. Although this requires frames for exchanging information about expectations themselves (e.g. by including representations of expectation networks in message contents), it renders an agent capable of making the expectations he has explicit. We will return to this issue of *meta-frame communication* in section 8.2.5

- *How to support the establishment of their meaning.*
  Even if other agents get to understand what is meant by a new type of message this does by no means guarantee that usage of it will spread. Some agents who hear the newly uttered symbol for the first time may find it useful and use it again when talking to other interaction partners, others may not need it and never use it again.

  Although it is quite reasonable to think of this process as "survival of the fittest symbols", it may be necessary to endow agents with a bias to use new symbols until they can properly assess their usefulness. This is because, initially, agents' opinions regarding the new symbol will be based on a singular experience that may not be representative for the average long-term usefulness of a symbol. Thus, there is a need for appropriate *symbol exploration strategies* if new symbols are not to fall into undeserved oblivion all too soon.

Despite the fact that many problems can arise in this process, letting new symbols emerge (and with them, totally new forms of interaction and coordination) would enable an agent society to develop the language that is best for solving its interaction problems. Apart from the fact that this might take some weight off designers' shoulders since they would not have to think of all necessary language constructs and frames *a priori*, it would also result in *communication self-design* according to the needs of a MAS so that, eventually, all necessary means of communication can be derived on the grounds of basic agent rationality.

8.2.5 Meta-frame communication

In the previous section it was claimed that being able to talk about one's expectations can aid the process of introducing new symbols to an existing communication system. *Meta-frame communication* in which frames themselves become the subject of conversation can also be used for many other purposes. In fact, it is a very powerful means of coordination that would strongly add to the social abilities of InFFrA agents.

Let us briefly explain what we mean by meta-frame communication by giving a few scenarios in which it would solve coordination problems:

- During frame-based communication, it is often the case that the flow of communication is blocked because either (or both) agents do not dispose of appropriate frames to understand what is going on. This happens when the other party has done something previously unexpected. In this case, it would be very reasonable to step back from the *first-order* framing process and to ask the other what his expectation was when he uttered the unexpected message. This would introduce a *second-order* framing procedure whose outcomes are not ordinary action consequences, but modifications to the first-order frame(s) applied before.

- Upon examination of his own frame repository, an agent may find that it would make sense to perform some modifications to it. For example, a frame might be redundant because it is never used, a set of frames should be re-combined in a different fashion to produce more effective coordination mechanisms, or some joint action combinations occur so frequently that short and simple frames should be available to spawn these joint actions right away.

  In all of these cases, the agent might decide to modify his repository, but it is rather doubtful that others will have performed the same modifications (silently) and, with this, it is quite probable that new framing problems will occur. Being able to inform others of frame adjustments or to negotiate such adjustments can obviously help to solve this problem.

Clearly, devising mechanisms for meta-frame communication necessitates the design of *meta-frames* that allow for exchanging information about frame conceptions between different agents, for negotiating new frames, etc. Also, it requires changing the utility model, as agents should pursue goals during communication about frames that are beyond the goals the agent has in the physical environment.

Although we have not worked out the details of how to extend m^{2}InFFrA in this direction, this certainly constitutes one of the most exciting and promising future research directions that could be followed. It is our opinion that meta-communication is a mechanism that has been largely overlooked especially for *conflict resolutions* purposes, where it enables agents to make their expectations explicit in cases in which these expectations differ a lot among interacting parties.

8.3 Closing Remarks

The suggested improvements to the methods we have proposed in this thesis illustrate what the possible directions of future research on the subject of computational interaction frames and frame-based social reasoning architectures are. Taken together with our results of chapter 6 and the applications of InFFrA described in chapter 7 this yields an impressive picture of the potential of our approach for interaction management in open multiagent systems.

Yet this constitutes only a small step in the endeavour of building agent-based systems capable of successful operation in open environments. The challenges posed by highly complex, dynamic and open applications in areas such as the Semantic Web, context-aware computing, knowledge management, etc. are manifold. Developing methods that aid in managing agent interactions is only one of them – albeit a very important one, in our view.

In this thesis, we have explored a new concept for reasoning about interaction at the *micro* level, i.e. at the level of "face-to-face" conversations between single agents with a particular focus on the interplay between these micro-social interactions and their cognitive processing by agents who participate in them. Using socio-theoretical foundations, we developed a semi-formal notion of the concept of "interaction frames" which was later formalised, implemented, and empirically validated. The essence of our efforts is that interaction frames and framing-based architectures can be successfully employed to model, analyse and manage agent-to-agent interactions in a wide range of complex applications.

What remains to be investigated in the future is how this approach can be combined with the macro-level of social processes in order to gain a full understanding of how to build multiagent systems in such a way that they can survive (and thrive) in open environments.

Bibliography

Aamodt, A. and Plaza, E. (1994). Case-based Reasoning: Foundational Issues, Methodological Variations, and System Approaches, *AI Communications* 7(1): 39–59.

Alonso, E. (1998). How Individuals Negotiate Societies, *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98)*, Paris, France, IEEE Computer Society Press, pp. 18–25.

Alonso, E. (1999). An individualistic approach to social action in Multi-Agent Systems, *Journal of Experimental and Theoretical Artificial Intelligence* 11: 519–530.

Anderson, J. R. and Lebiere, C. (1998). *The atomic components of thought*, Lawrence Erlbaum Associates, Mahwah, NJ.

Austin, J. L. (1962). *How to do things with Words*, Clarendon Press.

Balzer, W. and Tuomela, R. (2001). Social Institutions, Norms, and Practices, *in* R. Conte and C. Dellarocas (eds.), *Social Order in Multiagent Systems*, Kluwer Academic Publishers, Norwell, MA, Amsterdam, The Netherlands, pp. 161–180.

Barto, A. and Mahadevan, S. (2003). Recent Advances in Hierarchical Reinforcement Learning, *Special Issue on Reinforcement Learning, Discrete Event Systems* 13: 41–77.

Bauer, B., Müller, J. and Odell, J. (2000). An extension of UML by protocols for multiagent interaction, *Proceedings of the 4th International Conference on Multi-Agent Systems (ICMAS-2000)*, pp. 207–214.

Behrens, C. and Kashyap, V. (2002). The "Emergent" Semantic Web: A consensus approach for deriving Semantic Knowledge on the Web, *Real World Semantic Web Applications*, Frontiers in Artificial Intelligence and Applications, vol. 92, IOS Press.

Gerson, E. M., Henderson, A., Hewitt, C., Scacchi, W., Star, S. L., Suchman, L. and Trigg, R. (1988). The Unnamable: A White Paper on Socio-Computational 'Systems'. Unpublished draft manuscript.

Berners-Lee, T., Hendler, J. and Lassila, O. (2001). The Semantic Web, *Scientific American*.

Blumer, H. (1962). Society as1 Symbolic Interaction, *in* A. M. Rose (ed.), *Human Behavior and Social Process*, Routledge and Kegan Paul, London.

Blumer, H. (1966). Sociological implications of the thought of George Herbert Mead, *American Journal of Sociology* 71: 535–548.

Blumer, H. (1969). *Symbolic Interaction*, Prentice-Hall, Englewood Cliffs, NJ.

Blumer, H. (1986). *Symbolic Interactionism: Perspective and Method*, University of California Press, Berkeley, CA.

Bond, A. and Gasser, L. (1988a). An analysis of problems and research in DAI, *in* A. Bond and L. Gasser (eds.), *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, pp. 3–35.

Bond, A. and Gasser, L. (eds.) (1988b). *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA.

Boutilier, C. (1999). Sequential Optimality and Coordination in Multiagent Systems., *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden.

Bradtke, S. J. and Duff, M. O. (1995). Reinforcement Learning Methods for Continuous-Time Markov Decision Problems, *in* G. Tesauro, D. Touretzky and T. Leen (eds.), *Advances in Neural Information Processing Systems (NIPS-95)*, vol. 7, The MIT Press, Cambridge, MA.

Bratman, M. (1987). *Intentions, Plans and Practical Reason*, Harvard University Press, Cambridge, MA.

Bratman, M. E., Israel, D. J. and Pollack, M. E. (1988). Plans and resource-bounded practical reasoning, *Computational Intelligence* 4(4): 349–355.

Brauer, W., Nickles, M., Rovatsos, M., Weiß, G. and Lorentzen, K. F. (2001). Expectation-Oriented Analysis and Design, *Proceedings of the 2nd Workshop on Agent-Oriented Software Engineering (AOSE-2001) at the Autonomous Agents 2001 Conference*, Lecture Notes in Artificial Intelligence, vol. 2222, Springer-Verlag, Berlin et al.

Bui, H., Kieronska, D. and Venkatesh, S. (1996). Learning other agents' preferences in multiagent negotiation, *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, AAAI Press, Menlo Park, CA, pp. 114–119.

Burke, T. (1995). Dance Floor Blues: The Case for a Social AI, *Stanford Humanities Review* 4(2): 10–20.

Burmeister, B., Haddadi, A. and Sundermeyer, K. (1995). Generic configurable cooperation protocols for multi-agent systems, *in* C. Castelfranchi and J.-P. Müller (eds.), *From Reaction to Cognition. Proceedings of the Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-93)*, Lecture Notes in Artificial Intelligence, vol. 957, Springer-Verlag, Berlin et al.

Campbell, J. (1981). George Herbert Mead on Intelligent Social Reconstruction, *Symbolic Interaction* 4(2): 191–205.

Carbonell, J. G., Knoblock, C. A. and Minton, S. (1989). PRODIGY: An integrated architecture for planning and learning, *Technical Report CMU-CS-89-189*, Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

Carley, K. M. and Gasser, L. (1999). Computational organization theory, *in* G. Weiß (ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, MA, pp. 299–330.

Carmel, D. and Markovitch, S. (1996). Learning and using opponent models in adversary search, *Technical Report 9609*, Technion, Haifa, Israel.

Carmel, D. and Markovitch, S. (1998). Model-based learning of interaction strategies in multi-agent systems, *Journal of Experimental and Theoretical Artificial Intelligence* 10(3): 309–332.

Castelfranchi, C. (2000). Engineering social order, *Working Notes of the First International Workshop on Engineering Societies in the Agents' World (ESAW-00)*.

Castelfranchi, C., Dignum, F., Jonker, C. M. and Treur, J. (1999). Deliberate normative agents: Principles and architecture, *Proceedings of the Sixth International Workshop on Agent Theories, Architectures, and Languages (ATAL-99)*, Orlando, FL.

Chaib-draa, B. and Dignum, F. (2002). Trends in Agent Communication Language, *Computational Intelligence* 18(2): 89–101.

Chicoisne, G. and Pesty, S. (1999). Modèle de conversation et agents rationnels socialment corrects, *Atelier Thématique TALN*, Cargèse, France.

Ciancarini, P. and Wooldridge, M. J. (2001). *Agent-Oriented Software Engineering. First International Workshop, AOSE-2000, Limerick, Ireland, June 10, 2000*, Lecture Notes in Computer Science, vol. 1957, Springer-Verlag, Berlin et al.

Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems, *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, pp. 764–752.

Cohen, P. R. and Levesque, H. J. (1990a). Intention is choice with commitment, *Artificial Intelligence* 42: 213–261.

Cohen, P. R. and Levesque, H. J. (1990b). Performatives in a Rationally Based Speech Act Theory, *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pp. 79–88.

Cohen, P. R. and Levesque, H. J. (1991). Teamwork, *Noûs* 35: 487–512.

Cohen, P. R. and Levesque, H. J. (1995). Communicative actions for artificial agents, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 65–72.

Cohen, P. R. and Perrault, C. R. (1979). Elements of a Plan-Based Theory of Speech Acts, *Cognitive Science* 3: 177–212.

Conte, R. and Castelfranchi, C. (1996). From conventions to prescriptions: Toward an integrated theory of norms, *Proceedings of the ModelAge-96 Workshop*, Sesimbra, Italy.

Conte, R. and Dellarocas, C. (eds.) (2001). *Social Order in Multiagent Systems*, Kluwer Academic Publishers, Norwell, MA, Amsterdam, The Netherlands.

Craig, I. D. (1994). Agents That Model Themselves, *Technical Report CS-RR-266*, Department of Computer Science, The University of Warwick, Coventry, UK.

Crites, R. and Barto, A. (1996). Improving elevator performances using reinforcement larning, *in* D. Touretzky, M. Mozer and M. Hasselmo (eds.), *Advances in Neural Information Processing Systems 8*, The MIT Press, Cambridge, MA.

Dastani, M., van der Ham, J. and Dignum, F. (2002). Communication for Goal Directed Agents, *Proceedings of the Agent Communication Languages and Conversation Policies AAMAS-02 Workshop*, Bologna, Italy.

Dautenhahn, K. (ed.) (2000). *Human Cognition and Social Agent Technology*, Advances in Consciousness Research, vol. 19, John Benjamins Publishing Company, Amsterdam, The Netherlands.

Davidsson, P. (2001). Categories of Artificial Societies, *in* A. Omicini, P. Petta and R. Tolksdorf (eds.), *Engineering Societies in the Agents World II. Second International Workshop, ESAW 2001, Prague, Czech Republic, July 7, 2001*, Lecture Notes in Artificial Intelligence, vol. 2203, Springer-Verlag, Berlin et al.

Decker, K. S. (1987). Distributed problem solving techniques: A survey, *IEEE Transactions on Systems, Man, and Cybernetics* 17(5): 729–740.

Dellarocas, C. and Klein, M. (2001). Contractual Agent Societies, *in* R. Conte and C. Dellarocas (eds.), *Social Order in Multiagent Systems*, Kluwer Academic Publishers, Norwell, MA, Amsterdam, The Netherlands, pp. 161–180.

Demazeau, Y. and Werner, E. (eds.) (1992). *Decentralized A.I. 3. Proceedings of the Third European Workshop on Modelling Autonomous Agents in a Multi-Agent World, Kaiserlauten, Germany, August 5–7, 1991*, Elsevier Science Publishers, B.V., Amsterdam, The Netherlands.

Demazeau, Y., Müller, J. P. and Muller, J.-P. (eds.) (1990). *Decentralized A.I. Proceedings of the European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-89), Cambridge, UK, August 16–18, 1989*, Elsevier Science Publishers, B.V., Amsterdam, The Netherlands.

Demazeau, Y., Müller, J. P. and Muller, J.-P. (eds.) (1991). *Decentralized A.I. 2*, Elsevier Science Publishers, B.V., Amsterdam, The Netherlands.

Dignum, F. and Greaves, M. (eds.) (2000). *Issues in Agent Communication*, Lecture Notes in Artificial Intelligence, vol. 1916, Springer-Verlag, Berlin et al.

Dignum, F. and van Linder, B. (2002). Modeling Social Agents: Towards deliberate communication, *in* J. J.-C. Meyer and J. Treur (eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 7, Kluwer Academic Publishers, Norwell, MA, Amsterdam, The Netherlands, pp. 357–380.

Dignum, F., Morley, D., Sonenberg, E. and Cavedon, L. (2000). Towards Socially Sophisticated BDI Agents, *in* E. H. Durfee (ed.), *Proceedings of the Fifth International Conference on Multi-Agent Systems (ICMAS-00)*, IEEE Computer Society Press, Boston, MA, pp. 111–118.

Durfee, E. and Lesser, V. (1991). Partial global planning: A coordination framework for distributed hypothesis formation, *IEEE Transactions on Systems, Man, and Cybernetics* 21(5): 1167–1183.

Durfee, E. H. (1999). Distributed Problem Solving and Planning, *in* G. Weiß (ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, MA, chapter 3, pp. 121–164.

Durfee, E., Lesser, V. and Corkill, D. (1992). Distributed problem solving, *in* S. Shapiro (ed.), *Encyclopedia of Artificial Intelligence*, John Wiley & Sons, New York, NY, pp. 379–388.

Fagin, R., Halpern, J. Y., Moses, Y. and Vardi, M. Y. (1995). *Reasoning about Knowledge*, The MIT Press, Cambridge, MA.

Fallah-Seghrouchni, A., Haddad, S. and Mazouzi, H. (1999). Protocol engineering for multiagent interaction, *in* F. Garijo and M. Boman (eds.), *Multi-Agent System Engineering. Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-99)*, Lecture Notes in Artificial Intelligence, vol. 1647, Springer-Verlag, pp. 89–101.

Ferguson, I. (1992). Towards an architecture for adaptive, rational, mobile agents, *in* E. Werner and Y. Demazeau (eds.), *Proceedings of the Third European Workshop on Modelling Automonous Agents in a Multi-Agent World (MAAMAW-91)*, Elsevier Science Publishers, B.V., Amsterdam, The Netherlands, pp. 249–262.

Ferguson, I. (1995). Integrated Control and Coordinated Behavior: A Case for Agent Models, *in* M. J. Wooldridge and N. Jennings (eds.), *Intelligent Agents*, Lecture Notes in Artificial Intelligence, vol. 890, Springer-Verlag, Berlin et al., pp. 203–218.

Finin, T., Labrou, Y. and Mayfield, J. (1997). KQML as an agent communication language, *in* J. Bradshaw (ed.), *Software Agents*, AAAI Press/The MIT Press, pp. 291–316.

FIPA (1999a). Extending UML for the specification of agent interaction protocols. OMG Document ad/99-12-03. FIPA (Foundation for Intelligent Agents, http://www.fipa.org).

FIPA (1999b). Foundation for Intelligent Agents, http://www.fipa.org.

Fischer, F. (2003). *Frame-Based Learning and Generalisation for Multiagent Communication*, Diploma Thesis, Department of Informatics, Technical University of Munich, Munich, Germany.

Fischer, K. and Florian, M. (eds.) (2003). *Socionics: Its Contributions to the Scalability of Complex Social Systems*, Lecture Notes in Computer Science, Springer-Verlag, Berlin et al. To appear.

Fischer, K., Ruß, C. and Vierke, G. (1998). Decision Theory and Coordination in Multiagent Systems, *Research Report RR-98-02*, DFKI, Saarbrücken/Kaiserslautern, Germany.

Foner, L. N. (1993). What's an Agent, Anyway? a Sociological Case Study, *Agents Memo 93-01*, Agents Group, MIT Media Lab, Cambridge, MA.

Fornara, N. and Colombetti, M. (2002). Operational Specification of a Commitment-Based Agent Communication Language, *in* M. Gini, T. Ishida, C. Castelfranchi and W. L. Johnson (eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, Bologna, Italy, ACM Press, pp. 536–542.

Franklin, S. and Graesser, A. (1997). Is it an agent, or just a program?: A taxonomy for autonomous agents, *in* J. P. Müller, M. J. Wooldridge and N. R. Jennings (eds.), *Intelligent Agents III*, Lecture Notes in Artificial in Artificial Intelligence, vol. 1193, Springer-Verlag, Berlin et al., pp. 21–36.

Freund, Y., Kearns, M., Mansour, Y., Ron, D., Rubinfeld, R. and Shapire, R. E. (1995). Efficient Algorithms for Learning to Play Repeated Games Against Computationally Bounded Adversaries, *36th Annual Symposium on Foundations of Computer Science (FOCS-95)*, IEEE Computer Society Press, Los Alamitos, CA, pp. 332–343.

Fudenberg, D. and Tirole, J. (1991). *Game Theory*, The MIT Press, Cambridge, MA.

Gasser, L. (1991). Social conceptions of knowledge and action: DAI foundations and open systems semantics, *Artificial Intelligence* 47: 107–138.

Gasser, L. and Huhns, M. (1989). Themes in distributed artificial intelligence research, *in* L. Gasser and M. Huhns (eds.), *Distributed Artificial Intelligence*, vol. 2, Morgan Kaufmann, San Francisco, CA, pp. vii–xv.

Gasser, L., Braganza, C. and Hermann, N. (1987a). Implementing distributed AI systems using MACE, *Proceedings of the 3rd IEEE Conference on Artificial Intelligence Applications*, Orlando, FL, pp. 315–320.

Gasser, L., Braganza, C. and Hermann, N. (1987b). MACE: A flexible testbed for distributed AI research, *in* M. Huhns (ed.), *Distributed Artificial Intelligence*, Pitman, London, pp. 119–152.

Georgeff, M. and Lansky, A. (1987). Reactive reasoning and planning, *Proceedings of the 6th National Conference on Artificial Intelligence (AAAI-87)*, pp. 677–682.

Georgeff, M. and Rao, A. (1995). BDI agents: From theory to practice, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 312–319.

Giunchiglia, F., Odell, J. and Weiss, G. (eds.) (2003). *Agent-Oriented Software Engineering III. Third International Workshop, AOSE-2002, Bologna, Italy, July 15, 2002*, Lecture Notes in Computer Science, vol. 2585, Springer-Verlag, Berlin et al.

Gmytrasiewicz, P. J. (2002). Negotation as a Mechanism for Language Evolution, *in* M. Gini, T. Ishida, C. Castelfranchi and W. L. Johnson (eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, Bologna, Italy, ACM Press, pp. 559–560.

Gmytrasiewicz, P. J., Summers, M. and Gopal, D. (2002). Toward Automated Evolution of Agent Communication Languages, *Proceedings of the 35th Hawaii International Conference on System Sciences (HICSS-35)*.

Goffman, E. (1959). *The Presentation of Self in Everyday Life*, Doubleday, Garden City, NY.

Goffman, E. (1974). *Frame Analysis: An Essay on the Organisation of Experience*, Harper and Row, New York, NY. Reprinted 1990 by Northeastern University Press.

Goffman, E. (1983). The Interaction Order, *American Sociological Review* 48: 1–17.

Guerin, F. and Pitt, J. (2001). Denotational Semantics for Agent Communication Languages, *Proceedings of the Fifth International Conference on Autonomous Agents (Agents-01)*, ACM Press, pp. 497–504.

Gutknecht, O. and Ferber, J. (1998). A model for social structures in multi-agent systems, *Technical Report R.R.LIRMM 98040*, Laboratoire d'Informatique, de Robotique et de Microélectronique de Montpellier, Université Montpellier II, Montpellier, France.

Hewitt, C. (1986). Offices are open systems, *ACM Transactions on Office Information Systems* 4(3): 271–287.

Hewitt, C. (1991). Open information sytems semantics for distributed artificial intelligence, *Artificial Intelligence* 47: 79–106.

Hitzler, R. (1992). Der Goffmensch: Überlegungen zu einer dramatologischen Anthropologie, *Soziale Welt* 43: 449–461.

Hogg, L. and Jennings, N. (1997). Socially rational agents, *Proceedings of the AAAI Fall Symposium on Socially Intelligent Agents*, pp. 61–63.

Hu, J. and Wellman, M. P. (1998). Multiagent Reinforcement Learning: Theoretical Framework and an Algorithm, *Proceedings of the 15th International Conference on Machine Learning (ICML-98)*, pp. 242–250.

Huhns, M. and Singh, M. (1998a). Agents and multiagent systems: Themes, approaches, and challenges, *in* M. Huhns and M. Singh (eds.), *Readings in Agents*, Morgan Kaufmann, San Francisco, CA, pp. 1–23.

Huhns, M. and Singh, M. (eds.) (1998b). *Readings in Agents*, Morgan Kaufmann, San Francisco, CA.

Huhns, M. (ed.) (1987). *Distributed Artificial Intelligence*, Pitman/Morgan Kaufmann, San Fransicsco, CA.

Huhns, M. N. (2000). Interaction-Oriented Programming, *Agent-Oriented Software Engineering: first international workshop (AOSE-2000)*, Lecture Notes in Artificial Intelligence, vol. 1957, Springer-Verlag, Berlin et al.

JADE (2002). Java Agent Development Framework, http://jade.cselt.it.

Jain, A. K. and Dubes, R. C. (1988). *Algorithms for clustering data*, Prentice-Hall, Upper Saddle River, NJ.

Jennings, N. R., Faratin, P., Lomuscio, A. R., Parsons, S., Sierra, C. and Wooldridge, M. (2001). Automated Negotiation: Prospects, Methods and Challenges, *Journal of Group Decision and Negotiation* 10(2): 199–215.

Jennings, N. R., Parsons, S., Noriega, P. and Sierra, C. (1998a). On Argumentation-Based Negotiation, *Proceedings of the International Workshop on Multi-Agent Systems*, Boston,MA.

Jennings, N. R., Sycara, K. and Wooldridge, M. J. (1998b). A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems* 1: 7–38.

Joas, H. (1997). *The Creativity of Action*, The University of Chicago Press, Chicago, IL.

Jones, F. A. A. S. A. J. I. and Carmo, J. M. C. L. M. (1997). Action Concepts for Describing Organised Interaction, *in* R. A. Sprague (ed.), *Proceedings of the Thirtieth Annual Hawaii International Conference on System Sciences (HICSS-30)*, pp. 373–382.

Jung, C. G. and Fischer, K. (1998). A Layered Agent Calculus with Concurrent, Continuous Processes, *in* M. P. Singh, A. S. Rao and M. J. Wooldridge (eds.), *Intelligent Agents IV. Proceedings of the 4th International Workshop on Agent Theories, Architectures, and Languages (ATAL-97)*, Lecture Notes in Artificial Intelligence, vol. 1365, Springer-Verlag, Berlin et al.

Kaelbling, L., Littman, M. L. and Moore, A. W. (1996). Reinforcement learning: A survey, *Journal of AI Research* 4: 237–285.

Kendall, E. (1998). Agent roles and role models: New abstractions for multiagent system analysis and design, *International Workshop on Intelligent Agents in Information and Process Management.*

Kolodner, J. L. (1993). *Case-Based Reasoning*, Morgan Kaufmann, San Francisco.

Kone, M., Shimazu, A. and Nakajima, T. (2000). The state of the art in agent communication languages, *Knowledge and Information Systems* 2: 259–284.

Koning, J. L., Francois, G. and Demazeau, Y. (1998). An approach for designing negotiation protocols in a multi-agent system, *Proceedings IFIP-98*, Vienna/Budapest, pp. 333–346.

Kornfeld, W. and Hewitt, C. (1981). The Scientific Community Metaphor, *IEEE Transactions on Systems, Man and Cybernetics* 11(1): 24–33.

Kumar, S., Huber, M. J., Cohen, P. R. and McGee, D. R. (2002). Toward a Formalism for Conversation Protocols using Joint Intention Theory, *Computational Intelligence* 18(2): 174–228.

Kuwabara, D., Ishida, T. and Osato, N. (1995). AgentTalk: Coordination protocol description for multiagent systems, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, p. 455.

Labrou, Y. and Finin, T. (1997). A Proposal for a new KQML Specification, *Technical Report TR CS-97-03,* Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, Baltimore, MD.

Labrou, Y., Finin, T. and Peng, Y. (1999). Agent communication languages: the current landscape, *IEEE Intelligent Systems*, pp. 45–52.

Laird, J. E., Newell, A. and Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence, *Artificial Intelligence* 33: 1–64.

Langley, P. and Laird, J. E. (2002). Cognitive architectures: Research issues and challenges, *Technical report*, Institute for the Study of Learning and Expertise, Palo Alto, CA.

Lind, J. (2001). *Iterative Software Engineering for Multiagent Systems: The MASSIVE Method*, Lecture Notes in Artificial Intelligence, vol. 1994, Springer-Verlag, Berlin et al.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning, *Proceedings of the Eleventh International Conferencene on Multiagent Learning (ICML-94)*, New Brunswick, NJ, pp. 157–163.

Lorentzen, K. F. and Nickles, M. (2001). Ordnung aus Chaos – Prolegomena zu einer Luhmann'schen Modellierung deentropisierender Strukturbildung in Multiagentensystemen, *in* T. Kron, K. Junge and S. Papendick (eds.), *Luhmann modelliert. Ansätze zur Simulation von Kommunikationssystemen.*, Leske & Budrich.

Luce, R. D. and Raiffa, H. (1957). *Games and Decisions*, John Wiley & Sons, New York, NY.

Luhmann, N. (1984). *Soziale Systeme. Grundriß einer allgemeinen Theorie*, Suhrkamp, Frankfurt am Main, Germany.

Luhmann, N. (1995). *Social Systems*, Stanford University Press, Palo Alto, CA. Translated by J. Bednarz, Jr. and D. Baecker (originally published in 1984).

Maedche, A. and Staab, S. (2001). Learning Ontologies for the Semantic Web, *Proceedings of the Second International Workshop on the Semantic Web – SemWeb-2001, Hongkong, May 1, 2001*, pp. 51–60.

Maines, D. R. (1982). In Search of Mesostructure. Studies in the Negotiated Order, *Urban Life* 11(3): 267–279.

Malone, T. and Crowston, K. (1994). The interdisciplinary study of coordination, *ACM Computing Surveys* 26(1): 87–119.

Malsch, T. (2000). Naming the unnamable: Socionics or the sociological turn of/to distributed artificial intelligence, *Research Report RR-2*, Department of Technology Assessment, Technical University of Hamburg.

Malsch, T. (2001). Naming the Unnamable: Socionics or the Sociological Turn of/to Distributed Artificial Intelligence, *Autonomous Agents and Multi-Agent Systems* 4(3): 155–186.

Malsch, T. and Weiß, G. (2000). Conflicts in social theory and multiagent systems: On importing sociological insights into distributed artificial intelligence, *in* C. Tessier, L. Chaudron and H.-J. Müller (eds.), *Conflicting agents – Conflict management in multi-agent systems*, Kluwer Academic Publishers, Norwell, MA, Amsterdam, The Netherlands, chapter 4, pp. 111–149.

Malsch, T. (ed.) (1998). *Sozionik – Soziologische Ansichten über künstliche Sozialität*, Edition Sigma, Berlin.

Malsch, T., Lorentzen, K. and Paetow, K. (2000). The scalability paradox of multiagent systems, Internal working document, TU Hamburg-Harburg.

Malsch, T., Müller, J.-H. and Schulz-Schaeffer, I. (1998). Socionics: Introduction and Potential, *Journal of Artificial Societies and Social Simulation*.

Malsch, T., Paetow, K. and Rovatsos, M. (2002). Linkage liaisons – a scenario for the computational study of conflict and structure in multiagent systems, *Research Report RR-6*, Department of Technology Assessment, Technical University of Hamburg, Hamburg.

McBurney, P., Parsons, S. and Wooldridge, M. (2002). Desiderata for agent argumentation protocols, *in* M. Gini, T. Ishida, C. Castelfranchi and W. L. Johnson (eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, ACM Press, Bologna, Italy, pp. 394–401.

Mead, G. H. (1934). *Mind, Self, and Society*, University of Chicago Press, Chicago, IL.

Miller, D. L. (1981). The Meaning of Role-Taking, *Symbolic Interaction* 4(2): 167–175.

Minsky, M. L. (1975). A framework for representing knowledge, *in* P. H. Winston (ed.), *The Psychology of Computer Vision*, McGraw-Hill, New York, NY.

Mitchell, T. (1997). *Machine Learning*, McGraw-Hill, New York, NY.

Moss, S. and Davidsson, P. (eds.) (2000). *Multi-Agent Based Simulation. Second International Workshop, MABS-2000, Boston, MA, USA, July 2000*, Lecture Notes in Artificial Intelligence, vol. 1979, Springer-Verlag, Berlin et al.

Moulin, B. and Chaib-Draa, B. (1996). An overview of distributed artificial intelligence, *in* G. O'Hare and N. Jennings (eds.), *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, New York, NY, pp. 3–55.

Müller, H.-J. and Dieng, R. (eds.) (2000). *Computational Conflicts - Conflict Modeling for Distributed Intelligent Systems*, Springer-Verlag, Berlin et al.

Müller, J. P. (1997). A cooperation model for autonomous agents, *in* J. Müller, M. J. Wooldridge and N. Jennings (eds.), *Intelligent Agents III*, Lecture Notes in Artificial in Artificial Intelligence, vol. 1193, Springer-Verlag, Berlin et al.

Nagl, L. (1998). *Pragmatismus*, Campus Verlag, Frankfurt am Main, New York, NY.

Newell, A. (1990). *Unified theories of cognition*, Harvard University Press, Cambridge, MA.

Nicholson, A. E. and Brady, J. M. (1994). Dynamic belief networks for discrete monitoring, *IEEE Transactions on Systems, Man, and Cybernetics*, 24(11):1593–1610.

Nickles, M. and Lorentzen, K. F. (2003). Multiagent Systems without Agents – Mirror-Holons for the Derivation and Enactment of Functional Communication Structures, *in* K. Fischer and M. Florian (eds.), *Socionics: Its Contributions to the Scalability of Complex Social Systems*, Lecture Notes in Computer Science, Springer-Verlag, Berlin et al. To appear.

Nickles, M. and Rovatsos, M. (2004). Communication Systems: A Unified Model of Socially Intelligent Systems, *in* K. Fischer and M. Florian (eds.), *Socionics: Its Contributions to the Scalability of Complex Social Systems*, Lecture Notes in Computer Science, Springer-Verlag, Berlin et al. To appear.

Nickles, M. and Weiß, G. (2003). A Framework for the Social Description of Resources in Open Environments, *Proceedings of the 7th International Workshop on Cooperative Information Agents (CIA-2003)*, Lecture Notes in Computer Science, vol. 2782, Springer-Verlag, Berlin et al.

Nickles, M. and Weiss, G. (2004). Multiagent Systems without Agents – Mirror-Holons for the Compilation and Enactment of Communication Structures, *in* K. Fischer and M. Florian (eds.), *Socionics: Its Contributions to the Scalability of Complex Social Systems*, Lecture Notes in Computer Science, Springer-Verlag, Berlin et al. To appear.

Nickles, M., Rovatsos, M. and Weiss, G. (2002). A Schema for Specifying Computational Autonomy, *in* P. Petta, R. Tolksdorf and F. Zambonelli (eds.), *Engineering Societies in the Agents World III, Third International Workshop (ESAW-2002), Madrid, Spain, September 16–17, 2002, Revised Papers*, Lecture Notes in Artificial Intelligence, vol. 2577, Springer-Verlag, Berlin et al.

Nickles, M., Rovatsos, M. and Weiss, G. (2004a). Empirical-Rational Semantics of Agent Communication, *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-04)*, New York, NY. To appear.

Nickles, M., Rovatsos, M., Brauer, W. and Weiss, G. (2004b). Towards a Unified Model of Sociality in Multiagent Systems, *International Journal of Computer and Information Science* 5(1). To appear.

Nwana, H. (1996). Software agents: An overview, *The Knowledge Engineering Review* 11(3): 205–244.

Odell, J., Parunak, V. and Bauer, B. (2000a). Extending UML for agents, *in* G. Wagner, Y. Lespérance, and E. Yu (eds.), *Proceedings of the 2nd Agent-Oriented Information Systems (AOIS) Workshop at the 17th National Conference on Artificial Intelligence*, pp. 3–17.

Odell, J., Parunak, V. and Bauer, B. (2000b). Representing agent interaction protocols in UML, *Working Notes of the First International Workshop on Agent-Oriented Software Engineering (AOSE-2000)*.

O'Hare, G. and Jennings, N. (eds.) (1996). *Foundations of Distributed Artificial Intelligence*, John Wiley & Sons, New York, NY.

Ong, K.-L. and Ng, W.-K. (1998). A Survey of Multi-Agent Interaction Techniques and Protocols, *Technical Report CAIS-TR04-98*, School of Applied Science, Nanyang Technological University, Singapore.

Paetow, K. and Rovatsos, M. (2003). Grundlagen einer interaktionistischen Sozionik, *Research Report RR-8*, Department of Technology Assessment, Technical University of Hamburg, Hamburg.

Panzarasa, P. and Jennings, N. R. (2001). Negotiation and Joint Commitments in Multi-Agent Systems, *Proceedings of the Second International Workshop on Modelling Artificial Societies and Hybrid Organisations (MASHO-01)*, Vienna, Austria.

Panzarasa, P., Jennings, N. R. and Norman, T. J. (2002). Formalising collaborative decision-making and practical reasoning in multi-agent systems, *Journal of Logic and Computation* 12(1): 55–117.

Panzarasa, P., Norman, T. J. and Jennings, N. R. (1999). Modeling sociality in the BDI framework, *Proceedings of the First Asia-Pacific Conference on Intelligent Agent Technology (IAT-99)*.

Parr, R. E. (1998). *Hierarchical Control and Learning for Markov Decision Processes*, PhD Thesis, University of California, Berkeley, CA.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Francisco, CA.

Pitt, J. and Mamdani, A. (1999a). Designing Agent Communication Languages for Multi-Agent Systems, *in* F. Garijo and M. Boman (eds.), *Multi-Agent System Engineering. Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-99)*, Lecture Notes in Artificial Intelligence, vol. 1647, Springer-Verlag, pp. 102–114.

Pitt, J. and Mamdani, A. (1999b). A Protocol-Based Semantics for an Agent Communication Language, *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99)*.

Precup, D. (2000). *Temporal Abstraction in Reinforcement Learning*, PhD thesis, Department of Computer Science, University of Massachusetts, Amherst, MA.

Puterman, M. L. (1994). *Markov Decision Problems*, John Wiley & Sons, New York, NY.

Quintero, A., Ucrós, M. and Takhashi, S. (1995). Multi-agent systems protocol language specification, *Proceedings of the CIKM Workshop on Intelligent Information Agents*.

Rahwan, I., Ramchurn, S. D., Jennings, N. R., McBurney, P., Parsons, S. and Sonenberg, L. (2004). Argumentation-based negotiation, *Knowledge Engineering Review*. To appear.

Rahwan, I., Sonenberg, L. and Dignum, F. (2003). Towards Interest-Based Negotiation, *in* J. S. Rosenschein, T. Sandholm, M. Wooldridge and M. Yokoo (eds.), *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, Australia.

Raiffa, H. (1982). *The Art and Science of Negotiation*, Harvard University Press, Cambridge, MA.

Rao, A. S. and Georgeff, M. (1992). An abstract architecture for rational agents, *in* W. S. C. Rich and B. Nebel (eds.), *Proceedings of Knowledge Represenation and Reasoning (KR&R-92)*.

Reckwitz, A. (2000). *Die Transformation der Kulturtheorien. Zur Entwicklung eines Theorieprogramms*, Velbrück Wissenschaft, Weilerswist, Germany.

Rimassa, G. and Viroli, M. (2002). An Operational Framework for the Semantics of Agent Communication Languages, *Preproceedings of the Third International Workshop Engineering Societies in the Agents World (ESAW-02), Madrid, Spain, September 16-17, 2002*.

Rosenschein, J. and Zlotkin, G. (1994). *Rules of Encounter*, The MIT Press, Cambridge, MA.

Rovatsos, M. (2000). Agent-based opinion dissemination: A perspective for "social linkage" on the internet. Unpublished internal memo.

Rovatsos, M. (2001). Interaction frames for artificial agents, *Technical Report Research Report FKI-244-01*, AI/Cognition Group, Department of Informatics, Technical University of Munich, Munich, Germany.

Rovatsos, M. (2002–2004). *LIESON – User's Manual and Developer's Guide*, http://www7.in.tum.de/~rovatsos/lieson/users-manual.pdf.

Rovatsos, M. and Lind, J. (2000). Hierarchical Common-Sense Interaction Learning, *in* E. H. Durfee (ed.), *Proceedings of the Fifth International Conference on Multi-Agent Systems (ICMAS-00)*, Boston, MA, IEEE Computer Society Press.

Rovatsos, M. and Paetow, K. (2004). On the Organisation of Social Experience: Scaling up Social Cognition, *in* K. Fischer and M. Florian (eds.), *Socionics: Its Contributions to the Scalability of Complex Social Systems*, Lecture Notes in Computer Science, Springer-Verlag, Berlin et al. To appear.

Rovatsos, M. and Weiß, G. (2001). Achieving Multiagent Organisation by Organising Agent Experience, *in* Y. Demazeau (ed.), *Pre-proceedings of the 10th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW-01), Annecy, France*.

Rovatsos, M. and Wolf, M. (2002). Towards Social Complexity Reduction in Multiagent Learning: the ADHOC Approach, *in* K. Tumer and P. Stone (eds.), *Collaborative Learning Agents. Papers from 2002 AAAI Spring Symposium.*, also published as *Technical Report SS-02-02*, AAAI Press, Stanford, CA.

Rovatsos, M., Nickles, M. and Weiß, G. (2003a). Interaction is Meaning: A New Model for Communication in Open Systems, *in* J. S. Rosenschein, T. Sandholm, M. Wooldridge and M. Yokoo (eds.), *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, Australia.

Rovatsos, M., Nickles, M. and Weiss, G. (2004). An Empirical Model of Communication in Multiagent Systems, *in* F. Dignum (ed.), *Advances in Agent Communication*, Lecture Notes in Artificial Intelligence, vol. 2922, Springer-Verlag, Berlin et al.

Rovatsos, M., Weiß, G. and Wolf, M. (2002). An Approach to the Analysis and Design of Multiagent Systems based on Interaction Frames, *in* M. Gini, T. Ishida, C. Castelfranchi and W. L. Johnson (eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, ACM Press, Bologna, Italy.

Rovatsos, M., Weiß, G. and Wolf, M. (2003b). Multiagent Learning for Open Systems: A Study in Opponent Classification, *in* E. Alonso, D. Kazakov and D. Kudenko (eds.), *Adaptive Agents and Multi-Agent Systems*, Lecture Notes in Artificial Intelligence, vol. 2636, Springer-Verlag, Berlin et al.

Russell, S. and Wefald, E. (1991). Principles of rationality, *Artificial Intelligence* 49(1-3): 361–395.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence. A Modern Approach*, 2 edn, Pearson Education (Prentice-Hall), Upper Saddle River, NJ.

Russell, S. J. and Subramanian, D. (1995). Provably Bounded-Optimal Agents, *Journal of Artificial Intelligence Research* 2: 595–609.

Saam, N. J. and Harrer, A. (1999). Simulating norms, social inequality and functional change in artificial societies., *Journal of Artificial Societies and Social Simulation* 2(1).

Sadek, M. D. (1991). Dialogue acts are rational plans, *Proceedings of the ESCA/ETRW Workshop on the Structure of multimodal Dialogue*, pp. 1–29.

Sandholm, T. and Lesser, V. (1995). Issues in automated negotiation and electronic commerce: Extending the contract net framework, *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pp. 328–335.

Sandholm, T. W. (1999). Distributed rational decision making, *in* G. Weiß (ed.), *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, MA, pp. 201–253.

Sartor, G. (2001). Why Agents Comply with Norms and why they Should, *in* R. Conte and C. Dellarocas (eds.), *Social Order in Multiagent Systems*, Kluwer Academic Publishers, Norwell, MA, Amsterdam, The Netherlands, pp. 19–44.

Schank, R. C. and Abelson, R. P. (1977). *Scripts, Plans, Goals, and Understanding*, Lawrence Erlbaum Associates, Potomac, MD.

Schillo, M. and Fischer, K. (2001). The contract-net with confirmation protocol: A solution to a fundamental problem of DAI, *Technical memo TM-01-01*, DFKI, Saarbrücken/Kaiserslautern, Germany. in print.

Searle, J. R. (1969). *Speech Acts: An Essay on the Philosophy of Language*, Cambridge University Press, Cambridge, England.

Sen, S. and Weiß, G. (1999). Learning in multiagent systems, *in* G. Weiß (ed.), *Multiagent Systems*, The MIT Press, Cambridge, MA, chapter 6, pp. 259–298.

Sichman, J., Bousquet, F. and Davidsson, P. (eds.) (2003). *Multi-Agent Based Simulation II. Third International Workshop, MABS-2002, Bologna, Italy, July 15-16, 2002*, Lecture Notes in Artificial Intelligence, vol. 2581, Springer-Verlag, Berlin et al.

Sichman, J. S., Conte, R. and Gilbert, N. (eds.) (1998). *Multi-Agent Systems and Agent-Based Simulation. First International Workshop, MABS-1998, Paris, France, July 1998*, Lecture Notes in Artificial Intelligence, vol. 1543, Springer-Verlag, Berlin et al.

Singh, M. (1997). Interaction-oriented programming for the Web. Position Statement, *International Conference on Cooperative Information Systems (CoopIS-97)*.

Singh, M. (2000). A Social Semantics for Agent Communication Languages, *Proceedings of the IJCAI Workshop on Agent Communication Languages*.

Singh, M. P. (1993). A semantics for speech acts, *Annals of Mathematics and Artificial Intelligence* 8(1–2): 47–71.

Smith, R. (1980). The contract-net protocol: High-level communication and control in a distributed problem solver, *IEEE Transactions on Computers* 29(12): 1104–1113.

Smith, R. and Davis, R. (1981). Frameworks for cooperation in distributed problem solving, *IEEE Transactions on Systems, Man, and Cybernetics* 11(1): 61–70.

Steels, L. (1998). The Origins of Ontologies and Communication Conventions in Multi-Agent Systems, *Autonomous Agents and Multi-Agent Systems* 1(2): 169–194.

Steels, L. (2003). The evolution of communication systems by adaptive agents, *in* E. Alonso, D. Kazakov and D. Kudenko (eds.), *Adaptive Agents and Multi-Agent Systems*, Lecture Notes in Artificial Intelligence, Springer-Verlag, vol. 2636, Berlin et al., pp. 125–140.

Steels, L. and Vogt, P. (1997). Grounding adaptive language games in robotic agents, *in* C. Husbands and I. Harvey (eds.), *Proceedings of the Fourth European Conference on Artificial Life*, The MIT Press, Cambridge, MA.

Stone, P. and Veloso, M. (1996). Collaborative and Adversarial Learning: A Case Study in Robotic Soccer, *in* S. Sen (ed.), *Adaptation, Coevolution and Learning in Multiagent Systems. Papers from the 1996 AAAI Symposium*, Technical Report SS-96-01, AAAI Press, Menlo Park, CA, pp. 88–92.

Stone, P. (ed.) (2000). *Layered learning in multiagent systems. A winning approach to robotic soccer*, The MIT Press, Cambridge, MA.

Strauss, A. L. (1978a). *Negotiations: Varieties, Contexts, Processes and Social Order*, Jossey-Bass, San Francisco et al.

Strauss, A. L. (1978b). A Social World Perspective, *Studies in Symbolic Interaction* 1: 119–128.

Strauss, A. L. (1993). *Continual Permutations of Actions*, Aldine de Gruyter, New York, NY.

Strübing, J. (1998a). Bridging the Gap, *Symbolic Interaction* 21(4).

Strübing, J. (1998b). Multiagenten-Systeme als "Going Concern", *in* T. Malsch (ed.), *Sozionik – Soziologische Ansichten über künstliche Sozialität*, Edition Sigma, Berlin.

Sutton, R. and Barto, A. (1998). *Reinforcement Learning. An Introduction*, The MIT Press/A Bradford Book, Cambridge, MA.

Sutton, R. S., Precup, D. and Singh, S. (1999). Between MDPs and semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning, *Artificial Intelligence* 112: 181–211.

Tamma, V., Wooldridge, M. and Dickinson, I. (2002). An ontology for automated negotiation, *Proceedings of the International Workshop on Ontologies in Agent Systems (OAS-02)*, Bologna, Italy.

Tan, M. (1993). Multi-Agent Reinforcement Learning: Independent vs. Cooperative Agents, *Proceedings of the Tenth International Conference on Machine Learning*, pp. 330–337.

Tesauro, G. and Bredin, J. L. (2002). Strategic Sequential Bidding in Auctions using Dynamic Programming, *in* M. Gini, T. Ishida, C. Castelfranchi and W. L. Johnson (eds.), *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-02)*, Bologna, Italy, ACM Press, pp. 394–401.

Tessier, C., Müller, H.-J., Fiorino, H. and Chaudron, L. (2000). *Conflicting Agents – Conflict management in multi-agent systems*, Kluwer Academic Publishers, Norwell, MA, Amsterdam, The Netherlands.

Turner, J. H. (1988). *A Theory of Social Interaction*, Stanford University Press, Stanford, CA.

Verhagen, H. J. E. (2000). *Norm Autonomous Agents*, PhD thesis, Department of Computer and Systems Sciences, The Royal Institute of Technology and Stockholm University, Stockholm, Sweden.

Vidal, J. M. and Durfee, E. H. (1997). Agents learning about agents: A framework and analysis, *Collected Papers from AAAI-97 Workshop on Multiagent Learning*, AAAI Press, pp. 71–76.

Watkins, C. and Dayan, P. (1992). Q-learning, *Machine Learning* 8: 279–292.

Watson, I. and Marir, F. (1994). Case-based reasoning: A review, *The Knowledge Engineering Review* 9(4): 327–354.

Wegner, P. (1997). Why interaction is more powerful than computing, *Communications of the ACM* 40(5): 80–91.

Weick, K. (1979). *The Social Psychology of Organizing*, McGraw-Hill, New York, NY.

Weiß, G. (1995). Distributed Reinforcement Learning, *Robotics and Autonomous Systems* 15: 135–142.

Weiß, G. (1996). Adaptation and Learning in Multi-Agent Systems: Some Remarks and a Bibliography, *in* G. Weiß and S. Sen (eds.), *Adaption and Learning in Multiagent Systems*, Lecture Notes in Artificial in Artificial Intelligence, vol. 1042, Springer-Verlag, Berlin et al.

Weiß, G. (1998). A Multiagent Perspective of Parallel and Distributed Machine Learning, *Proceedings of the 2nd International Conference on Autonomous Agents*, pp. 226–230.

Weiss, G. (2001). Agent Orientation in Software Engineering, *Knowledge Engineering Review* 16(4): 349–373.

Weiß, G. and Dillenbourg, P. (1999). What is “multi” in multiagent learning?, *in* P. Dillenbourg (ed.), *Collaborative Learning: Cognitive and Computational approaches*, Pergamon Press, chapter 4, pp. 64–80.

Weiß, G. and Sen, S. (eds.) (1996). *Adaption and Learning in Multiagent Systems*, Lecture Notes in Artificial in Artificial Intelligence, vol. 1042, Springer-Verlag, Berlin et al.

Weiß, G. (ed.) (1997). *Distributed Artificial Intelligence Meets Machine Learning*, Lecture Notes in Artificial in Artificial Intelligence, vol. 1221, Springer-Verlag, Berlin et al.

Weiß, G. (ed.) (1999). *Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence*, The MIT Press, Cambridge, MA.

Weiß, G., Rovatsos, M., Nickles, M. and Meindl, C. (2003). Capturing Agent Autonomy in Roles and XML, *in* J. S. Rosenschein, T. Sandholm, M. Wooldridge and M. Yokoo (eds.), *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03)*, Melbourne, Australia.

Wolf, M. (2002). *Strategic Opponent Classification in Multiagent Systems: Potential and Limitations of Social Complexity Reduction*, Master's thesis, Computer Science Department, Technical University of Munich.

Wooldridge, M. J. (1999). Intelligent Agents, *in* G. Weiss (ed.), *Multiagent Systems*, The MIT Press, Cambridge, MA, pp. 27–77.

Wooldridge, M. J. and Jennings, N. (1995a). Agent theories, architectures, and languages: A survey, *in* M. J. Wooldridge and N. Jennings (eds.), *Intelligent Agents*, Lecture Notes in Artificial in Artificial Intelligence, vol. 890, Springer-Verlag, Berlin et al., pp. 1–39.

Wooldridge, M. J. and Jennings, N. (eds.) (1995b). *Intelligent Agents*, Lecture Notes in Artificial in Artificial Intelligence, vol. 890, Springer-Verlag, Berlin et al.

Wooldridge, M. J. and Jennings, N. R. (1995c). Intelligent Agents: Theory and Practice, *The Knowledge Engineering Review* 10(2): 115–152.

Wooldridge, M. J., Weiss, G. and Ciancarini, P. (eds.) (2002). *Agent-Oriented Software Engineering II. Second International Workshop, AOSE 2001, Montreal, Canada, May 29, 2001*, Lecture Notes in Computer Science, vol. 2222, Springer-Verlag, Berlin et al.

Yokoo, M., Durfee, E., Ishida, T. and Kuwabara, K. (1992). Distributed Constraint Satisfaction for Formalizing Distributed Problem Solving, *Proceedings of the Twelfth IEEE International Conference on Distributed Computing Systems*, pp. 614–621.