An adaptable agent-based control model for rural electrification for small wind and solar hybrid off-grid systems

Anna-Kaarina Seppälä[1,2]
1. Technical University of Munich
2. University of Toulouse III

Keywords: agent-based modelling, energy management, rural electrification, off-grid, hybrid renewable energy system, C++

***Abstract* – This paper presents a multi-agent control system for off-grid power management in wind-solar hybrid energy systems, with special focus on rural electrification. The underlying framework is designed with special attention on modularity and adaptability. The light-weight implementation ensures optimal functionality on low-cost, low-performance hardware essential for cost minimization in rural contexts. The power management method is based on multi-agent systems (MAS) utilising the Belief-Desire-Intention (BDI) paradigm. The proposed MAS for power management context comprises solar, wind, battery storage, load and central controller (MGCC) agents. The design process and communication structure of the agent-based system are presented.**

I. INTRODUCTION

An estimated 1.1 billion people world-wide have no access to electricity [1], [2], and lack of energy is a major contributor to poverty [3]. Rural electrification is the process of bringing energy access to those in need, mostly in isolated and rural regions. In this context, off-grid renewable energy systems are the most appropriate solution [4].

Renewable energy is widely available, and independent from the global fossil-fuel market [3]. But, due to the intermittency of wind and solar-based power generation, combining multiple sources with energy storage is essential [5], [6]. This type of hybrid configuration creates the need to control the energy flow among components within the energy system [4]. Intelligent, fast and adaptable local control at a low price is a necessity [11].

Current power management methods rely mostly on a centralised supervisory control and data gathering, or SCADA approach, which has proven inflexible and insufficient for the needs of distributed energy systems [13]. Over the past decade, *multi-agent systems*, or *MAS,* have emerged as a viable option for microgrid control [9].

Agents are autonomously operating (software or hardware) entities that react to changes in their environment [11]. *Intelligent agents* are more complicated systems characterised by their pro-activeness, i.e. their ability to take initiative and pursue internal goals. They also possess a degree of social ability, and can interact with each other [12]. *A multi-agent system* consists of a group of *agents* or *intelligent agents*. A fundamental attribute of MAS is the absence of an explicit, hard-coded system-level goal. Instead, the desired outcome arises as emergent behaviour from the actions and deliberations of individual agents [8], [11].

One of the main advantages of MAS is their suitability for distributed applications, stemming from the inherent modularity of agent-based systems. Agents can be easily added or removed without significantly affecting the functionality of the overall system. This increases fault-tolerance and allows for plug-and-play [9]. Another factor that makes MAS ideal for distributed problems is the agents' capability to function with local and restricted knowledge. Thus even in the event of total communication loss, individual agents can stay operative [10].

In distributed power generation, MAS may be used for monitoring and diagnostics, self-healing, power scheduling and dispatch, as well as voltage and frequency (if AC bus) control, etc. [11]. In this context, agents usually represent physical components of the microgrid, such as generators and loads.

To make the control system affordable in a rural setting, the cost of both hardware and software should be kept to a minimum. This critically limits the available computational power and prevents the use of commercial software.

An effective power management methodology for off-grid applications is the main contribution of this paper. Special attention is given to creating an adaptable, fast and performant MAS framework, upon which the power management system is built.

II. RELATED WORK

There is an extensive body of research on multi-agent systems in the field of energy systems control. Many works focus on microgrids that may either be

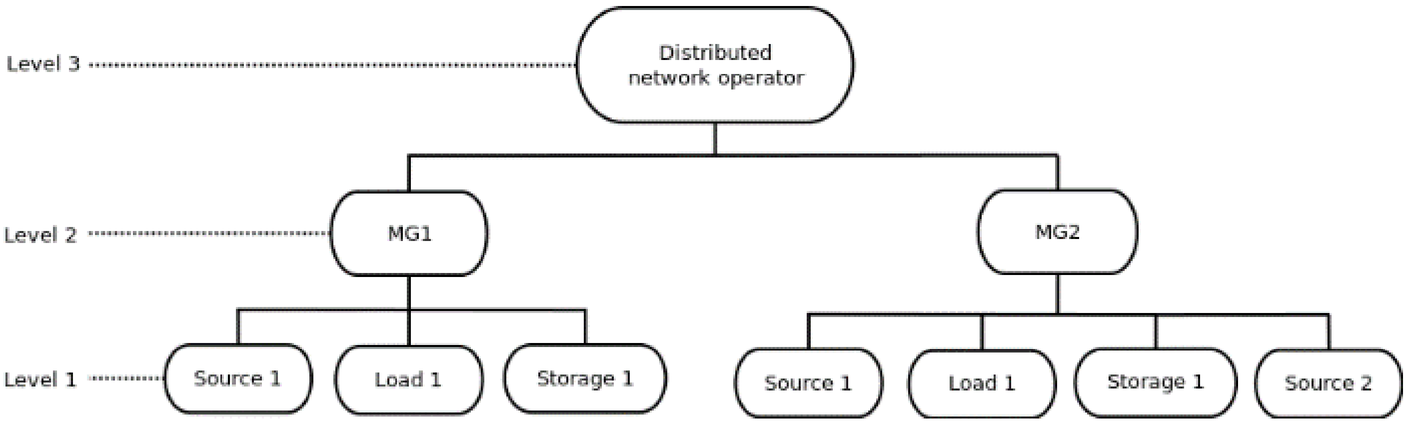

Figure 1: The three layers of agent hierarchy in a power system.

connected to the main grid, or work in islanded mode [9]. Such applications usually employ a two- or three-level hierarchy of agents. The top level of a three-level hierarchy corresponds to a distribution network operator connecting the microgrid to other microgrids or the main power network. The medial layer comprises the central controller, responsible for coordinating the components of an individual microgrid. The lowest layer consists of agents representing these components (see Figure 1) [9], [14].

The vast majority of MAS have been realised in the commercial Matlab/Simulink environment [4], rendering them unreproducible in a rural, low-cost scenario. Open-source agent-development frameworks, such as java-based JADE and ZEUS, have also widely been used to create power management systems [14]. Such frameworks, however, tend to favour expressivity over run-time performance, making them too heavy for embedded systems with limited computational power and strict real-time requirements [15], [16].

Some researchers have proposed their own agent-based power control systems [17], [18], [19], specifically for the rural Indian scenario. Implemented in embedded C and light enough to run on devices with very little memory, they provide the bare minimum of agent-based control. However, these systems run on a central microcontroller, thus losing the advantages MAS offer in distributed control. A malfunction in the microcontroller jeopardises the whole system. Furthermore, the MAS presented in [17] and [19] are platform-specific, and not based on a framework, which makes possible alterations and expansion of the microgrid cumbersome.

Finally, Betts and Müller present a component-based C++ framework for agent-based systems in [20], [21], aiming for modularity and usage of low-performance hardware. Nevertheless, the usage of polymorphism, dynamic data allocation and smart pointers makes the framework impractical for devices with heavy memory constraints.

III. PROBLEM STATEMENT

A control strategy is responsible for supply stability, maximization of the power output, and power quality, such as voltage regulation at the user end [4], [7]. Solutions in the literature have been designed for urban scenarios with the possibility of main-grid connection, thus providing overly complicated solutions for rural contexts, where off-grid power generation is essential, not supplementary [17]. The added computational burden, along with the use of commercial software, render current control systems impractical when the implementation cost hast to be minimised.

Furthermore, most developed MAS are strongly dependent on the chosen system architecture, and cannot easily be adapted to changes within the system [7]. The lack of adaptability and re-usability of existing control systems is detrimental to widespread rural electrification, as the development of customised control systems is seldom affordable in low-income areas. On the contrary, an adaptable system could be adopted by a wider range of users, in various scenarios, promoting development.

IV. METHOD

In the absence of an adaptable, light-weight framework for computationally restricted agent-based systems, one is proposed as part of the goal of developing a low-cost power management system for rural electrification. C++ was chosen for the implementation as it is one of the fastest and most widely-developed high-level programming languages [15], available on a wide range of hardware [20]. The framework makes extensive use of template metaprogramming and static polymorphism to minimise the computational burden of the framework. For the same reason, dynamic memory allocation is avoided. In this section, an overview of the framework is first given, after which the final agent-based power management system is presented.

A. Agent Design Method

The proposed multi-agent framework is based on the design methodology created at the Australian Artificial Intelligence Institute (AAII) and introduced by Kinny et al., which provides a technique for the modelling of BDI (Belief Desire Intention) agents [22]. BDI is a framework that describes the behaviour of intelligent agents, and how they can be made to reach internal goals [23]. In the AAII the MAS design process, two distinct levels of abstraction are adopted; the external, and the internal viewpoint.

1. External Viewpoint

The external agent model describes the hierarchical relationship between different agents in a given MAS, as well as the interactions between them.

2. Internal Viewpoint

The internal functionality of individual agents is captured by the BDI architecture, and an agent is fully defined by the events it perceives, its actions, the beliefs it holds about its environment, the goals it may pursue, and the plans it may adopt to do so [22]. These are expressed by the following models:

- Belief model: The beliefs of an agent represent the information it has of its environment and internal state. These affect the actions available to the agent at any given time. Beliefs may or may not correspond to reality.
- Goal model: This model defines all possible goals for the agent, and the events that can trigger a response. Goals are desires that the agent is actively pursuing and committed to until they either succeed or are deemed impossible.
- Plan model: The plans of an agent describe the procedures an agent can adopt to achieve its goals. Each plan is a tree-like structure of elements, subgoals and atomic actions. When the agent commits to a plan element, it is denoted as an intention.

In the MAS design process, the external model is created first, after which the internal structure of each agent is determined. Figure 2 shows a schematic of the AAII methodology.

B. Multi-Agent Control System

The developed MAS model was created using the AAII design methodology. The current version consists of a two-level hierarchy, with a *microgrid central controller* (MGCC) taking the role of an auctioneer in the scheduling process, and providing reference voltage and possibly frequency to the

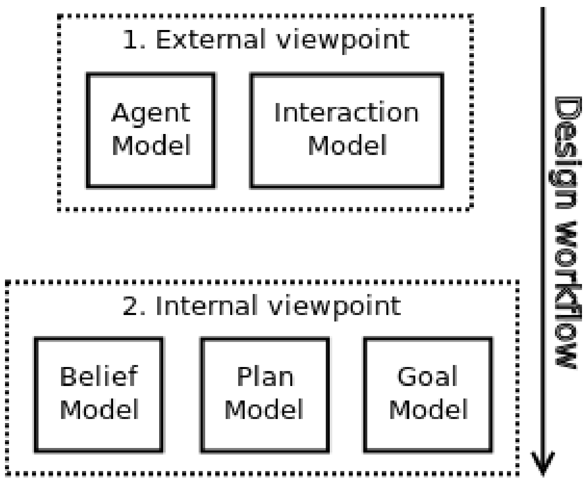

Figure 2: The AAII methodology of creating agent-based systems.

lower layer that comprises the source, load, and storage agents. At hourly intervals, all agents take part in an auction-based scheduling scheme, initiated by the MGCC Agent. First, Load Agents send their requests (desired amount of power) to the Source Agents, which will consider incoming requests, then send in their production offers, or bids, consisting of the amount of offered power and the corresponding price. Load Agents accept offers until production matches demand, or until there are no offers left. In the latter case, a Load Agent may request a bid from available Storage Agents to initiate a similar negotiation procedure as earlier with the Source Agents. If the overall demand is not satisfied at the end of the auction, some of the load will be shed. Conversely, if there is overproduction, sources will offer the surplus to the Storage Agents. If the problem still exists after negotiations, the Source Agent concerned will have to direct its unsold power into a dump load or disconnect itself from the grid. The MGCC agent will wait for all other agents to report back before ending the auction.

The hierarchy and interactions of the agents is shown in Figure 3.

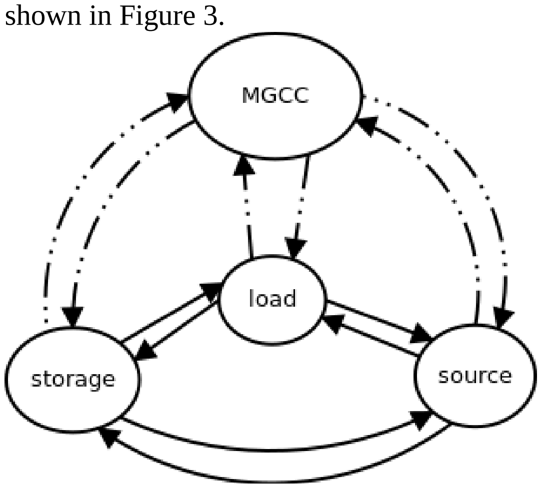

Figure 3: Interaction between agents in the microgrid.

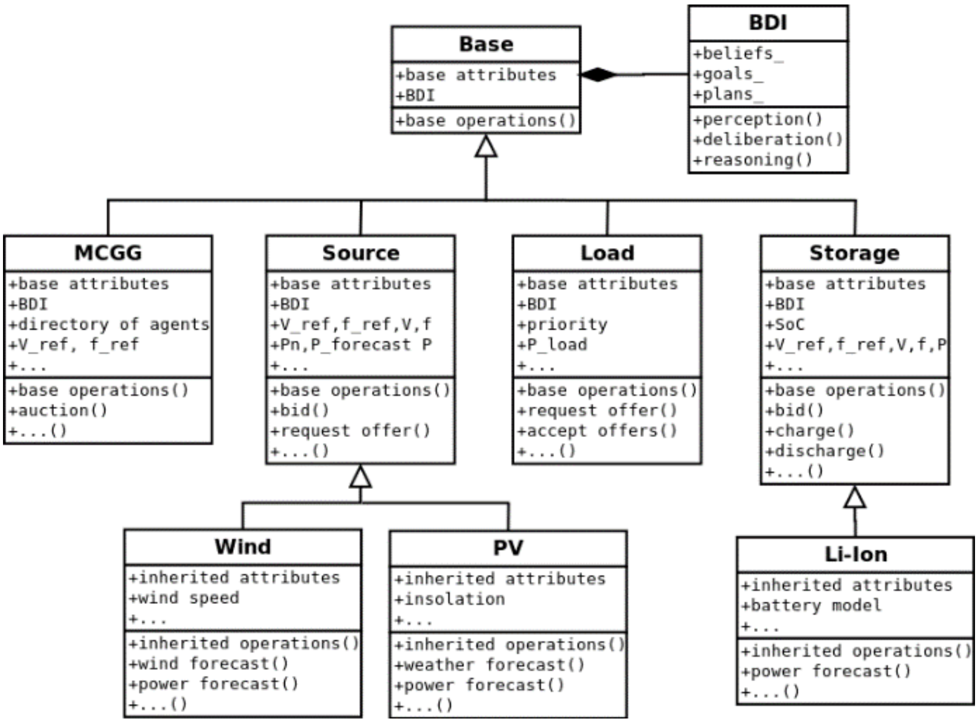

Figure 4: UML diagram of the agent-based control system.

In the C++ implementation, all of the agents inherit from a base class that provides the BDI model and the messaging functionality, etc. More specialised agents again inherit from more general ones (e.g., a wind resource agent will inherit the general source agent class). See Figure 4 for the UML diagram of different components of the system.

In the following, an overview of the internal models of each agent type is given.

1. *MGCC Agent*

The beliefs of the MGCC Agent mostly concern the other agents in the power system, and it has no selfish goals. It holds a list of active and inactive agents, and thus knows which agents to send an invitation to when it initiates the hourly scheduling auction. It also keeps track of system time to ensure the auction is started and ended on time. Furthermore, it receives voltage and possibly frequency measurements from the common bus to compute the difference between the desired and current values.

The desires of the MGCC Agent are to hold an auction at hourly intervals, and to keep the voltage of the off-grid system at reference levels. If a source is power levels that differ from the scheduled amount, the MGCC Agent can request that source to disconnect. If unresponsive, the MGCC and force the disconnection.

2. *Source Agent*

The Source Agent has beliefs both about the environment and about itself. It also needs to keep track of time, and receives the voltage and frequency reference values from the MGCC. At the start of an auction, the MGCC also provides it with the IDs of the active Load and Storage Agents whom to send requests to. During negotiation, critical loads are given priority and are the first to receive a generation offer. The internal information of the Source Agent consists of its nominal power, current power and the forecasted power for the next hour, upon which its generation offer will be based. It also has a notion of its power production cost and a profit margin with which it calculates its bidding price. If a Source Agent mostly loses or wins auctions, the profit margin could be modified but in the current version, it is constant.

Like the MGCC, the Source Agent also desires to keep voltage at the reference levels imposed by the former. It also wishes to maximize its profit during the hourly auction.

The goal to maximise profit gives rise to three different plans: Selling available power, diverting surplus to a dump load or, in the worst-case scenario, disconnecting oneself from the microgrid.

3. *Load Agent*

The beliefs of the Load Agent consist of time, and the information of active sources and storage provided by the MGCC Agent at the start of the auction. Also, the agent knows its power need, and whether or not the load it represents is controllable or critical. Controllable loads may be shed without complications in case of underproduction, and critical loads are given priority during power scheduling negotiations.

In the current MAS, the Load Agent has only one desire; satisfying its power need. To do so, two plans are available. Either the agent succeeds in buying all the needed power from sources and storages or it needs to shed some or all of the load.

4. *Storage Agent*

The Storage Agent is aware of its own power storing capacity, and its current state of charge (SoC). It also has SoC safety limits, within which the storage unit needs to operate. Instantaneous and reference values for voltage and possibly frequency are also part of its belief set.

The desires of the Storage Agent include keeping its SoC between accepted limits, controlling voltage and frequency (if AC bus) of the common voltage bus, and helping others during auction. If its SoC allows it, the agent will either help Load Agents by discharging in case of power shortage, or take on the surplus of the Source Agents in the event of overproduction. The plan structure for helping others is shown in Figure 5.

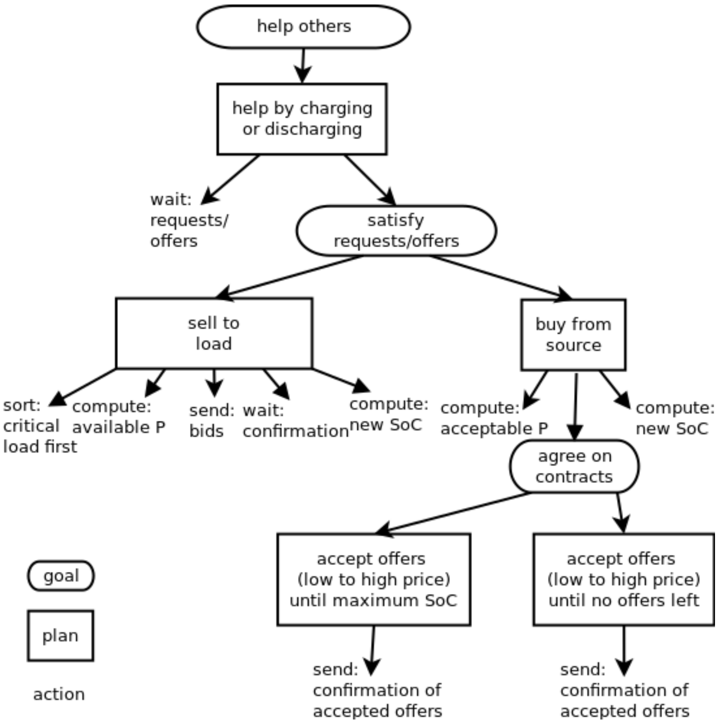

Figure 5: Plan structure of the goal of helping others during auctioning. To fulfil a goal, one of the available plans is selected. All steps of that plan must succeed to achieve the goal. If a plan has multiple steps, they are acted upon from left to right.

All of the agents are single-mindedly committed to their goals, and will thus pursue them until they are either achieved or deemed impossible to attain.

VI. CONCLUSION AND PERSPECTIVE

This paper addresses the need for an adaptable control method by proposing an open-source, modular control system capable of conforming to different system architectures and demand-side requirements. The modularity of the underlying framework ensures that it can be extended if the energy system is changed due to growing storage capacity or power needs, and modified by adding or removing beliefs, goals or plans of the agents. New agents can easily be created. For instance, a distribution network operator agent could be added to the system in the case it is connected to the main power network.

Agent-based modelling strategies are explored in the context of energy management and control. Each system component is modelled as an autonomous unit with individual goals, such as maximising the power output or maintaining optimal state-of-charge. Agents are designed to reason based on their own state, and changes within the rest of the system. The control system is implemented in C++, and is deliberately optimised for embedded systems with limited computational capabilities. This ensures low cost of implementation necessary for the rural electrification context.

The next steps for the project include implementation of missing features, such as improved power forecasting algorithms for specialised Source Agents, and possibly removing the MGCC Agent in order to reduce the probability of single-point failures. The responsibility of auctioning synchronization could be passed from one agent to the next depending on their activities and connection status (whether they are connected to or disconnected from the rest of the off-grid).

Further into the future, the power management system will also be tested on real-time devices, such as low-cost microcontrollers to prove the capabilities of the underlying network.

REFERENCES

[1] A. Doukas and A. Ballesteros, “Clean Energy Access in Developing Countries: Perspectives on Policy and Regulation,” *World Resour. Inst. Keys To Achiev. Univers. Energy Access Ser.*, vol. 2/3, 2015.

[2] B. K. Sovacool, M. Bazilian, and M. Toman, “Paradigms and poverty in global energy policy: research needs for achieving universal energy access,” *Environ. Res. Lett.*, vol. 11, no. 6, p. 64014, 2016.

[3] J. Sumanik-Leary, M. Delor, M. Little, M. Bellamy, A. Williams, and S. Williamson, *Engineering in Development - Energy*. London, England: Engineers Without Borders UK, 2014.

[4] A. Chauhan, “A review on Integrated Renewable Energy System based power generation for stand-alone applications: Configurations, storage options, sizing methodologies and control,” *Renew. Sustain. Energy Rev.*, vol. 38, pp. 99–120, 2014.

[5] IEA, “Technology Roadmap: Energy Storage,” 2015.

[6] V. O. Okinda and N. A. Odero, “A Review of

Techniques in Optimal Sizing of Hybrid Renewable Energy Systems," *Int. J. Res. Eng. Technol.*, vol. 4, no. 11, pp. 153–163, 2015.

[7] L. Olatomiwa et al., "Energy management strategies in hybrid renewable energy systems: A review," *Renew. Sustain. Energy Rev.*, vol. 62, pp. 821–835, 2016.

[8] M. N. C. Macal, "Introductory Tutorial: Agent-Based Modelling and Simulation," *Proc. 2014 Winter Simul. Conf.*, 2014.

[9] R. Roche et al., "Multi-Agent Systems for Grid Energy Management: A Short Review", *IEEE IECON*, pp. 3341-3346, 2010

[10] J. J. Gomez-Sanz et al., „Reviewing Micrigrids from a Multi-Agent Systems Perspective", *Energies*, vol. 7 Issue 5, pp. 3355-3382, 2014

[11] S. D. J. McArthur et al., "Multi-Agent Systems for Power Engineering Applications – Part I: Concepts, Approaches, and Technical Challenges" *IEEE Trans. Power Systems*, vol. 22, no. 4, pp.1743-1752, 2007

[12] M. Wooldridge, P. Ciancarini, "Agent-Oriented Software Engineering: The State of the Art", *Lecture Notes in Computer Science*, vol. 1957, pp. 1-28, 2002

[13] M. Merdan et al., "Multi-Agent System for Self-Optimizing Power Distribution Grids", *ICARA – Proceeding of the 5th Conference*, pp. 312-317, 2011

[14] A. Kantamneni et al., "Survey of multi-agent systems for microgrid control", *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 192-203, 2015

[15] B. Herd, "An agent-oriented programming approach for C++-based simulation models", *Technical Report TR-16-02, King's College London*, 2016

[16] A. H. Moin, "Sense-Deliberate-Act Cognitive Agents for Sense-Compute-Control Applications in the Internet of Things & Services", *LNICST*, vol. 150, pp. 23-28, 2015

[17] R. Rajesh, "Multi-Agent System Framework for Rural Indian Microgrids", *IEEE Conf. on Clean Energy and* Technology, pp.76-81, 2013

[18] R. Rajesh et al., "Implementation of Arduino-based Multi-Agent System for Rural Indian Microgrids", *IEEE Innov. Smart Grid Technol. – Asia*, pp.1-5, 2013

[19] M. Veeramani et al., "An Efficient Microgrid Management System for Rural Area using Arduino", *IJETT, vol. 40, no. 6,* pp. 335-341, 2016

[20] J. Betts, B. Müller, "A Component-based C++ Framework for Developing BDI Agents", *SAI Intelligent Systems Conf,* .2015

[21] B. Müller, J. Betts, "Incorporating BDI-Agent Concepts into Object-Oriented Programming", *CEUR Workshop Proceedings,* vol. 1269, 2014

[22] D. Kinny et al., "A Methodology and Modelling technique for Systems of BDI Agents". Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World (LNAI Volume 1038), pp. 56-71. Springer-Verlag: Berlin, Germany, 1996.

[23] F. Schönmann, "BDI-Architektur (Beliefs – Desires – Intentions)", *Seminar: Kognitive Modellierung animierter Agenten*, 2003