

TECHNISCHE UNIVERSITÄT MÜNCHEN
Forschungs- und Lehrereinheit XI
Angewandte Informatik / Kooperative Systeme

A Meta Model for Team Recommendations in Open Innovation Networks

Michele Brocco

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. Daniel Cremers
Prüfer der Dissertation:	1. Univ.-Prof. Dr. Johann Schlichter
	2. Univ.-Prof. Dr. Martin Bichler

Die Dissertation wurde am 28.03.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 06.07.2012 angenommen.

Abstract

Open Innovation has become an important new paradigm to integrate external knowledge into innovation processes of organizations. The size of innovator networks resulting from this brings in new challenges in context with management of work units. The composition of teams overarching organizational fields or tasks may serve as a useful example for that. For this purpose IT support can be applied, hence generating recommendations for suitable teams. This dissertation identifies related difficulties and thoroughly investigates aspects of team composition by the means of interdisciplinary literature research and of interviews with experts in this domain. Based on this analysis, a meta model is being conceptualized which makes it possible to instantiate team composition methods from the fields of practice, sociology and psychology. The meta model is being evaluated on the basis of several team composition methods and of three consecutive team recommender systems which draw on this meta model.

Abstract

Kurzfassung

Open Innovation ist ein wichtiges, neues Paradigma um externes Wissen in den Innovationsprozess von Organisationen einzugliedern. Die daraus resultierende Größe der Innovatorennetzwerke bringt einige neue Herausforderungen im Zusammenhang mit dem Management von Arbeitseinheiten mit sich. Ein Beispiel für diese Aufgaben ist die organisationsübergreifende Zusammensetzung von Teams. Zu diesem Zweck kann IT Unterstützung verwendet werden, welche Empfehlungen für geeignete Teams generiert. Diese Arbeit identifiziert die damit gekoppelten Schwierigkeiten und untersucht ausführlich Aspekte der Teamzusammensetzung durch eine interdisziplinäre Literaturrecherche und Interviews mit Experten der Domäne. Basierend auf diese Analyse, wird ein Meta Model konzipiert welches es ermöglicht, eine Vielzahl von Teamzusammensetzungsmethoden aus der Praxis, Soziologie und Psychologie zu instanzieren. Das Meta Model wird durch die Instanzierungen etlicher Teamzusammensetzungsmethoden und anhand dreier darauf aufbauender Team Recommender Systeme evaluiert.

Kurzfassung

Acknowledgments

This dissertation would not have been possible without the help and support of many people.

I would like to thank first and foremost my advisor, Prof. Dr. Johann Schlichter for giving me the opportunity to join his research group and providing all the precious feedback and for the insightful review of my thesis. Moreover, I would like to thank him for giving me the freedom and opportunity to work on this interesting topic. I am also grateful to Prof. Dr. Martin Bichler for being my second advisor, providing the great ideas and, last but not least, for helping me constantly within a very short time frame.

I would like to thank all our present and former group members that I worked with, for their numerous valuable and fruitful discussions, the great working climate and the legendary chats during lunch time and coffee breaks. They made me especially enjoy the time I spent at work.

I owe my deepest gratitude to Stephanie and Yi Li for the English proofreading of my thesis during their rare free time.

Credits go also to my brothers and family for their support and encouragement. It is needless to emphasize again, that all this would not have been possible without the support of my parents, who have given me the education opportunity that provided the foundation of this work.

Last but not least, my sincere appreciation goes to my girlfriend Mei for her patience, support and understanding during the writing of this thesis.

Acknowledgment

Contents

Abstract	3
Kurzfassung	5
Acknowledgment	7
Table of Contents	9
List of Figures	13
List of Tables	15
List of Abbreviations	17
1 Introduction	19
1.1 Problem Statement	20
1.2 Research Methodology	21
1.3 Contributions	22
1.4 Outline	23
2 Open Innovation	25
2.1 Open Innovation and Open Innovation Support	25
2.1.1 The Open Innovation paradigm	26
2.1.2 Open Innovation Platforms	28
2.1.3 KoPIWA	29
2.1.4 IT Support Systems for Open Innovation	31
2.2 Recommender Systems	38
2.2.1 Traditional Recommender Approaches	39
2.2.2 State of the art	44
2.2.3 Evaluation of Recommender Systems	55
3 Team Composition	59
3.1 Approaches from behavioral sciences	61
3.1.1 Empirical studies	62

3.1.2	Models of Team Behavior	69
3.1.3	Analytical frameworks	76
3.1.4	Summary	77
3.2	Approaches from practice	78
3.2.1	Interviews	78
3.2.2	Empirical Results	80
3.2.3	Summary	82
3.3	Algorithmic Approaches	83
3.3.1	Semantic-based	84
3.3.2	Social Network-based	84
3.3.3	Behavior-based	85
3.3.4	Ideal combination of attributes	86
3.3.5	Clustering-based	87
3.3.6	Latent semantic analysis	88
3.3.7	Summary	89
3.4	Summary, Analysis and Comparison of Approaches	90
4	A Meta Model for Team Recommender	95
4.1	Advantages of Team Recommendation Systems	96
4.2	Challenges of Algorithmic Recommendation of Teams	97
4.3	Aspect Entities of Team Composition	100
4.3.1	Team Member Variables	100
4.3.2	Output/Outcome Variables	101
4.3.3	Team Intermediate Variables	102
4.3.4	Team External Variables	103
4.3.5	Dependencies	106
4.3.6	Constraints	108
4.4	Categorization Analysis	109
4.5	Solution Approach	112
4.5.1	Generic Solution vs. Tailored Solution	114
4.5.2	A Meta Model for Team Recommendation	116
4.6	From Team Composition Aspect Entities to Meta Model Components	116
4.7	Meta Model Formalization	120
4.7.1	Formalization Summary	124
4.7.2	Meta Model Benefits	125
4.8	Modeling Team Composition Strategies	126
4.8.1	Pelled’s Model	126
4.8.2	Bouncken and Winkler Model	130
4.8.3	Mello and Ruckes Model	136

4.9	Summary	139
5	Application in Recommendation Algorithms	141
5.1	Constraint-based	142
5.1.1	Selected Meta Model Components	143
5.1.2	Recommendation Approach	144
5.1.3	Algorithm Summary	148
5.2	Case-based	152
5.2.1	Selected Meta Model Components	152
5.2.2	Recommendation Approach	154
5.2.3	Algorithm Summary	160
5.3	Model-based	164
5.3.1	Selected Meta Model Components	165
5.3.2	Recommendation Approach	167
5.3.3	Algorithm Summary	172
5.4	Summary	180
6	Evaluation	181
6.1	Evaluation Criteria and Approach	181
6.2	Expressiveness	183
6.3	Applicability (Evaluating Algorithms)	183
6.3.1	Constraint-based	184
6.3.2	Case-based	190
6.3.3	Model-based	197
6.3.4	Summary	204
6.4	Evaluating regarding Challenges	205
6.5	Discussion and overall Valuation	207
7	Conclusion	213
7.1	Summary	213
7.2	Extensions and future work	215
	Bibliography	217
A	Interviews	233
A.1	Outline	233
A.2	Statements	234

Contents

List of Figures

2.1	Open innovation funnel [Chesbrough et al., 2006]	27
2.2	Competence tag acquisition for the KoPIWA platform	34
2.3	Knowledge sources in recommender systems [Felfernig and Burke, 2008]	38
2.4	CBR is the process of solving new problems based on the solutions of similar past problems [Leake, 1996].	48
2.5		48
2.6	Case-based recommender approach example [Smyth, 2007]	49
3.1	Categorization of related work on team composition	60
3.2	Influences on creativity and innovation in MCITs [Bouncken and Winkler, 2008]	66
3.3	Hackman’s normative model of group effectiveness [Crowston et al., 2004]	69
3.4	Pelled’s model on demographic diversity and work group outcomes [Pelled, 1996]	75
3.5	Classification of related work based on the degree of formalization	92
4.1	Classification of team composition aspects	100
4.2	Vertical vs. horizontal aspects of team composition	111
4.3	Schematic illustration of a generic solution for a team recommender system	113
4.4	Team composition model for social factors (cf. [Brocco et al., 2010b])	118
4.5	Pelled’s model implemented by the meta model (dependency specifications omitted for clarity)	128
5.1	Specification of team composition core model and context (TEV) used	153
5.2	Competence taxonomy used in the example	161
5.3	Taxonomy of problem type for the example	162
5.4	Taxonomy of communication types in the example	162
5.5	Model-based recommendation process (Flow Chart).	168
5.6	Wizard interface used to configure the team recommender implemented	173
5.7	General settings for team recommender	173
5.8	Team composition strategy window	174
5.9	Configuration of analytic mappings in the user interface	175
5.10	Summary of configured parameters and progress bar for the computation of team recommendations	175
5.11	Window with team recommendations ranked by output scores	176

List of Figures

6.1	General settings of the constraint-based team recommender	187
6.2	Tab for setting the requirements for specific job positions	188
6.3	Result window with team recommendations and data on past projects of the teams if available	189
6.4	Medians of testruns for constraint-based team recommender	190
6.5	Communication media taxonomy constructed by the expert and used in the recommendation approach	192
6.6	Taxonomy of creativity techniques provided by the expert	193
6.7	Front end of the IdeaStream platform	197
6.8	Geographical distribution of experiment participants	199
6.9	Performance of some approaches with a small amount of training data. . .	203

List of Tables

2.1	Contrasting principles of Closed and Open Innovation [Chesbrough, 2003]	28
4.1	Advantages of possible team recommendation approaches	115
5.1	Example set of team member candidates	151
5.2	Best teams and evaluated analytic mappings	151
5.3	Example of features configuration for <i>computer science</i> teams.	161
5.4	Result of similarity assessment for the given query.	164
5.5	Instantiated team composition models and section of occurrence	180
6.1	Features configuration for the experiment.	194
6.2	Evaluation result: comparison of ranks generated by domain-expert and by the case-based recommender	195
6.3	Result of 10-fold cross-validation for predicting the <i>minimum average of creativity</i>	201

List of Tables

List of Abbreviations

BDI	Belief-desire-intentions
CBR	Case-based reasoning
CSCW	Computer supported cooperative work
CSP	Constraint satisfaction problem
CSS	Creativity support system
CV	Curriculum vitae
HR	Human resources
ICT	Information and communication technology
IP	Intellectual property
IQV	Index of qualitative variation
IS	Information systems
IT	Information technology
KoPIWA	Kompetenzentwicklung und Prozessunterstützung in Open Innovation -Netzwerken der IT-Branche durch Wissensmodellierung und Analyse
MCIT	Multicultural innovation teams
OB	Organizational behavior
OI	Open innovation
POI	Point of interest
R&D	Research and development
SME	Small and medium enterprises
SNA	Social network analysis
TCM	Team composition model
TEV	Team external variables
TMV	Team member variables

List of Abbreviations

1. Introduction

Teams are probably the most important organizational unit nowadays. As opposed to single individuals, they have access to a wider range of knowledge and information and gain synergy advantages through collaboration and communication.

The extent of these effects depends on the properties of a team such as its size or the skills of the team members. Many definitions for the term “team” have emerged over the last decades. Kozlowski and Ilgen [Kozlowski and Ilgen, 2006] for example define a team as:

- Two or more individuals who
- Socially interact (face-to-face or, increasingly, virtually);
- Possess one or more common goals;
- Are brought together to perform organizationally relevant tasks;
- Exhibit interdependencies with respect to workflow, goals, and outcomes;
- Have different roles and responsibilities; and
- Are together embedded in an encompassing organizational system, with boundaries and linkages to the broader system context and task environment.

How well a team performs depends on many factors: Firstly, a team’s output depends on the team members that are part of the team itself. Secondly, external factors influence the team when performing a task. Therefore the composition of teams and the support of a team’s context have been focussed by socio-psychological research as well as economic sciences for many decades.

The strategy of composing a team depends on the experience, goal and possibilities of the person in charge of composing the team. A successful soccer team, for instance, can only be assembled if the management has enough money to purchase outstanding soccer players and if the coach is able to comprehend which eleven players are suitable for a

1. Introduction

specific tactic, match and/or opponent.

This task becomes even more difficult when the players are not known, i. e. if a team has to be built from scratch or the coach just joined the soccer club. In this case, the coach would have to gather information from different sources, in order to characterize the players and to find out which players are appropriate for a specific match. The respective acquisition problem, as well as the composition task itself are very time-consuming and can easily become complex.

Similar complications can be found in a seemingly very different domain, namely open innovation networks. These consist of actors (organizations, freelancers, customers or other individuals) that want to collaborate on innovations by opening arbitrary phases of their innovation process. These arising collaboration networks are called *open innovation networks*. In these networks, people often do not know each other and are thus not capable of effectively grouping individuals together to form a team. This has to be added to the fact that the large number of possible team members and influencing factors that need to be taken into consideration are not easy to manage for a human being. Thus a support system that processes the necessary information can help facilitate the decision making process when a team is to be composed.

1.1. Problem Statement

With the above mentioned main challenges for team composition in mind, this dissertation is intended to make a contribution to finding possible algorithmic support systems to the team composition problem in the open innovation domain. A solution to this problem can make collaboration across organizational units easier than it currently is, since some of the barriers of assembling inter-organizational teams can be reduced. This is especially important because the success of the open innovation paradigm is heavily dependent on the initiation of new cooperations. Hence, any contribution in this direction must be considered as valuable.

Currently, project initiators (e. g. product managers, project managers) strongly rely on contact persons in other companies (brokers) that either assign team members for open innovation projects directly, or forward the request to other persons. The composition is thereby based on the broker's preferences and opinions instead of the project initiator's preferences. The brokering mechanism is very time consuming and costly in terms of transaction costs. Moreover, this process is also disadvantageous because in certain cases it requires to reveal the project idea to the broker.

In general, recommender systems can be used to filter relevant items from a large pool of possible items. In this case, they can be used by the project initiator to compose a team in a more unbiased way and see which partners in the open innovation network are suitable for a cooperation. Furthermore, the recommendation of a team through a recommender system is faster and less expensive than the brokering process, since it can be reiterated whenever the requirements (and hence the composition strategies) change. Given an open innovation network in which possible collaborators expose relevant data to the open innovation community a support system is therefore advantageous. In order to address the outlined issue, the research question of “*How can effective teams be algorithmically composed in open innovation networks?*” has to be investigated.

1.2. Research Methodology

The analysis of the team composition (and accordingly team recommendation) problem involves several steps.

First, the domain of open innovation networks has to be comprehensively analyzed. Then, existing paradigms for generating recommendations and the domain of team composition have to be thoroughly reviewed. The results of these reviews have to be considered when developing a team recommendation system. The completion of this work package is primarily done via literature review and interviews with domain experts in the field of team composition. At the same time, this analysis defines the requirements for a valid team recommendation approach in an analytical as well as empirical manner.

Afterwards, the team recommendation problem is characterized and the previously introduced existing approaches evaluated through the above-mentioned in-depth review of the team composition domain.

According to the conclusions made within this characterization, an approach that regards the previously defined requirements is developed. For a more detailed evaluation, prototypes that demonstrate the feasibility of the proposed approach on the one hand and the compliance with the gathered requirements on the other are implemented. These pieces of information are then used to summarize the advantages and drawbacks of the developed approach and to inspire future research.

In this spirit, this paper follows a design science approach. To be more precise, it satisfies the seven guidelines for design science in information systems research provided by Hevner et al. [Hevner et al., 2004] as follows:

1. Introduction

Guideline 1 (Design as an Artifact): This work contains the description of several artifacts that were created: a categorization of important aspect entities for team composition, methods to operationalize these aspect entities, a model mapping these aspects and several algorithms working on this model.

Guideline 2 (Problem Relevance): The relevance of the investigated problem is demonstrated through the presence of a vast array of related work in this field and empirical evidence from interviews with domain experts.

Guideline 3 (Design Evaluation): In order to show the functionality of the approach appropriate evaluation methods are discussed and applied such that a discussion of the developed solution can be provided.

Guideline 4 (Research Contributions): Novel contributions are generated within this work (refer to Section 1.3)

Guideline 5 (Research Rigor): This approach is designed based on related work and empirical evidence. Furthermore it is evaluated through statistical methods and quality measures derived from the recommender system research where applicable and with qualitative evaluations otherwise. In this way, this piece of work also provides a discussion for appropriate evaluation methodologies to the problem of algorithmically recommending teams.

Guideline 6 (Design as a Search Process): The review of work in the problem domain (open innovation) as well as the solution domain (recommender systems) addresses the guideline that interprets design science as a search process for finding appropriate solutions. Despite the fact that not all iteratively produced artifacts could be included in this thesis, this work provides descriptions and evaluations of the most valuable artifacts generated.

Guideline 7 (Communication of Research): The major contributions contained in this work have been published in the academic (e.g. [Brocco and Groh, 2009], [Brocco et al., 2010a], [Brocco et al., 2011b], [Brocco and Asikin, 2011]) as well as in the practitioner’s community (e.g. [Brocco and Groh, 2010]).

1.3. Contributions

This thesis on the analysis of appropriate team recommendation systems for the open innovation domain provides several main contributions:

- A comprehensive and interdisciplinary review and discussion of related work in the team composition field
- Interviews with experts of the team composition domain in the ICT¹ sector
- A discussion on important aspects of team composition
- An analysis of the team recommendation problem
- Several team recommendation algorithms that can be applied to recommend teams
- A quantitative evaluation of a team recommender system based on evaluation metrics popular in the recommender systems domain
- Findings that provide guidelines for future work in this field of research

1.4. Outline

This thesis is composed of seven chapters:

This chapter presents a summary of the motivation and the main research question of this thesis and illustrates the research design and the main contributions of this paper.

Chapter 2 more thoroughly describes the problem domain. To be more precise, the open innovation paradigm is introduced and discussed. At the same time this chapter constitutes why and when IT support is beneficial for supporting several activities performed during collaboration in open innovation processes. Team recommendation is clearly one of these IT support mechanisms. For this reason, the second part of this chapter provides the foundations of recommender systems research making it possible to understand why current recommendation approaches are not easily adaptable to the team recommendation case.

Chapter 3 presents an interdisciplinary literature review on team composition. This review shows how teams are composed nowadays and what factors influence the performance of a team. This is necessary to understand which aspects have to be regarded when recommending teams. This chapter also gives an overview of existing algorithmic approaches for composing teams. A discussion on the various approaches addressing team composition concludes this chapter.

Chapter 4 deals with the problem of team recommendation. Challenges and pecu-

¹Information and communication technology

1. Introduction

liarities of team recommendation are presented. On the basis of these analyzes and on the findings of Chapter 3, a generic approach for team recommendation is proposed. The functionality of this generic approach is shown in several examples

Chapter 5 shows the usage of this generic approach denoted as *meta model for team recommendation* in the domain of recommendation systems. To be more precise, three team recommender algorithms that employ instances of the meta model are developed as contribution of this dissertation. Thereby, the applicability and functionality of the approach is more clearly demonstrated.

Chapter 6 first discusses how the proposed approach can be evaluated. Several criteria that can be used to evaluate the proposed recommendation approach are elaborated on. Given these results, the approach and the algorithms are evaluated in several manners, in order to provide various indications of the benefits and drawbacks of this approach.

Chapter 7 concludes this paper by summarizing the critical aspects of this thesis and by providing guidelines for the utilization of the proposed team recommendation approach. The last section of this work uses the evaluation provided in Chapter 6 to indicate what the related research questions and improvements that can be addressed in future work are.

2. Open Innovation

This chapter provides insight into the world of open innovation, IT support for open innovation and recommender systems. The first section defines the open innovation paradigm and depicts the advantages of opening the innovation process. It also describes the application domain of the team recommendation approach and introduces a popular environment for collaborations in open innovation networks, namely open innovation platforms. In addition various IT support mechanisms for open innovation are described. The second section gives a general overview of recommender systems and presents a categorization of existing recommender approaches that are used to filter items the same way as teams may be filtered out from a whole range of possible teams in the open innovation scenario.

2.1. Open Innovation and Open Innovation Support

Open innovation is a new paradigm that fosters the cooperation of actors in an innovation process by opening arbitrary phases of the innovation process to other innovators outside an organization's boundary. In the following, the open innovation approach is introduced in more detail through aggregation of several articles from the economic sciences. Community (web-) platforms for open innovation are discussed as a means of bringing innovators together. In relation to that, the KoPIWA project, whose primary goal is competence management and fostering collaboration in open innovation networks is introduced. Platforms such as the KoPIWA platform, can include services and tools that support users in the different phases of the open innovation process. Among other things these services can support the distributed generation of ideas or provide several awareness services. They also offer support when looking for information or people in the innovator network. The section "IT Support Systems" gives a brief overview of such support systems by describing some examples.

2.1.1. The Open Innovation paradigm

Nowadays innovation is a crucial success factor for companies and organizations. Despite this, it has not been possible for companies to fully unfold its potential throughout the recent years. Henry Chesbrough comprehensively described the open innovation paradigm from various perspectives ([Chesbrough, 2003],[Chesbrough et al., 2006]). He compares several aspects of open innovation to the closed innovation approach.

Chesbrough summarizes the closed innovation process as an innovation process where “... *companies must generate their own ideas and then develop them, build them, market them, distribute them, service them, finance them, and support them on their own*”. Based on his research he suggests that effective innovation in certain markets and fields has to follow different guidelines.

In the last decades many firms that were applying closed innovation either lost significant impact on the market or great opportunities to further develop. Chesbrough mentions many such examples including Lucene, Xerox (Palo Alto Research Center PARC), IBM, Siemens and many others. According to him, such companies were not able to recognize a fundamental change in the way innovation was created. These companies continued investing in research instead of scanning the world of start-up companies. Start-ups were candidates for an acquisition or provided valuable research results and knowledge that could have either been licensed or purchased.

An impressive example for closed innovation and its possible consequences is Xerox PARC. Some of the ideas and technologies developed at the PARC research lab that were rejected as not valuable by Xerox, later became very valuable in newly created enterprises (e.g. 3Com, Adobe, Apple, Microsoft, etc.). In the meantime the role of Xerox in high-technology markets has decreased drastically compared to the 1980s.

The reason why the closed innovation paradigm in a certain environment loses its edge lies in so-called “erosion factors”. The growing mobility of highly experienced and skilled people - one of these erosion factors - allows for new and smaller companies to employ well-trained staff from big companies without having to pay any compensation for their training and skills acquired during their time in the large company. By means of these newly gained human resources, smaller companies are able to innovate at nearly the same level as large firms because they save on high training costs.

Another erosion factor is the growing presence of venture capital which facilitates ideas and research to be converted into valuable new companies.

In addition, Chesbrough mentions the increased significance of the fast time to market

for many products and services as a reason for the decline of closed innovation: expensively acquired knowledge on a specific technology cannot be exploited as much as it was possible in the past. More knowledgeable customers and suppliers have also been an “issue” ever since they started requiring new innovations and consequently new effective competitors emerged that could satisfy their needs.

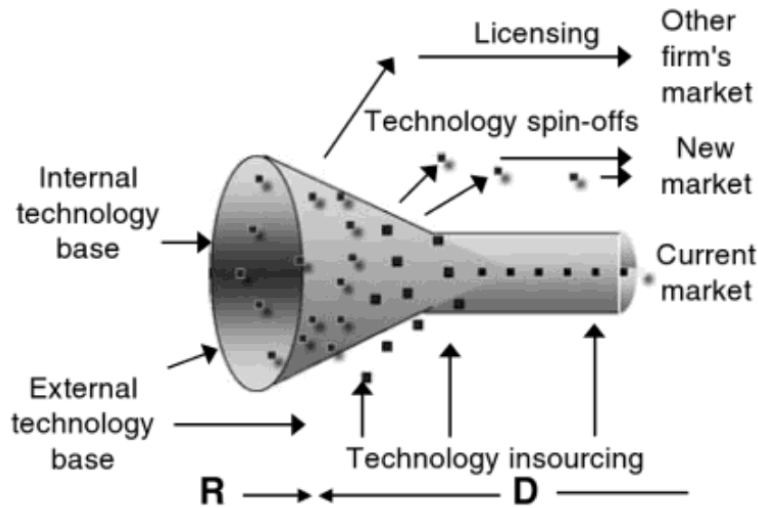


Figure 2.1.: Open innovation funnel [Chesbrough et al., 2006]

Whenever these circumstances apply, closed innovation assumptions are no longer valid¹. In order to address these new circumstances, open innovation approaches are emerging. Their main goal consists in combining internal **and external** ideas and sources to develop new products. Furthermore, internal **as well as external** paths to market can be chosen for services or product development. That is, innovations can also be taken to market by external channels, not only by a company itself. Examples for such cases are start-up companies or the sale of licenses for ideas. Similarly, ideas can also be acquired from outside (customers, other organizations, freelancers, etc.) and brought to market as own products (Figure 2.1). Table 2.1 provides a comparison of the open vs. closed innovation principles.

The main advantage of the open innovation approach resides in not “wasting” ideas and innovations that do not seem valuable at the time by creating spin-offs by selling technology in order to rapidly diffuse. Closing intellectual property (IP for short) is no longer profitable from a long-term perspective, since competitors often find ways to invent around a company’s IP. Having said that, timing is crucial: opening too early may lead to losing potential profit, opening too late may lead to losses of respective market shares and

¹for more details on these assumptions see [Chesbrough, 2003]

2. Open Innovation

facing stronger competitors that have acquired the necessary knowledge in the meantime [Harhoff et al., 2003].

Closed Innovation Principles	Open Innovation Principles
The smart people in our field work for us.	Not all the smart people work for us. We need to work with smart people inside <i>and</i> outside our company.
To profit from R&D, we must discover it, develop it, and ship it ourselves.	External R&D can create significant value; internal R&D is needed to claim some portion of that value.
If we discover it ourselves, we will get it to market first.	We don't have to originate the research to profit from it.
The company that gets an innovation to market first will win.	Building a better business model is better than getting into market first.
If we create the most and the best ideas in the industry, wins.	If we make the best use of internal and external ideas, we will win.
We should control our IP, so that our competitors don't profit from our ideas.	We should profit from others' use of our IP, and we should buy others' IP whenever it advances our own business model.

Table 2.1.: Contrasting principles of Closed and Open Innovation [Chesbrough, 2003]

2.1.2. Open Innovation Platforms

When opening and using knowledge and ideas of other people and organizations, new collaboration networks arise: open innovation networks. There are several advantages of innovator networks: First of all, the networks' interconnections allow finding previously inaccessible knowledge. Apart from this, the network as a set of actors itself, represents wide knowledge, skills and ideas that can be employed when needed [Powell and Grodal, 2005]. The actors in such networks can be customers, users [von Hippel, 2007], companies or freelancers that interact, in order to create innovative products or services.

As stated in Powell and Grodal's work, the success of innovation networks (and hence also **open** innovation networks) is heavily dependent on “[...] *the ability to convey and transfer ideas that are not easily codified*”. In order to be able to communicate and exchange information on the innovation process and the ideas, **collaboration platforms** are usually used. Due to various specific instantiations of the open innovation process, tailored support is required. For this reason, IT platforms (most frequently implemented as web-platforms) are used as an infrastructure for open innovation networks.

Possible instantiations of the open innovation process may vary in terms of the specific innovation phases that are opened to other actors and collaboration, the selected members in the network, or the domain of innovation. Examples for the variety of these platforms can be seen for instance at <http://www.openinnovators.net/list-open-innovation-crowdsourcing-examples/>. The categorization illustrates that each platform has slightly different application domains or phases of the innovation that are supported: idea generation, innovation development, kick-off of collaborations etc.

The presence of such a variety of open innovation platforms indicates that platforms are a common tool for supporting open innovation and, hence, the network of innovators taking part in the innovation process.

Powell and Grodal's work and the amount of web-platforms indicate that IT platforms are very important to support such innovator networks. This is also emphasized by the number of projects funded by the German Federal Ministry of Education and Research [Fokusgruppe Management offener Innovationsprozesse, 2010] that aims at developing appropriate platforms. Consoli and Patrucco also describe the importance of IT platforms under several aspects. They emphasize the advantages in terms of knowledge sharing, coordination and communication and describe when and where their application is advantageous in the case of distributed innovation ([Consoli and Patrucco, 2007]) such as in open innovation networks.

2.1.3. KoPIWA

One of the projects funded by the German Federal Ministry of Education and Research, is the KoPIWA² project [Zeini et al., 2008]. The project partners are small and medium enterprises (short SMEs) of the digital economy sector in Germany, two universities (the Technical University Munich and the University of Duisburg-Essen) and an innovation research institute from the Ruhr area. The project had five main goals:

1. Management of interactive partnerships in value creation: how can companies interact through partnership in order to create innovation? Which problems have to be considered?
2. Knowledge and human resource management for (open) innovation
3. Innovation in the "dark field"

²Kompetenzentwicklung und Prozessunterstützung in Open Innovation-Netzwerken der IT-Branche durch Wissensmodellierung und Analyse

2. *Open Innovation*

4. Open Innovation in SME

5. Software-based model for a competence fostering innovation strategy

Regarding the first goal, several case studies with the SMEs have been conducted, in order to analyze the open innovation process with respect to its properties and challenges. The results provide the basis for a platform whose goal is to motivate actors in the BVDW (Bundesverband Digitale Wirtschaft) community to innovate in their respective community.

Through an appropriate learning environment, innovators in the network should be capable of acquiring new knowledge needed to create innovations. Thus, exchanging knowledge from inside the company to the innovator network and vice versa plays an important role. Furthermore, this goal is also achieved through capturing and identifying trends that motivate actors in the network to improve their skills.

Innovation in the “dark field” is defined as specific tasks and aspects in the innovation process which cannot be directly perceived from the outside. For instance, informal acquisition of new skills by individuals in an organization is rarely documented and therefore cannot be effectively used for competency development because of the difficulty to associate the new skill with the person that acquired it. With respect to this goal, the project aims at finding approaches to make currently invisible work and skills (including tacit knowledge) visible [Attwell, 2007].

Current research on open innovation (cf. [Hoppe, 2006]) has discussed that SMEs in general tend to have fewer cooperations with external partners than large companies. Especially in the field of digital economy it has not yet been possible to find any evidence for existing cooperations. In order to foster cooperations community-networks are induced through case studies with partners that are currently using Open Source-like business models. On the basis of social network analysis, different innovator roles are identified and their behavior and importance in the network is characterized, to develop appropriate mechanism for supporting and motivating the respective roles.

With respect to the last goal, one of the outputs of the KoPIWA project is a community platform that implements the theoretical results achieved while pursuing the above-mentioned goals. Hence, the resulting technical infrastructure comprises specific open innovation tailored services apart from the state-of-the-art collaboration tools such as mailing lists, blogs or forums.

2.1.4. IT Support Systems for Open Innovation

As with the KoPIWA platform, most of the open innovation platforms use IT to support their innovation process. Depending on their specific open innovation approach (e.g. their open innovation process) and their focus, different services and tools are applied. This section presents some of the IT support tools used and describes the advantages of their application.

Each tool and service that supports the innovation process has its own goal. IT support in general is needed when the various steps of a task whose objective is innovation become complex or challenging as it is the case in the team composition/recommendation scenario. In open innovation networks, the complexity increases mainly due to the heterogeneity and number of actors involved in the network, as well as the amount of information generated by the open innovation community. Another factor making effective collaboration more difficult is that the actors are mostly distributed and not co-located [Wong and Burton, 2000]. In the following sections some examples for IT support tools are illustrated together with their respective goals. Included are generic collaboration tools, creativity support systems, competence management systems, awareness support systems and expert finders.

Generic Collaboration Tools

As the most popular approach to implement open innovation platforms are web 2.0 community platforms, state-of-the-art collaboration tools for virtual web 2.0 communities are often applied. This concerns wikis, blogs, forums, tagging and rating systems, etc. One reason for this is that much research on this type of collaboration media has already been conducted. To name merely two of numerous examples that can be found in relevant literature, Moore and Serva analyzed which motivational factors play a role when contributing on forums, blogs, wikis and knowledge bases such as tagging frameworks [Moore and Serva, 2007] while Holtzblatt et al. [Holtzblatt et al., 2010] analyzed factors impeding wiki use in the enterprise.

Moreover the tools are very generic so that they can be employed for collaboration in many phases of the open innovation process. This can include the idea generation phase, which can be done through posting forum threads, as well as through developing an idea on a wiki page.

Though flexibility is the most obvious advantage of these tools a too flexible tool does not provide tailored support so that some semantic is lost. If, for example, a forum is used

2. Open Innovation

for the purpose of generating ideas, it becomes difficult to distinguish ideas from comments and hints within a thread. In this case, additional services and features of idea generation on top of those generic tools may be very complex to develop as they would need a more precise elaboration, such as content analysis, natural language processing, etc.

Another example would be the use of a wiki as a project management tool in an open innovation platform. Here too, it is obvious that certain particular aspects of project management may not be regarded by this collaboration tool.

Nevertheless, generic collaboration tools are used in nearly every open innovation platform, because there is a high probability that every open innovation community member is familiar with this kind of communication media. Furthermore, based on the therewith generated content, some valuable data can be extracted e. g. through social network analysis (SNA).

Creativity Support Systems

As described in the above section, idea generation is one of the most important phases in the innovation process (and hence also in open innovation) since “*all innovation begins with creative ideas.*” [Amabile et al., 1996]. Computer supported creativity is valuable especially for the management of ideas, the guidance it provides during creative processes and the support of specific contexts in which ideas are generated. Regarding this, Rhodes has developed a model for creativity that consists of the four aspects *creative persons*, *creative products*, *creative processes* and *creative press* (or context) [Rhodes, 1961].

Most creativity support nowadays focuses on the dimension process. This includes for example tools for creativity techniques such as brainstorming or mind mapping with a tailored software. “Tailored” in this context denotes the fact that characteristic elements of creative processes are modeled in the software and can thus be acquired. In the case of creative processes, this includes ideas, ratings, contributors, the creative process with its phases etc. Some of the creativity support systems (CSS) provide tailored support with respect to creativity techniques [Forster and Wartig, 2009]. Other tools are used for creative problem solving, even if they do not provide specific structuring for creative processes. Plain virtual whiteboards are an example for such tools, i.e. they do not explicitly model a creative process, but can still be used to generate ideas. Some other solutions allow distributed collaboration over the web such as Bubbl.us (cf. www.bubbl.us), Mindmeister (cf. www.mindmeister.de), ThinkTank 2.0 (cf. www.groupsystems.com) and others [Forster, 2008]. Distributed creative problem solving may be especially interesting for the open innovation scenario since actors involved in the innovation process

are often not co-located.

Regarding Rhode's defined dimension press, (i. e. physical external factors that characterize the context of creative problem solving) not much research on IT support has been conducted so far, despite the evidence that these factors significantly influence the outcomes of creative processes. For instance, only very few tools for creativity support explicitly take the context of idea generation, such as the spatial or temporal distribution of team members into account. Therefore, approaches going towards a context-aware IT support for creativity could consist of user interface adaption depending on location and time (e. g. mobile or multi-touch interfaces) [Frieß et al., 2012]. Hence, aspects of press in relation to IT-support are yet to be investigated more thoroughly. This includes interrelational aspects between individuals in a team. These "press" aspects also overlap with the "person" dimension described by Rhodes so that they generally cannot be treated separately. The team composition problem for teams taking part in creative problem solving processes can therefore be treated as a problem related to the dimension "person" **and** "press" in the context of creative problem solving tasks.

Competence Management

Managing competencies in open innovation networks is crucial for reacting to trends and thus create innovations. Being in the field of innovation itself also requires having special competencies. Whenever a new technology arises, competencies in that field are required to create innovations based on this new technology. Since, as mentioned in Section 2.1.2, innovation and communication in open innovation networks is manifested in open innovation platforms, IT can be used to collect competence information and to build corresponding services for competence management and development.

Competencies can be defined within the framework of two different categories: self-perceived competencies and competencies perceived from the outside. However it is also possible to create a third category that represents the "real" competencies. Their assessment is highly demanding. One possible way would be to extract pieces of information generated by the user in a quantitative manner and to use text mining and natural language processing tools. These methods however rely on a rich knowledge base.

Several approaches can be applied to define self-perceived competencies. Probably the most common approach is to enter corresponding meta data into its own profile. Furthermore, competence extraction from CV (curriculum vitae) data can be used.

Competencies perceived from the outside can be gathered from project managers', colleagues' or other people's assessments. In the KoPIWA project, for example, specific

2. Open Innovation

The image shows a user interface for adding tags and competence tags. It consists of two main sections: 'Tags' and 'Competence Tags'. Each section has a text input field, an 'Add Tags' button, and 'Or Select Tags Suggestions' buttons. The 'Tags' section has a blue box around the input field containing the text 'etwas x'. The 'Competence Tags' section has a yellow box around the input field containing the text 'schreiben x'. At the bottom of the interface are 'Save' and 'Cancel' buttons.

Figure 2.2.: Competence tag acquisition for the KoPIWA platform

competence tags attached to contributions (forum posts, wiki articles, blog entries, etc.) by members of the open innovation community are captured. In this way, manifested competencies can be acquired (Figure 2.2).

Assessing competencies in this informal way tends to be dissatisfactory. Competence tags, as tags in general, are unstructured by nature and are affected by several problems of classical information retrieval such as spelling mistakes, special characters, meta-noise or different languages [Manning et al., 2008]. Besides, the unstructured nature of tags complicates the development of services based on relationships between tags, and hence, which is the case here, competencies. Comparing unstructured competence tags such as “Java mobile” and “Java” devoid of any structure will not produce any result and thus indicate no similarity.

Therefore, structuring mechanisms can be applied to achieve an arbitrary expressiveness. These mechanisms can be entirely manual, semi-automatic or automatic as shown by Kammergruber et al. [Kammergruber et al., 2010].

The KoPIWA project uses two different approaches for competence extraction and structuring. The first approach extracts competencies from job application websites and semi-automatically inserts them in a competence ontology which is then, in a second stage, manually reviewed [Ziebarth et al., 2009]. As a second approach, the KoPIWA platform provides tools to collaboratively structure competence tags (and also ordinary tags) inside the open innovation community so that expressiveness at thesaurus level can be achieved. This comprises the relations synonym, homonym, acronym and hypo-/hypernyms between competencies.

The first mentioned method generates a large dataset which is advantageous: With the help of these datasets new knowledge on the relationships between competencies can be gained. Having a rather complete ontology and hence a common understanding for

competence relationships available within the open innovation community requires less additional effort from the community itself. The maintenance of the ontology, however, is very time consuming so that it is not suitable for rapidly changing competence branches.

The second approach has the advantage that changes can effectively be operated by authorized community members. The structure can immediately be changed and applied wherever necessary. This approach particularly supports the assumption of having no fixed point in the evolution of a competence ontology and that the structure constantly changes especially in the domain of innovations.

The acquisition of competencies in open innovation platforms is very important for the use of expert finder systems or team recommender systems because they often play a major role in the selection of appropriate work partners or team mates (as is elucidated in Section 3.2). Once competencies are acquired and structured, a team recommendation engine can more effectively find people with the required competence portfolio by taking similar or related competencies into account.

Awareness Support Systems

Due to the large amount of users, activities, sites and services available in open innovation networks, community members do not have a comprehensive overview of the dynamics of the open innovation network. They are usually only *aware* of particular parts and contributions generated in the community. Missing awareness has negative effects on the motivation of community members and often complicates the retrieval of relevant information [Schlichter et al., 1998].

Greenberg and Gutwin define four types of awareness [Greenberg and Gutwin, 1996]:

1. *Informal Awareness*: Information about co-workers, their working domain and their availability
2. *Social Awareness*: Information about someone's emotional state, his interests etc.
3. *Awareness over group structure*: Information about roles and responsibilities of individuals in a group
4. *Workspace Awareness*: Information about the cooperation with other group members. That includes e.g. information about interaction with other users, their competencies, activities and intentions

Awareness plays a key role in open innovation networks because members have to

2. Open Innovation

become aware of new ideas, trends and new contributions to specific topics. Moreover, when contributing to a specific open innovation project, the effort invested into it should be made visible to the community by making it possible to clearly assess who provided which input. On this basis, a contribution accounting system can be implemented that can be used to specify the distribution of intellectual property over a specific common open innovation project. All these mentioned aspects are crucial for an effective collaboration between actors (organizations, freelancers, customers) in an open innovation network.

The contributions and their quality in an open innovation community can be aggregated to serve as input for more complex services that provide information on a higher level of abstraction. One of these systems has been developed during the KoPIWA project and is called “Fame Mirror”. The Fame Mirror’s goal is to make a community member’s fame visible to the community by acknowledging his or her contributions and the quality of these contributions. In this manner, perceived members’ competencies are assessed. In addition to written contributions member activity in the community is also taken into account. By taking these aspects into consideration three different flavors of “Fame” measures were developed: unweighted fame, weighted fame and universal fame. *Unweighted fame* accounts only for the rated contributions of a community member. The *weighted fame* of a community member consists of the unweighted fame, weighted by the raters’ fame. *Universal fame* represents the weighted fame and adds other types of user activity in the community such as log-ins, member invitations, rating activities, etc. For further details on the Fame Mirror in the KoPIWA project refer to the corresponding article of Groh et al. ([Groh et al., 2010]). Besides addressing the goal of awareness, the fame mirror system could be used for other services such as a criterion for effective team composition in an open innovation platform.

Expert Finders

Finding individuals with specific skills in a community as large as an open innovation community consisting of e. g. three large scale companies, is a time-consuming task if done manually. In this case expert finder systems may support people in their search.

Probably the easiest way to implement such a system is by formulating appropriate queries in a user database containing, for example, the needed skills. This however does not always yield good results. For a start, formulating the query itself is not always straightforward. The same information retrieval problems occur as described in the “Competence Management” subsection. Secondly, this may not always be the best strategy for looking for someone that is to provide answers to specific questions.

One way of developing an expert finder in a large company is to map the company's current best practices to a corresponding system. This approach has been followed by McDonald and Ackermann [McDonald and Ackerman, 2000]. In contrast to content-based expert finders, such as the system proposed by Streeter and Lochbaum [Streeter and Lochbaum, 1988], they try to incorporate two different types of heuristics for locating experts in the expert finding process within a medium-sized medical software company with 100 employees. These two heuristics are based on the change history of software products and on the solution of similar tasks in the tech support area respectively. Through participant observation and interviews, they identified three independent stages for expert finding: identification, expertise selection and escalation. Expertise identification is defined by McDonald as the problem of finding a set of candidates who are likely to have the desired expertise. Expertise *identification* relies on prior experience with others, key people like an expertise concierge, and historical artifacts. Expertise *selection* is the process of picking an expert among the found persons. This step is influenced by aspects such as organizational criteria, availability or social components. Whenever breakdowns in finding appropriate contact persons through identification and selection occur, these two processes are reiterated. Thus, the problem is diffused in the expert network and implicates a growth of activity in the community. This reiteration is denoted as *escalation*.

The system assesses expertise by generating a profile out of daily work products instead of using explicit feedback on people. The approach is evaluated in another work of McDonald ([McDonald, 2001]) where people were asked to explicitly rate the recommendations of the expert finder. It turned out that recommendation based on a profile computed from the change history of a software products was more valuable than heuristic accounting of similar problem solutions.

Apart from that, a very high correlation has been found for the assessment of user skills by other users in this company, so that this data could be used to evaluate this system.

Related work on expert finders (e. g. [Streeter and Lochbaum, 1988], [Ru et al., 2008], [Reichling and Wulf, 2009]) shows how complex the task of finding appropriate persons in a large organizations or enterprises is. Several aspects have to be modeled and considered also depending on the specific implementation environment. If locating single people is a challenging task whenever the number of possible persons increases, it can be stated that finding appropriate teams is at least as challenging. Furthermore, when the interrelations between team members have to be considered, the task of finding and composing an appropriate team for a task becomes even more complex.

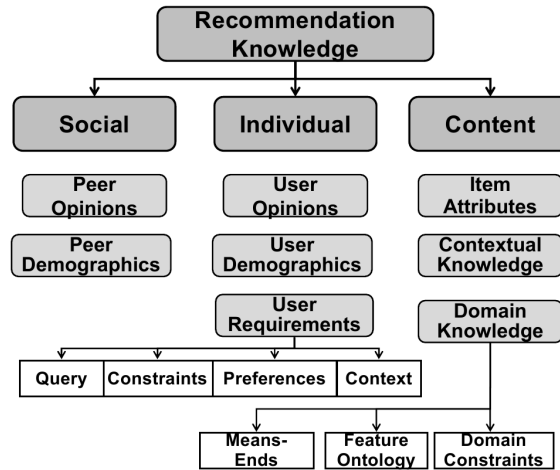


Figure 2.3.: Knowledge sources in recommender systems [Felfernig and Burke, 2008]

2.2. Recommender Systems

In the current era of overflowing data, information filtering is crucial. Finding appropriate information among a large set of possibly valuable items is time-consuming and requires a great deal of effort. Recommender systems address the issue of information overload and try to predict a user's preference with respect to specific items. On a more general level, they try to recommend items out of a set I to a user $u \in U$, where U is the set of all users in the system. In that sense, the presented team composition problem of this thesis, is strictly related to the recommendation problem: a set of 200 persons in an open innovation network theoretically gives $2^{200} - 200$ teams (items) among which the user u has to choose a favorite team. Moreover, recommender systems may be applied also for other tasks such as finding appropriate information items in open innovation platforms (Section 2.1.2).

Recommender systems can base their computations on several different kinds of data. Felfernig and Burke ([Felfernig and Burke, 2008]) distinguish between social, individual and content knowledge (depicted in Figure 2.3). When designing a recommender system, it has to be decided on which knowledge source the system is to draw on. This also applies to the team recommender system. That is why this thesis investigates also appropriate knowledge sources.

Choosing or designing a recommendation algorithm that uses the data provided by the knowledge source to generate recommendations is another step required for the development of a recommender. Selecting the recommendation algorithm is very important and

depends on the requirements of the recommendation task. In domains in which feedback is rare, knowledge-based recommender systems may for instance be more suitable than recommenders relying on learning techniques.

The following sections concentrate on this second component of a recommender system, namely the recommendation algorithm. An overview of the traditional recommendation techniques and their enhancements shall help to more clearly understand the functionality of recommender systems in general.

The first part of this section introduces the most common recommendation approaches to convey the basics of recommender systems. In the second part, some state-of-the-art approaches are presented to give an overview of current research on recommender systems. The insight provided by these sections provides the background information necessary for developing appropriate team recommendation systems.

2.2.1. Traditional Recommender Approaches

Recommender systems try to predict a user's preference based on some prior knowledge. The different techniques can be distinguished by their underlying background data, the input data and the algorithms used. *Background data* refers to the data that is needed before the recommendation process is initiated. *Input data* is all the data that is submitted by the user in order to generate a recommendation. Both, the input and the background data are then combined with a specific *algorithm* to generate appropriate recommendations. On the basis of this characterization ([Burke, 2002]) the most popular traditional recommender approaches are briefly introduced: rule-based, content-based, knowledge-based, demographic and collaborative filtering recommenders [Brocco, 2008].

2.2.1.1. Rule-Based Recommender

Rule-based recommender systems are probably the simplest way to filter specific items. The background data for this approach is a set of rules implemented by the developers of the recommender system. Input data are items previously chosen by the user. The algorithm then evaluates specific item choices of the user given the rule set provided by the system. A rule might express the following: if a user buys item i , then recommend item j .

Clearly, the quality of the recommendations heavily depends on the quality of the rule set entered by the system developer. Thus, for creating an effective rule-based recommender domain knowledge is required. Since however, maintenance and creation of

2. Open Innovation

the rule set are quite expensive and since the rule set developed is static and cannot be adapted through automated learning, the major outstanding advantage of this system is that recommendations can be arbitrarily controlled and steered. This may prove to be advantageous to push specific items.

To illustrate an example, a rule-based recommender could be used for recommending teams based on specific tasks, i. e. using context instead of drawing on items as explained above. A related rule could read as follows: *“if the task is a programming task, recommend the team composed of the following four programmers . . .”*.

2.2.1.2. Content-Based Recommender

Items can be described with the help of associated meta data. This meta data can be explicitly defined or generated automatically (e. g. keyword extraction in text documents) and represents the background data of this recommendation technique. Meta data may be for example the genre of a movie (comedy, thriller, etc.) or some keywords describing its content. As input data, user feedback (implicit or explicit) is used. Having been fed the meta data of items and the user feedback into a content-based recommender, a classifier generates a model that is henceforth used to compute recommendations.

The shortcoming of this approach resides in the meta data itself: such data has to be defined, in order to comprehensively describe the item. The problem here is that identifying what meta data is relevant for the choice of a specific item type is quite a task. In addition, content-based filtering is affected by the so-called *cold start problem*, meaning that when there is no data available, the quality of the classifiers constructed, is very low. That is the case whenever a new user joins the system and the recommender has no data upon which it can base its computations. Mitigating cold start issues in recommender systems is one of the research fields in recommender systems (e. g. [Lam et al., 2008],[Schein et al., 2002]).

When using content-based recommender systems for team recommendation, appropriate meta data on teams has to be specified. Applying meta data to teams is however very complicated due to the high variety of influencing factors. An appropriate categorization of teams with meta data is therefore challenging to find.

2.2.1.3. Knowledge-Based Recommender

The term knowledge-based recommender refers to a multitude of different recommender systems based on knowledge. This knowledge, operationalized in different manners, pro-

vides the needed background data for the recommender. As input data, user preferences or needs are entered into the system (e. g. in the shape of queries). The algorithm then derives recommendations depending on the specific recommender system.

With reference to Figure 2.3, knowledge-based recommenders are able to exploit all the depicted knowledge sources [Felfernig and Burke, 2008]. In the following, the similarity based algorithm for recommendation developed by Robin Burke ([Burke, 2000]) for a restaurant recommender is presented to demonstrate the functionality of these recommender systems.

Burke’s system allows the user to choose a reference restaurant i from a set of known restaurants. By choosing a specific restaurant, the user informs the recommender that he likes eating in a similar place. After that the recommender generates recommendations based on the similarity between this reference item i with other unknown restaurants. The user then can navigate through different recommendations by indicating his preferences with respect to specific properties. In the case of this restaurant recommender these properties are price, “cuisine” type, style, atmosphere and occasion. One way to navigate the recommendations could therefore be indicating that a cheaper restaurant than the current recommended one is preferred.

The choice and definition of the different similarity metrics and properties used to find “similar” restaurants is demanding. Moreover, also the priority of specific properties has to be considered with knowledge-based recommenders. In the example of the restaurant recommender for instance, Burke stated that the “cuisine” type is the most important aspect that has to be considered. A personalization of these priorities also makes sense in certain scenarios.

The knowledge-based approach is one of the most effective recommender systems but is strongly dependent on domain knowledge. Generating the required knowledge implies three fundamental steps:

1. Collect information on the items (e. g. TU Munich is a university)
2. Collect functional knowledge (e. g. a location with a nice landscape usually does not include many buildings)
3. Collect user information (e. g. demographic data, preferences, requirements)

To perform these three tasks a great deal of effort is required. That is the main drawback of knowledge-based recommenders. Two additional examples for knowledge-based recommenders are given in Section 2.2.2.3 and 2.2.2.4.

2. Open Innovation

Depending on the specific flavor of a knowledge-based recommender, the team recommendation problem with this engine needs to be carefully modeled so that it covers the required aspect entities and similarity computations. For example, modeling task similarity may be challenging under certain circumstances.

2.2.1.4. Demographic Recommender

The name “demographic recommender” is based on the fact that this recommender uses a user’s demographic attributes to generate recommendations i.e. demographic information on the users U of the system constitutes the background data of this paradigm. The demographic attributes of the target user u are provided as input data to produce recommendations.

Demographic attributes may be nationality, age, religion, salary, etc. Principally, this recommender approach works the same way as collaborative filtering (see Section 2.2.1.5 for details): similarity between users is computed so that items liked by a similar user are recommended.

The problem with demographic recommenders is primarily related to privacy issues. Entering personal data into information systems can sometimes be dangerous, because it may be used for other purposes. This especially concerns personal data such as demographic data.

In the team recommendation scenario this would mean that demographic similarity between the project managers is chosen in order to recommend teams, since they are the users of the recommender. It is intuitively clear that this will unlikely result in a good recommendation.

2.2.1.5. Collaborative Filtering Recommender

Probably the most popular and most frequently used recommendation technique is collaborative filtering. The term “collaborative filtering” is derived from the fact that unlike the other above-mentioned recommenders (except for the demographic recommender), recommendations are generated with the help of other users through ratings of specific items. These represent the background data of this system. Input data is a vector of ratings of the user u for different items in I . In the next step similarity between users (user-based collaborative filtering) or items (item-based collaborative filtering) is computed by means of these ratings. For a comparison of user-based and item-based approaches see [Sarwar et al., 2001].

Generally speaking, memory-based and model-based approaches can be distinguished as the two types of collaborative filtering approaches. *Memory-based* collaborative filtering uses the whole dataset to generate predictions while *model-based* approaches create a model from the dataset which is then subsequently used to generate recommendations. This dataset in the simplest case consists of users, items and ratings (also known as user-item matrix).

The core component of memory-based collaborative filtering systems is the neighborhood calculation. Depending on this calculation, the quality of recommendations varies. A common way to compute the neighbors (in terms of rating behavior) of a user u is to measure correlation values between the user ratings. Other measures may be distance or similarity of users. This neighborhood computation is used as weight w for the prediction of a user's u rating r for a specific item i

$$r_{u,i} = \bar{v}_u + \kappa \sum_{j=1}^n w(u,j)(v_{j,i} - \bar{v}_j) \quad (2.1)$$

where \bar{v}_k is the mean vote for user k . κ is a normalizing factor that ensures that the weights sum up to 1 [Chen, 2005]. j is an indexed variable representing all other users in the dataset. Very popular weight functions are the *Pearson correlation coefficient* or the *cosine similarity* [Breese et al., 1998]. The first is also often used in statistics for identifying relationships between random variables or observed data values, which suggests that recommendation problems can be solved with statistical methods. Other distance functions that are used as weight in the above equation can, for example, rely on the social network of a user [Groh and Ehmig, 2007].

Collaborative filtering has a big advantage over other types of recommenders because it is domain independent, meaning it can be applied to several different types of items with no effort. Neither does it require any form of maintenance, such as the inclusion of meta data or the development of domain dependent rules or knowledge bases. However, as content-based recommender systems, collaborative filtering recommenders suffer from the cold start problem.

Apart from that, collaborative filtering is not very appropriate for recommending teams either since it may merely be able to predict which team might be recommended by a similar user of the recommendation engine (e. g. project managers). The lack of previous ratings would moreover lead to problems for the prediction of a team's output (which is analogous to ratings in collaborative filtering).

2.2.2. State of the art

Due to the mentioned specific drawbacks of traditional recommender systems, the effectiveness of single approaches is limited. For this reason researchers have developed new, more sophisticated approaches for enhancing the quality of recommendations. One such possibility to overcome the disadvantages of the individual recommendation systems while exploiting their advantages is to combine several specific approaches into new ones. These techniques are denoted as *hybrid recommenders*. Whenever the situation in which recommendations are provided matters, *context-aware recommender systems* can be used to augment traditional approaches since they consider the context, for which recommendations have to be generated. Another variation of traditional recommender systems are *case-based recommenders* which belong to the class of content-based recommenders [Smyth, 2007]. They use case-based reasoning techniques to generate recommendations. In addition, another type of recommenders are *constraint-based recommenders* which make use of constraint reasoning to recommend items.

Below, these approaches are described as examples for the many other variations of recommender systems available. For an overview on other approaches the reader is referred to [Peter Brusilovsky, 2007] and [Ricci et al., 2011].

2.2.2.1. Hybrid Recommender

As described in Section 2.2.1, each of the traditional recommender systems has its advantages and disadvantages. To exploit the advantages and attenuate the disadvantages hybrid solutions can be developed that combine two or more recommender systems to a new approach. Here the combination strategy plays a major role. Among others, popular combination strategies for hybrid recommender systems are *weighting*, *switching*, *mixing*, *cascading* and *feature augmentation* recommenders [Burke, 2002, Burke, 2007].

Weighted hybrid recommenders usually consist of several recommendation engines that work in parallel. Once the recommendations are generated, depending on the recommendation engine, the result items are weighted. For instance a hybrid recommender consisting of a content-based and a collaborative filtering recommender may weigh the results of the content-based filter with a value x and those of the collaborative filter with a value y . Afterwards, the sum is computed and the items with the best rank are recommended. The challenge of this approach consists in tailoring the weights depending on the items and context. Furthermore, Burke stated that currently weights are static which in some cases is not convenient in terms of quality of recommendations: depending on the situation (or context) weights may change [Burke, 2002].

Hybrid recommenders using the *switching* strategy consist of a set of recommenders among which one is chosen in a specific situation. The specific situation is defined by guidelines that control the switching of the recommenders. Such a guideline could for example induce the selection of a specific recommender in case it produces more recommendations than the others.

The *mixed* strategy works in a similar way as the weighted hybrid: results from different recommender systems are merged into a single result set. However, mixed strategy recommenders generally use slightly different heuristics that do not explicitly consider weights. One of the possible strategies could be counting how often items have been recommended by the single recommender systems and then excluding items that have not been recommended more than once.

As the name suggests, *cascading* hybrid recommenders, cascade single recommenders, i. e. recommendations generated by the previous recommender are then subsequently used as input dataset for the next recommender. This technique corresponds to the pipe and filter architecture used in software engineering. This implies, that an appropriate recommendation hierarchy has to be developed that, for instance, takes into account performance issues: Less demanding recommenders may for example be used in a first step to filter out some items, while more demanding ones should work on a smaller dataset in a second step.

Feature augmentation is a technique that uses one recommender to generate a new feature for the items which in turn is used by the following recommender in order to generate recommendations. Robin Burke describes a feature augmentation hybrid consisting of a content-based and a collaborative filtering component [Burke, 2007]. The content-based component in this example generates a classifier out of the existing features of an item and uses that to rate items and thus generate more ratings on which the collaborative filter can rely. This augmentation for practical reasons is usually performed off-line. Constructing a feature augmentation hybrid is complex and requires a good knowledge of the recommendation logic and the features needed by the single recommenders. However, a feature augmented hybrid is possible for most combinations of traditional recommendation approaches.

Several different hybrid recommenders could be developed to recommend teams. However, the choice of the single components and combination strategy is not straightforward. Intuitively the previously mentioned cascading strategy would be advantageous to limit the search space for recommendations. Hence, these approaches can be useful for the implementation of a team recommender system.

2.2.2.2. Context-Aware Recommender

Traditional recommender systems lack context-awareness. Recommendations generated in a specific context or situation are often not applicable to other contexts. Thus, a recommender system can in specific cases generate better recommendations when the context is known. Recommending a specific gas station for example, is not as valuable when there is enough fuel in a vehicle as it is when there is not. In this example the context dimension would be the fuel level. Another example for the importance of context may be the recommendation of leisure activities where recommendations that take the actual weather into account can be regarded as more valuable. How the context is defined, depends on the specific application. The most frequent context dimension found in recommender applications are time and place. As stated, there are also other factors such as the weather, the fuel level or another person's presence, all of which can be used as context dimension.

Several approaches for incorporating context into recommender systems have been developed so far. Adomavicius et al. ([Adomavicius et al., 2005]) developed a context-aware recommender system based on reduction. This approach consists in reducing the user-item-context multidimensional structure to a simple two dimensional matrix (e.g. a user-item matrix) by selecting ratings that correspond to a specified context. In the following, an example for this context reduction is presented.

The researchers acquired context-data for ratings in a movie dataset. This dataset provided the time a movie was rated/watched, whether there was a companion (e.g. girlfriend, other friends or parents) and where the movie was watched (e.g. movie theater, TV, DVD, etc.). These three aspects are considered as the context. By selecting for which situation the recommendation is needed, the corresponding aspects are selected and thereby the user-item-context matrix is reduced to a simple two-dimensional user-item matrix. Thereby, traditional recommendation approaches based on ratings could be applied for items rated in similar situations. Their evaluation showed significant improvements by means of the recommendation quality.

Annie Chen proposed incorporating context into collaborative filtering recommenders [Chen, 2005] by including the similarity of context in Equation 2.1 (Section 2.2.1.5) as weight parameters instead of using the similarity of users with respect to their rating behavior only. How the similarity of context is determined depends on the context type. An overview on similarity metrics is provided e.g. by [Leake et al., 2002, Cunningham, 2009].

Context-aware recommender systems based on learning approaches particularly suffer from the cold-start problem. The integration of context in the recommendation process leads to a more sparse user-item matrix so that less explicit feedback (e.g. ratings) can be

found for an item than in the context-unaware case. This is why corresponding mechanisms have to be developed to counter this effect.

Understanding team composition and hence team recommendation as a context-aware problem is absolutely necessary. Due to the definition of context used in recommender systems however, these concepts cannot be directly applied. Instead, concepts derived from context-aware recommender systems can be fruitful to implement a team recommender that is aware of changing environmental aspects such as changing tasks, changing organizational constraints etc.

2.2.2.3. Case-Based Recommender

A slightly different concept for recommendation is implemented by case-based recommenders. These recommender systems are derived from case-based reasoning (CBR) which is a problem solving approach that makes use of the solutions of similar problems that have been previously solved. Both, problem description and solution description are stored in a repository which is denoted as *case base*. Whenever a solution for a new problem is needed, the problem description is analyzed and compared to existing problems in the case base. According to the similarity between this query (problem description) and the case base, solutions for similar already solved problems are retrieved and returned as possible results.

The retrieved solutions, that are based on the similarity assessment, can be further *adapted* to the current problem if necessary. Therefore, case-based reasoning differs from other problem solving methods (e. g. machine learning) in their reliance on concrete experiences instead of problem solving knowledge in form of codified rules and strong domain models. Figure 2.4 depicts the basic idea of solution retrieval in CBR.

Aamodt & Plaza introduced in [Aamodt and Plaza, 1994] the classic model of problem solving cycle in CBR that is widely used as the general aspect of CBR. The cycle consists of the following individual tasks that are also depicted in Figure 2.5:

1. *retrieving* one or more most similar experienced cases,
2. *reusing* the solution of that case in one way or another to solve the problem,
3. *revising* the knowledge or solution proposed by this previous case,
4. *retaining* the parts of this experience likely to be useful for future problem solving by incorporating it into the existing case base.

2. Open Innovation

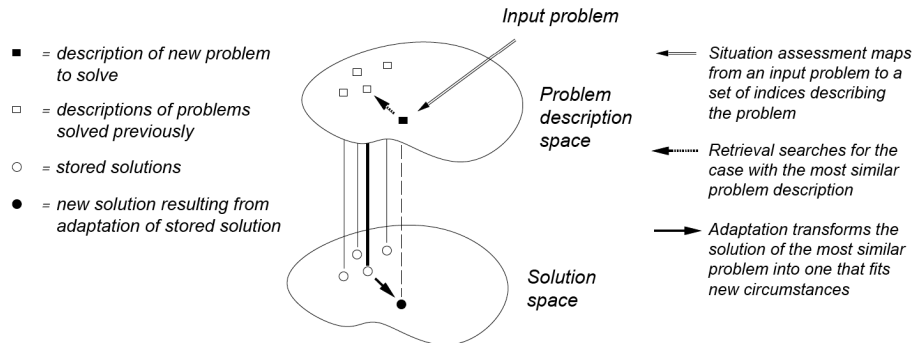


Figure 2.4.: CBR is the process of solving new problems based on the solutions of similar past problems [Leake, 1996].

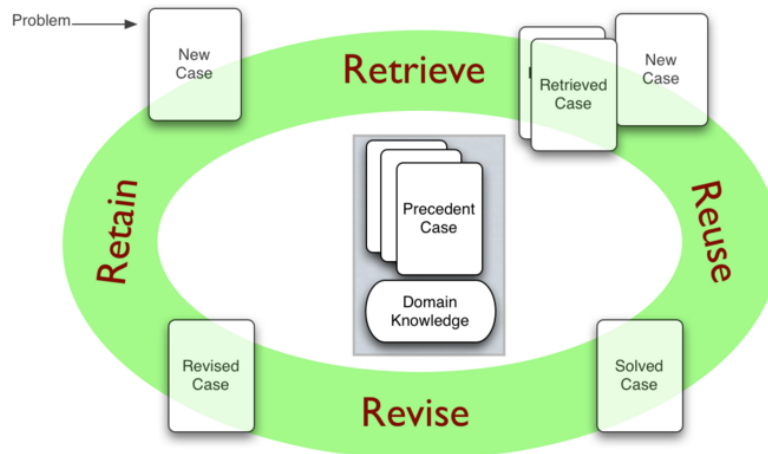


Figure 2.5.: Problem solving cycle in CBR³

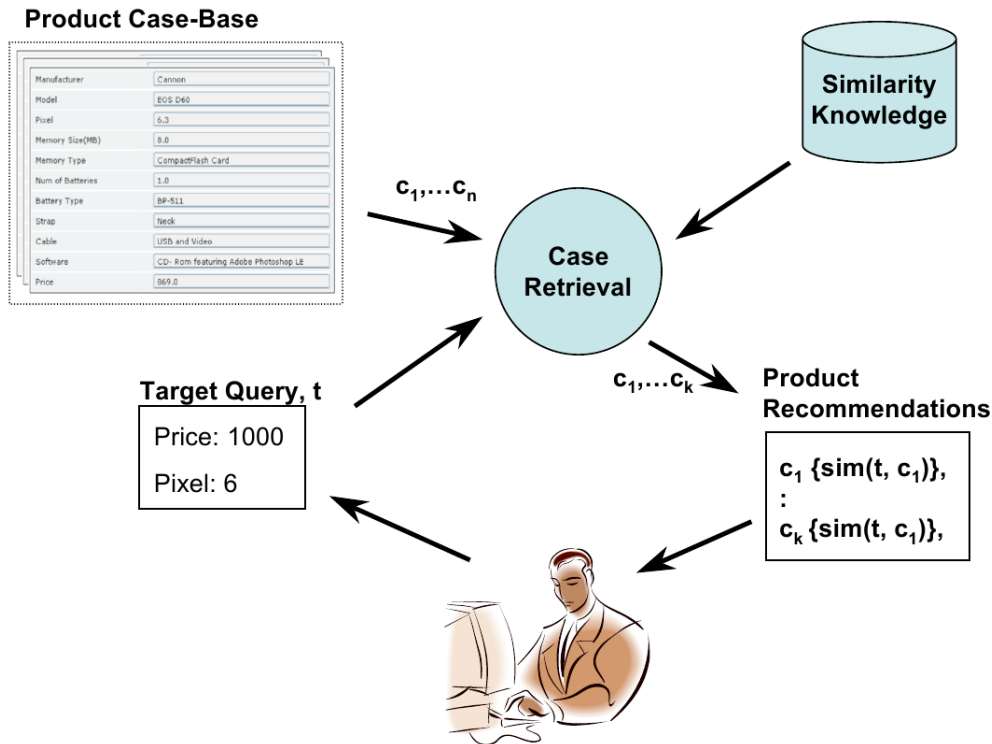


Figure 2.6.: Case-based recommender approach example [Smyth, 2007]

Case-based recommendation can also be regarded as a particular type of knowledge-based recommender (in contrast to Smyth’s categorization that states that case-based recommender are content-based filters [Smyth, 2007]) where the knowledge resides especially in the similarity measurements used [Felfernig and Burke, 2008]. Figure 2.6 depicts a typical example of case-based recommendation using digital cameras as items and their description as problem specification. The description in this specific case consists of *features* such as manufacturer, model, pixel, memory type, etc.

As already mentioned, similar items have to be identified in order to find appropriate solutions and hence recommendations for a given query. The similarity assessment involved in this process is the core component of a case-based recommender. Generally speaking, the more precise the similarity assessment, the better the quality of the recommendation given this approach. Similarity assessment of various feature types is one of the widest areas of research on CBR. Cunningham [Cunningham, 2009], Avramenko and Kraslawski [Avramenko and Kraslawski, 2006] and Leake et al. [Leake et al., 2002] provide an overview on various different similarity measures for different feature types. The most common feature type is the numeric feature. A numeric feature can be categorized

³<http://cbrwiki.fdi.ucm.es/>

2. Open Innovation

further in *symmetric* and *asymmetric* features, which again requires distinctive similarity metrics. Symmetric features only consider the absolute distance between a target query and an actual value. For example, if it is required that a team member is 20 years old, this type of feature does not distinguish between a team member that is 18 and another that is 22 years old. An example for an asymmetric feature is the required experience of a team member. If, for instance, a member with five years of experience is required, a candidate member with six years of experience should be more preferable than another with only four years of experience. Most of these individual similarity measures can also be applied to other feature types provided that the features are mapped to numerical values. For instance, the feature “grade” can have categorical values which can be mapped as “A=1”, “B=2”, etc. The feature types that do not require any mapping to numerical values (nominal-scaled features) can use simple string matching (comparison of categorical features) or Levenshtein distance (distance between nodes in a tree of hierarchical feature values).

Case-based reasoning in the context of team recommendation implies that a very particular strategy for recommending teams is used. That is the assumption of similar teams having to solve similar tasks with a similar output. Given the fact that this assumption applies, case-based recommender may be a possible way to implement a team recommender.

2.2.2.4. Constraint-Based Recommender

Constraint-based recommender systems are a subcategory of knowledge-based recommender systems (Section 2.2.1.3) and therefore have the same drawbacks and advantages. The property that characterizes a constraint-based recommender is the way recommendations are generated out of the knowledge base (constraint reasoning). Constraint reasoning is a field of artificial intelligence that is used to solve constraint satisfaction problems, i.e. solve a problem, given specific constraints. Typical constraint satisfaction problems are the graph coloring problem or the 8-queens problem. In constraint reasoning problems are structured by values of respective variables that have to meet specific predefined constraints ([Russell and Norvig, 2003]). Each variable usually has a domain so that only values of the corresponding associated domain can be assigned to variables. An example for a domain for variables of the team recommendation problem is the gender of team members (i.e. possible values for the variable are *male* or *female*).

This paradigm can also be used to generate recommendations when domain constraints and user constraints have to be regarded (cf. Figure 2.3). In some collaborative or content-based approach, it is also possible to consider user input. Knowledge- and specifically constraint-based approaches have their emphasis on a user’s need in a specific situation

[Felfernig and Burke, 2008].

Users usually give their preference with respect to desired items in constraint-based recommenders. As this kind of recommender is not affected by the cold start problem, it is suitable for domains in which items are rarely purchased or chosen. In the domain of constraint-based recommendation, a conversational component is very important, especially when the entered constraint cannot be completely satisfied. In that case it is particularly important, that a constraint-based recommender elicits in a dialog with the user, which constraints can be relaxed or which constraints have been relaxed, in order to produce some recommendations.

In addition, instead of *hard constraints*, *soft constraints* can be applied. While in hard constraints the tuples of values are either allowed or not, soft constraints allow for a partial satisfaction. These are useful when a problem is not solvable with hard constraints (over-constrained). They also make it possible to discriminate the solution tuples more meticulously, i. e. to what extent the constraints are met. If a query requires experts with 5 years of organizational tenure, hard constraints would only return tuples describing people with exactly 5 years. Modeling this problem as soft constraint would return a ranked list of tuples that would also include people with 4, 3, 2, etc. years of experience. Several models for soft constraints are known so far. The most popular are hierarchical, partial, valued and semiring-based models. [Bartak, 2002, Rossi et al., 2006]

The knowledge base of constraint-based recommenders consists of a set of properties that can be set in a user query, and attributes of the items. Furthermore, they include a set of constraints representing compatibility or incompatibility of these queries. In the domain of team recommendation, given the query variables *team size* and *project volume*, one could insert in the knowledge base that if a project volume of 200.000 € is entered in the query, a team size of less than 4 persons is not valid. Respectively constraints on the item side may detect incompatibilities between specific item properties and give feedback to the user on how to possibly solve them. Finally, filter constraints connect the user query to the items (e. g. a team with high tenure is a team in which the sum of the single team members' tenure is equal to 30 years). As a result, items fulfilling the defined constraints are returned.

In order to facilitate the choice of specific products, the item attributes can be added to the results (e. g. reliability, performance etc.). These can be combined in a utility function with parametrized weights, so that the user can be supported when selecting an item among the returned set of all items that satisfy the constraints.

To address the consideration of organizational constraints such as availability, filter-

2. Open Innovation

ing according to partner organizations etc., a constraint-based algorithm using hard constraints may be indicated. But for more finely grained considerations an appropriate, more sophisticated model using soft constraints is required. More insight into the world of constraint-based recommenders can be gained from [Zanker et al., 2010].

2.2.2.5. Compositional Recommender

In contrast to the above-mentioned recommender systems, compositional recommender systems do not recommend single items, but rather a collection of items. In that sense, there is an analogy to the problem of recommending a team, i. e. a *set* of people instead of single individuals is recommended. Particular to this approach is the fact that it considers the set of items as a single item, rather than recommending the first k items. This is due to the particular utility function of the items to be recommended: whenever the utility of a set of items cannot be computed as the sum of the utility of each single item, dependencies exist that have to be considered. E. g. in the domain of team composition two persons on their own do not produce the same output as when they work together. Hence, they should not be recommended separately but rather as a group of items with a respective predicted utility.

Few compositional recommenders are known in research so far. One example is Aguzzoli et al.'s work [Aguzzoli et al., 2002], which describes a system for recommending music compilations consisting of several different tracks. The authors argue that tracks are composed into a compilation according to their context which in their case is specified by a music genre or a topic (e. g. "California dreaming music"). Thus, they consider each compilation as a single item. This item in their system also represents a case in a case base: their approach therefore includes a case-based reasoning engine for recommending compilations. In contrast to traditional CBR systems, their approach does not rely on content features for assessing similarity (see Section 2.2.2.3) between cases (i. e. compilations). Their reasoning is that content features cannot easily and adequately represent the cases (compilations). There is, for example, no common agreement on the different music genres.

So, they developed a collaborative case-based recommender that maps concepts of case-based recommendation to the classical collaborative filtering approach. In particular, they developed a measure that replaces the notion of ratings between user and items with the notion of how much a collection "likes" a single track. This measure is computed through the co-occurrence of tracks in specific compilation subsets. With this measure, the correlation between two compilations can be computed in the same fashion as in traditional collaborative filtering approaches: compilations that are *very similar to the*

current compilation (with respect to the above-mentioned “rating” measure) are returned as a recommendation. As described in Section 2.2.1.5 in the traditional case *similarity between users* or items, by means of ratings, is used.

The approach described here is a good example for using and combining techniques from different recommendation systems so that they can be used in a new, more complex domain than the product recommendation domain. It indicates that for the team recommendation approach several elements of other recommendation systems can be used.

2.2.2.6. Collaborative Filtering using Latent Factor Models

The latent factor collaborative filter has recently become one of the most popular approaches in the recommender research field. It emerged as the best recommender approach from the Netflix Prize. Netflix is a company providing online-movie rental and video streaming in the United States. Netflix started a competition called “Netflix Prize” in 2006 (cf. www.netflixprize.com). Its goal was to outperform the collaborative filtering algorithm developed by Netflix by 10%. The quality metric used was the root mean squared error (RMSE). The winner was awarded a sum of 1 million US Dollars in 2009. All best teams in this competition applied latent factor collaborative filtering approaches.

In contrast to the neighborhood approaches described in Section 2.2.1.5, latent factor analysis tries to extrapolate non-visible or inherently obvious influence factors in ratings. Such factors range from gender to orientation to children or completely uninterpretable dimensions.

One of the popular instantiations for latent factor analysis is matrix factorization where vectors of factors inferred from the rating behavior are associated to user and items in order to predict a matrix

$$X \approx UI \tag{2.2}$$

where U contains users and their susceptibility with respect to the latent factors and I contains items and the extent of the latent factors they exhibit. X is the desired user-item matrix with the predicted ratings.

If the factors included in a user vector are very similar to the corresponding factors of the item vectors, the items are recommended. An advantage of matrix factorization models is that not only ratings can be used to generate recommendations, but also other arbitrary data such as implicit feedback.

In order to predict a user’s rating using matrix factorization, users as well as items

2. Open Innovation

have to be mapped to a space of fixed dimensionality, denoted as f . $|f|$ represents the amount of latent factors to be used. For items and users appropriate vectors are created of the same dimension $|f|$, e. g. q_i for an item i and p_u for a user u . In q_i the extent of each of the elicited latent factors for the item i is included, i. e. the vector expresses to what extent these factors apply to the item i . Similarly for a user u the vector p describes the amount of interest of this user for the corresponding (latent) factors. Values in p and q can be positive as well as negative. To predict the rating of a user u for item i , the dot product $q_i^T p_u$ has to be computed. Thus,

$$r_{u,i} = q_i^T p_u \quad (2.3)$$

where r is the rating of a user for a specific item. The challenge here is to approximate the factors given the existing ratings. To this end, the regularized square error has to be minimized:

$$\min_{q,p} = \sum_{(u,i) \in \kappa} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \quad (2.4)$$

κ stands for the set of all user-item pairs, for which a rating exists. λ is a constant that is responsible for the regularization and can be determined with the use of cross-validation (see Section 2.2.3).

One way of minimizing the regularized square error is by computing the error of some randomly created matrices and trying to minimize the error iteratively. This method is known as *gradient descent*. Another related approach is the *alternating least squares* approach. For a detailed description of these and other approaches for approximating latent factors refer to [Takács et al., 2008] and [Koren et al., 2009].

Using latent factor analysis may be a good solution for the team recommendation problem. However, as a collaborative filtering approach, it heavily depends on generated data that may not be available. Furthermore, this kind of approach is only valuable for domains that do not constantly change. A further drawback is the difficulty of explaining these latent factors to the user in charge of assembling a team and the inconvenience in context with the transparency of recommendations. One possibility to use latent factors for team recommendation would be through application of reduction-based context-aware approaches like [Adomavicius et al., 2005]. The context could be represented by aspects such as the task or the environment the team has to perform in. The sparsity of data would however cause dissatisfactory results.

2.2.3. Evaluation of Recommender Systems

One of the most challenging parts of recommenders in research is their evaluation. To evaluate a recommender's quality it usually has to be ensured that the recommendations provided correspond to a user's taste or expectation. However, this is not the only possible criterion to evaluate a recommender system against. Also the interaction of the user could be evaluated or the transparency of recommendations, the novelty etc.

A recommender system can generally be evaluated online (that is during its application), offline (with an existing dataset) or in a user study. Offline evaluations are advantageous to measure the accuracy of recommender systems while they are not appropriate when external factors may influence a user's preference. For this purpose the more time consuming online experiments are preferable. User studies on the other hand allow for the investigation of finer-grained aspects and evaluation criteria such as in the case of transparency or interaction with the user interface.

The most popular approach to evaluate a recommender's prediction power is by using offline evaluations, because on the same dataset several recommendation algorithms can be tested. The online method is more time consuming because for each recommendation approach a significant amount of test users has to participate in the evaluation. A disadvantage of most of the offline evaluations is that only items are considered that have been rated. In case participants have only rated a certain category of movies (e. g. movies they like) this method may be problematic.

In offline evaluations *K-fold cross validations* are often performed to assess a recommender's quality. In this type of evaluation the dataset is randomly partitioned in K folds of approximately the same size. In each run, K-1 partitions are used to train the recommender system (training set) and the remaining partition is used to compare the result of the recommender with the actual ratings or preferences. This test is performed for each of the K folds and the results of each evaluation are finally averaged. The metrics used are arbitrary. This type of evaluation is robust against overfitting and has the advantage of all data records being used as a training and testing set.

Depending on the type of prediction that has to be evaluated, several metrics can be used. Generally, two different accuracy metrics can be distinguished: the accuracy of ratings/output measures and the accuracy of usage. The first accuracy is usually used to evaluate whether a recommender system is able to predict ratings a user assigned to a specific item (e. g. a movie). Examples for this kind of measures are the root mean squared error (RMSE) and the mean average error (MAE). The first metric penalizes higher errors more than the second metric.

2. Open Innovation

If S is a test set with ratings r for user-item pairs (u,i) it is necessary to compare these ratings with the predicted ratings $\hat{r}_{u,i}$ of the recommender system in order to evaluate its accuracy. The two above described measures are accordingly defined as:

$$RMSE = \sqrt{\frac{1}{|S|} \sum_{(u,i) \in S} (\hat{r}_{(u,i)} - r_{(u,i)})^2} \quad (2.5)$$

$$MAE = \frac{1}{|S|} \sum_{(u,i) \in S} |\hat{r}_{(u,i)} - r_{(u,i)}| \quad (2.6)$$

While these metrics are used to evaluate the accuracy of the predicted rating, the second class of metrics (accuracy for usage) has its origin in information retrieval where it is important to deliver “relevant” documents to the user. For this reason it is important to define when an item becomes relevant. For this purpose a threshold has to be defined for rating-based systems which indicates when an item is relevant. E. g. , for ratings between 0 (worst) and 5 (best) it could be defined that relevant items must have a minimum rating of 4. In other words, a binary result set has to be induced. Afterwards, the recommender has to classify items as relevant and not relevant. The returned items that are relevant are labeled as “true-positives”. Relevant items that were not recommended are denoted as “false-negatives”. Irrelevant items that were recommended are called “false-positives” whereas irrelevant items that were not recommended are “true-negatives”. Given corresponding abbreviations (tp,fn,fp,tn) the measures precision and recall are defined as:

$$\text{Precision} = \frac{\#tp}{\#tp + \#fp}$$
$$\text{Recall} = \frac{\#tp}{\#tp + \#fn}$$

Precision indicates the percentage of relevant items within the recommended set. The recall measure represents the percentage of relevant items within the whole set of relevant items. Generally, it’s difficult to achieve a high value for both the measures. More likely, there is a tradeoff between the two measures: recommending more items normally increases the recall, while the precision gets lower. For this reason the so called F_1 – score (also known as F measure) is used to normalize the recommendation quality. This metric represents the harmonic mean of precision and recall:

$$\text{F-Measure} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

These are not the only metrics and criteria that can be used to evaluate recommender systems. For a better and comprehensive overview on evaluation methodologies, criteria and metrics compare the works of Shani and Gunawardana [Shani and Gunawardana, 2011] and Adomavicius and Tuzhilin [Adomavicius and Tuzhilin, 2005].

2. *Open Innovation*

3. Team Composition

To implement a team recommender system knowledge about the application domain and possible algorithmic solutions has to be acquired. Chapter 2 provided the fundamentals in this regard. To approach the solution domain however another topic has to be investigated: team composition itself. For this reason the goal of this chapter is to analyze from an interdisciplinary perspective, how teams are composed and thereafter propose an appropriate approach towards team recommendation.

Several studies performed in the socio-psychological sciences as well as in the organizational sciences have indicated that team composition critically affects a team's output/outcomes (e. g. [Kozlowski and Ilgen, 2006, Trower and Moore, 1996]) or emphasized team composition as important field for future research (e. g. [Powell et al., 2004]). Hence, the problem of composing effective and appropriate teams is not only relevant for the open innovation case, but also more generally for any kind of team work (e. g. virtual, distributed teams). Performance output such as effectiveness, innovativeness and creativity seem to improve when effective teams are composed.

Different approaches exist to assemble teams for short, medium and long term projects. Some of them are manually applied, some already make use of IT support. The composition is however not the only thing that influences a team's output and outcomes (e. g. performance). In order to build a recommendation system for team compositions the whole domain has to be clearly understood and considered. This comprises, for instance, the elicitation of influence factors for team performance that are not inherently associated with the composition of a team such as the task a team has to perform or organizational constraints. For example, despite a "good" team composition, the team may not perform well because of inconvenient constraints (e. g. not enough time to accomplish the task). Literature on team composition and team performance is very useful for identifying this type of influence factors.

These literature findings are however not always enough to comprehensively describe the team composition domain since they are rarely applied in practice. Sometimes they do not cover contextual constraints that in practice play a major role. Since the team recommendation approach described in this thesis principally addresses open innovation

3. Team Composition

networks and thereby companies, the topic of team composition has also to be investigated from a practical point of view, i. e. as project managers (or other persons in charge to assembly teams) experience it in their daily activities.

In summary, this chapter introduces works on team composition from three different areas: behavioral science, practice and computer science. *Approaches from behavioral science* in the nomenclature of this thesis cover all works on team composition done in the social and organizational sciences and have not been applied for algorithmic use. *Approaches from practice* can be regarded as the best practices and team composition strategies used in companies. Finally, team composition systems existing in IT and computer science research that are able to algorithmically compose teams are denoted as *algorithmic approaches*. This categorization of related work is depicted in Figure 3.1.

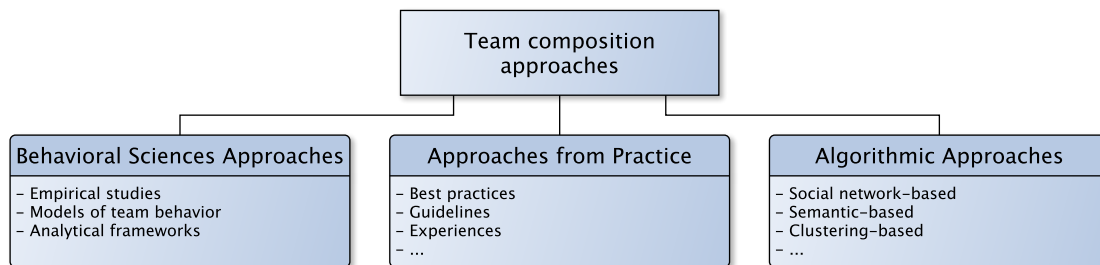


Figure 3.1.: Categorization of related work on team composition

In the next section approaches from behavioral science on the composition of effective teams and on the modeling of team performance that mostly base on case studies, literature reviews or other related research are introduced. Section 3.2 presents team composition strategies used in practice and elicited as one contribution of this thesis, from a survey on team composition conducted in companies of the ICT sector located in Germany. Afterwards, existing approaches for the algorithmic composition and recommendation of teams are presented. The chapter ends with the discussion of these three categories (approaches from behavioral science, practice and algorithmic approaches) with respect to their role in the development of a team recommender system.

The goal of this chapter and all its contained sections is to

- a) show different areas in which team composition has been investigated and used,
- b) gain an insight into what factors play a role in existing team composition strategies,
- c) show the variety of aspects influencing a team's performance, and finally
- d) investigate how the various approaches relate to each other.

Hence, the specific results from a semantic perspective (e. g. whether diversity of ages really affects team performance or not) are not in the focus of this chapter. The focus is on the methods used to derive these results (e. g. statistical analysis, interviews), the class the identified aspects belong to (e. g. age is a property of a person, while diversity is a property of a team) and the domain the work has been created in (e. g. management sciences).

3.1. Approaches from behavioral sciences

Most of the investigations on team composition have been conducted in the sociopsychological and the organizational sciences (e. g. organizational behavior). A large number of studies and articles analyze the influence of certain aspects such as the organization, the type of task or the communication media on team performance and outcomes [Hackman, 1987]. In addition to these fields of research studies performed in the domain of computer supported cooperative work (CSCW) are also valuable in this regard: they give insight into influence factors of teams using electronic group support systems to collaborate.

Some of the approaches coming from these fields have a retrospective view on team composition i. e. they descriptively analyze, why and under which circumstances teams performed well with respect to previously defined hypotheses (e. g. in lab experiments). Similarly, works such as case studies analyze (often qualitatively) past behavior of teams. Given the findings of these studies, the effectiveness of future team compositions for specific situations can be improved. All these mentioned studies thence are *empirical studies*.

Another subset of studies tries to give guidelines for good team compositions by reasoning composition strategies on the foundations of previous work (these works are labeled in the following sections as *models of team behavior*).

Finally, works based on *analytical frameworks* such as game-theoretical models can be consulted in order to study effective team composition. Given a modeling framework, the team composition problem is first mapped as problem of this problem class and then solved with the tools and mechanisms provided by the framework.

In the following, some representative examples of these three different behavioral sciences approaches are introduced. The examples have been selected from several articles, but cannot cover all research done in this domain. Nevertheless, they represent typical research in this domain.

3. Team Composition

In order to provide a better overview of existing approaches the following subsections are structured based on the taxonomy of approaches described in the chapter introduction, namely empirical studies, models of team behavior and analytical frameworks.

3.1.1. Empirical studies

Empirical studies on the factors influencing team performance are a valuable source of information for effective team compositions. Example subcategories belonging to this type of work are *(field) experiments*, *case studies* or *field studies*.

Usually, in *(field) experiments* hypotheses are generated e. g. on the basis of previous work. Then, teams are observed with respect to these hypotheses. The observations made during the experiment are used to validate the hypotheses in a statistical manner. If a significant correlation within variables contained in the observations for the relationship assumed in the hypotheses exists, the hypotheses can be accepted. In some cases, no hypotheses are created. Instead, given the available variables, possible correlations are investigated. To summarize, work in this category employs quantitative research methods.

Case studies and *field studies* are usually undertaken to exploratively elaborate new hypotheses, to provide descriptions or to test theories [Eisenhardt, 1989].

This subsection describes studies for effective team work in product development and creative problem-solving. The last part of this subsection provides an excursus into studies that are not strictly related to the HR (human resources) domain. Nevertheless, they are mentioned in this context because (i) they are interesting for extending team recommendation to other domains, (ii) they may provide valuable data to partially evaluate a team recommendation approach.

The following paragraphs merely present a small portion of the available literature in this domain which comprises thousands of similar works from organizational sciences, organizational behavior, sociology or psychology or computer supported cooperative work. A valuable overview belonging to the latter category can be found, for example, in [Fjermestad and Hiltz, 2000] where several studies on the impact of group support systems on team performance and other team output are summarized. Furthermore, related insights from the other categories can be gained from the following works among others: [Trower and Moore, 1996], [Convertino et al., 2008], [Jackson and Joshi, 2004] and [Bradner et al., 2005].

Demography and design: predictors of new product team performance

One of the most active researchers in the analysis of team performance is Deborah G. Ancona. In her study with David Caldwell ([Ancona and Caldwell, 1989]) she investigated whether demography is related to team performance.

Demography, as defined by the authors, was captured through the coefficient of variation of tenure and the Teachman entropy of functional areas the team members worked in, such as marketing, manufacturing or engineering¹.

Team performance was rated by team members on eight dimensions including efficiency, quality, technical innovation, adherence to schedules, adherence to budgets, coordination, work excellence and ability to resolve conflicts. However, a principal component analysis yielded a single factor. Therefore, the eight items were averaged to form an overall measure for each of the participants (and then again averaged to obtain an overall team performance measure).

The study conducted by Ancona and Caldwell comprised 47 new product teams in five high-tech companies in the IT, analytic instrumentation and photography sector within a time span of 1.5 to 3 years. 10 was the average number of team members per analyzed team.

The most interesting results of this study indicated that there are very strong direct relationships between demography and team performance while also other, less strong, indirect relationships were found: Functional diversity as well as diversity of tenure correlated with lower level of performance. Ancona and Caldwell motivated this negative impact with the extremely high level of diversity found among the teams which might not be advantageous for implementing ideas.

Informal contacts and performance in innovation teams

In certain cases, the relationships between team members can influence a team's performance not only when conflicts in a team arise (e. g. see Section 3.1.2), but also when there exist very tight friendship relations.

Kratzer et al. investigated the relationship of informal contacts and performance in innovation teams [Kratzer et al., 2005]. To be more precise, they observed whether the frequency of informal contacts and more specifically whether the cohesiveness of friendly and friendship relations affect team performance. For this purpose they analyzed social

¹for more details on these measures consult [Harrison and Klein, 2007]

3. Team Composition

network data of 44 innovation teams. Informal contacts were categorized into *friendly communication* (informal contacts via email, phone etc.) and *friendship relations* (informal meetings for free-time activities). Team performance was assessed by self-report, i. e. each team member within a team had to assess their perceived team performance. The cohesiveness was determined by the mean number of contacts per week per team member. This means that the cohesiveness was 1 in case all team members had at least one informal contact (for friendly network) with everyone else within a week. Regarding the friendship network, cohesiveness was highest (1) when all members of the team met each other during their free time, outside the work context. As control variables they included team tenure, team size and phase of the innovation process.

Their investigation hypotheses were:

- H1: The cohesiveness of friendly ties has an inversely U-shaped relationship to team performance.
- H2: The cohesiveness of friendship relations has an inversely U-shaped relationship to team performance.
- H3: The cohesiveness of friendship relations has a positive relation to team performance.

It has to be noticed that the difference between H2 and H3 resides in the type of relationship which is U-shaped for H2 and linear for H3.

Their statistical analysis brings evidence for the relation between frequency of friendship ties and positive team performance (H3) thus, not exhibiting the inversely U-shaped relation assumed in H2. H1 could be accepted since there was significant evidence that friendly ties have positive as well as negative consequences on team performance, depending on its extent.

National and cultural diversity in global innovation teams. Creativity and innovation as a function of cultural team composition

In the field of open innovation a work by Bouncken and Winkler provides interesting insights in terms of team composition for distributed, global, multi-cultural innovation teams [Bouncken and Winkler, 2008]. Herein the open innovation aspect resides in the collaboration of several organizational units distributed around the world.

The above-mentioned study analyzes several aspects of diversity regarding multi-cultural innovation teams (MCITs) which are said to increase products' attractiveness to

foreign and global markets. Great diversity in teams however, may also lead to process losses and poor performance as different cultural backgrounds easily give rise to misunderstandings and complicate working relationships.

Inspired by work of Hofstede [Hofstede, 1984] as well as the work of Edward Hall and Mildred Hall [Hall and Hall, 1990] on cultural diversities, Bouncken and Winkler analyzed specific effects of diversity on creativity and innovation by conducting a longitudinal study on five large multi-cultural innovation teams (some of them consisting of three sub-teams) over two years. They observed how communication styles affect a team's innovativeness in the domains food and cosmetics. The company the teams belonged to was pursuing radical innovations.

Bouncken and Winkler assume that communication styles differ according to the nationality and culture of team members. During their study they observed that the diversity in terms of time perception (as described in [Hall and Hall, 1990]) among the culturally different team members impeded creativity and innovation.

Moreover, the study yielded that in bi-national teams differences in power distance and "cultural context" lead to lower levels of creativity of the solutions and less innovation. "Cultural context" refers to the number of unwritten behavior rules in a specific society. If in bi-national teams half of the team is "low-context", meaning that the number of unwritten behavior rules in their culture is rather low, and the other "high-context" the team will not perform well². Furthermore, a great power distance between team leader and team members also does not contribute positively to the team's performance.

Bouncken and Winkler's last proposition states that a highly diverse team in terms of culture and nationality can positively influence individual motivation and group cohesion and thus be a source of creativity and innovation. This finding was based on multicultural teams with strong English skills. Figure 3.2 replicates the mentioned propositions that represent the results of Bouncken and Winkler's exploratory case study.

As depicted in this figure, context, time and power distance (framed in the black box) can affect team work quality in a negative manner, if they are different within a team. This effect is accentuated in bi-national teams with nationalities that differ with respect to these three cultural dimensions.

²for more information on these criteria see [Hall and Hall, 1990]

3. Team Composition

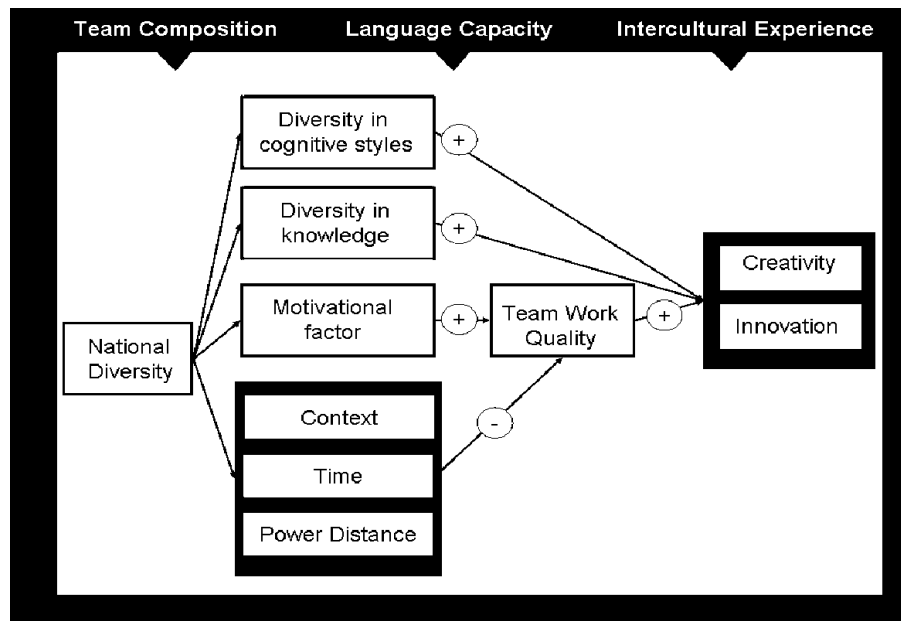


Figure 3.2.: Influences on creativity and innovation in MCITs [Bouncken and Winkler, 2008]

Excursus: Studies for Massive Multiplayer Online Role-Playing Games Teams and Wiki Teams

As mentioned in the preamble of this section, the contributions presented in the following are not strictly related to the field of open innovation and may appear to be somewhat off-topic. However, they do provide a more general view on the team recommendation problem since they show that the problem of effective team composition can be extended to other domains. Furthermore, they can provide precious input data for evaluating the overall feasibility and partially the quality of a team recommender system. As will be dealt with in Chapter 6, the evaluation of team recommender systems is very challenging, because of the high context variety and the difficulty to acquire adequate data sets. For this reason these approaches are very valuable and reflect current research on algorithmic team composition.

The first work by Shim and Srivastava [Shim and Srivastava, 2010] presents statistics on multiplayer games. To be more precise, it shows what properties effective teams in a massive multiplayer online role-playing game named Everquest II (cf. <http://www.everquest2.com>) possess by evaluating data collected during nine months of player and team activity. The dataset contains 63,707 users of different levels of expertise. The available game data included character id, character sub-class, race, task, time stamp of

task completion, group size (in case of teams), average group level, experience points and location (location in which the task was completed). Furthermore, from this data Shim and Srivastava defined several individual- and group-performance metrics. Two performance metrics for the evaluation of teams as a whole were defined. The first metric is an Efficiency Index which is a function of experience points gained and the play time. The second is a Casualty/Survivability Index that measures the fraction between the percentage of casualties during operation and the percentage of survived players that accomplished a task.

Among other things they investigated to what extent the team size influences team performance. Their statistical analysis showed that playing in a team increases efficiency up to a group size of six and decreases afterwards. Starting from a group size of 12 team members it is more effective to play alone than in a team. A similar behavior can be observed in relation to the Success Index which indicates the ratio between the number of killed monsters and all casualties (monsters + own deaths) for each single player: the average Success Index for a task within a team is higher than the average Success Index for solo players for the same task up to a size of 18 team members, whereby teams consisting of six players seem to be the most effective. Furthermore, Shim and Srivastava also investigated the impact of diversity in terms of level of the players and task difficulty on team performance. For a detailed discussion and presentation of these results refer to [Shim and Srivastava, 2010].

It should be noted, that the researchers also created a model that they plan to use for recommending teams in this game based on the findings of their study. For instance, their significance analysis suggested them to build a linear regression model to predict team performance based on group size. A second model is to predict team performance according to the average level of individuals in a team. A third multiple linear regression model addresses the prediction of team performance based on both, group size and group level. All three models were suggested as a start for creating a team recommendation approach since they exhibit high coverage and significance.

Another interesting case study elucidated measures to assess good team performance among Wikipedia author groups [Turek et al., 2010]. Turek et al. defined several variables in a team such as trust (which is operationalized by the number of words authors copied and pasted from other authors' contributions in the article), distrust (which is operationalized through the deletion of text), acquaintance (operationalized by the distance measured in number of words between statements of two authors in the discussion tab of a Wikipedia article) and knowledge (which is operationalized by the activity of an author in a certain category).

3. Team Composition

As performance measures the authors defined the sum of the first three mentioned variables (i. e. trust, distrust and acquaintance) and normalized it in two different ways. Since knowledge consisted of another type of relation (namely author-category instead of author-author) slightly different performance metrics were used. The first metric specifies the average team expertise in each of the categories that were relevant for writing the article. As second metric the authors looked for the team members with the highest knowledge values in the different categories. Among them, the expert with the lowest knowledge value was selected. His knowledge value was then used as result for this second metric.

In order to analyze these metrics, they partitioned Wikipedia articles in three sets. The first one consisted of *good* articles. These partition included featured articles that meet all specific required criteria (e. g. not lacking citations) and gained their status through voting. In addition, other *good* articles were included in this partition that were well written but did not meet all required criteria to be denoted as “featured”. The second partition includes all other completed articles (which was established through the number of edits). The last partition consisted of unfinished articles and was hence not included in the analysis.

With the help of these two partitions the researchers investigated whether good and normal teams significantly differ in terms of the before mentioned measures trust, distrust, acquaintance and knowledge. Their comparison indicates that the trust and acquaintance values are significantly different in good teams and normal teams: Good teams possess much higher values than normal teams. These values can hence be used for predicting team performance in a recommendation system for wiki teams (e. g. teams creating wiki articles in an open innovation network).

Summary

It is apparent that, all empirical studies provide a good operationalization of rather complicated aspects. For instance, Kratzer’s work provided operationalizations for the concept of “friendly relations”. In empirical studies all output and outcomes measures are operationalized and can be very diverse in terms of semantic. To characterize teams, aggregated team member aspects are often used such as the average or standard deviation of certain values. While case studies do not always provide concrete values since they are often evaluated quantitatively, experiments and longitudinal studies using quantitative methods do provide them by making use of statistical tools such as linear regressions or correlation analyzes.

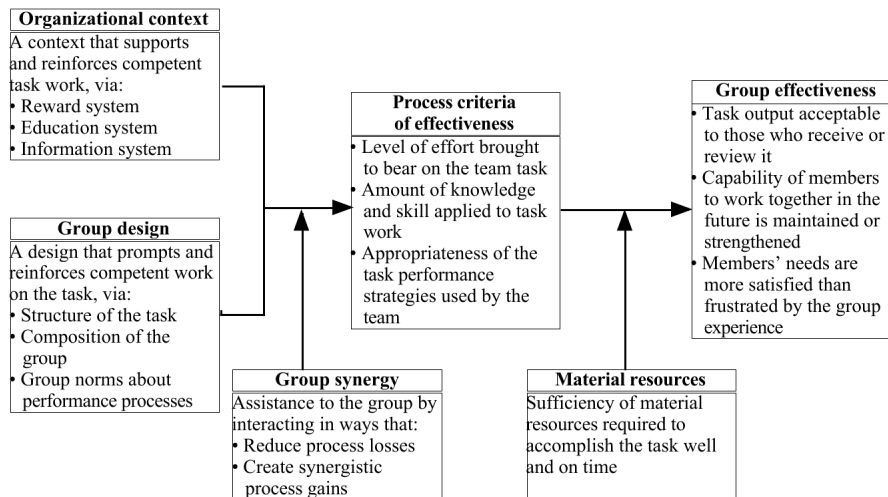


Figure 3.3.: Hackman's normative model of group effectiveness [Crowston et al., 2004]

3.1.2. Models of Team Behavior

The second branch of approaches from behavioral science elucidated in this chapter consists of *models of team behavior*. In contrast to the empirical studies presented in the last section, studies employing models of team behavior try to summarize aspects from previous work. The findings of the collected contributions are compared and discussed such that argumentatively a new theory is developed. In most cases, these theories are not further quantitatively evaluated since they often have a merely explanatory purpose and are based on descriptive studies themselves.

Normative model

As stated in the introduction of this chapter, it is necessary to understand all the complex interdependencies between a team and the quality of its problem-solving activities. For this reason, researchers have been trying for many decades to capture influencing aspects to help when setting up teams and their problem-solving environment. Richard J. Hackman for instance, developed a normative model for work teams that is displayed in Figure 3.3 [Hackman, 1987]. His model, together with other classic works such as [Steiner, 1972] and [McGrath, 1984] is a typical Input-Process-Output (I-P-O) model [Ilgen et al., 2005]. The main idea of I-P-O models is that inputs lead to processes that in turn lead to outcomes in a linear and sequential manner. Hackman's framework was one of the leading research models in the last decades and is therefore summarized in the following.

3. Team Composition

Group effectiveness of organizational teams (in contrast to teams in e.g. laboratory experiments) is very complicated to measure because very often objective measures such as the number of ideas a team generated in a creativity session in an experiment cannot be applied. Moreover, unlike experimental teams, team members of teams in organizations have to continue working with each other after performing their common task.

For these reasons Hackman defined the following three aspects as criteria for assessing group effectiveness:

- a) *Task output acceptable to those who receive or review it*: even though this is a subjective measure of quality, Hackman claims that objective measures are only available in few domains and that the choice of objective measures would thus restrict the applicability of his model.
- b) *Turnover*: This measure indicates how likely it is that team members would like to remain member of the same team for future projects.
- c) *Satisfaction in terms of personal team members' needs*: The task performed in the team should not frustrate a team member. It should rather satisfy him such that his personal needs can be addressed as well as possible. This measure, too, can be considered a subjective measure.

The proposition of the sociologist is this group effectiveness of a team depends on three factors: the **effort** invested by a team to accomplish the task, the level of **knowledge and skills** the team members use to accomplish the task and the **performance strategies** applied by the team itself to perform the task. Hackman refers to these three factors as **process criteria of effectiveness**.

As can be recognized in Figure 3.3, three other factors can influence the process criteria of effectiveness namely **group design**, **organizational context** and **group synergy**. Each of these factors has an influence on the optimization of skills, effort and performance strategies. These influences are described in the following paragraphs.

Group design refers to guidelines for structuring a team, structuring the task that has to be performed and the development of group norms within a team.

The *composition of a team* intuitively has to include talented people. Hackman claimed that in certain cases such people may be overlooked due to political reasons. It is also very important to include the right amount of people so that the task can be accomplished in the given time. Assigning too many people may also be counterproduc-

tive in terms of costs and quality of the solution. Besides technical **skills** interpersonal skills also play an important role in team composition, especially in very diverse teams. One or two individuals that are not capable of working collaboratively may impede the performance of the entire team. In line with other work in the field of team composition, Hackman also emphasizes the importance of finding the right trade-off between heterogeneous and homogeneous teams with respect to knowledge and skills so that they can effectively cooperate despite their different functional and cultural backgrounds.

Apart from composition the *task structure* has an influence on a team's output/outcomes, especially in terms of the **effort** expended. Hackman listed conditions under which people are more likely to experience their work as meaningful:

- *The group task requires members to use a variety of relatively high-level skills.*
- *The group task is a whole and meaningful piece of work, with a visible outcome.*
- *The outcomes of a group's work on the task have significant consequences for other people (e. g. the organization members or external clients).*
- *The task provides group members with substantial autonomy for deciding about how they do the work - in effect, the group "owns" the task and is responsible for the work outcomes.*
- *Work on the task generates regular, trustworthy feedback about how well the group is performing.*

When creating tasks for teams it is thus advantageous to consider these conditions, in order to increase the effectiveness of a team.

Regarding the way teams should behave in specific situations, *group norms* represent a precious tool. Group norms are a set of "rules" valid within a team. They are usually defined when a team is first composed, but can be changed after specific period of times, if a team is ready to reconsider them. The main goal of these rules is to increase the likelihood, that members of the team will develop effective **strategies** to perform a given task. For very familiar tasks these norms are tacit and usually do not need to be explicitly discussed. Norms have to be clearly specified and accepted by the team members for them to act accordingly. Furthermore, group norms should also account for the investigation and analysis of strategic planning activities and situation scanning i. e. they should encourage that these activities are performed.

3. Team Composition

Organizational context is a variable that comprises reward, education and information systems as well as material resources that can be exploited by the team.

A *reward system* is a system or an infrastructure that provides reward to the team when a task is performed well. In terms of **effort** invested into a team's problem-solving process, a reward system can be very beneficial because of its impact on motivation. Hence, an appropriate design of a reward system is not neglectable. For example, moderately difficult objectives that have to be achieved should be formulated. Research in this field has shown that teams work towards the achievement of specified objectives (e. g. a delivery date, quality or quantity to be achieved). Furthermore, feedback on the progress towards these goals has to be provided. Another important part of a reward system includes positive consequences whenever a team or team member performed very well. The type of reward depends on the specific team and environment. Sometimes, as for instance in free/open source software projects, recognition of excellence and appreciation will suffice, e. g. , reputation increases [Raymond, 2001]. To this implicit reward often also more explicit and material rewards have to be added. Irrespective of the specific reward, it is important that the team understands that the reward is given only if it performs well. Hence, rewards for moderate effectiveness should be avoided. Obviously, the reward also has to address the entire team rather than individuals even though single members might not have deserved it. Reward systems which have limited contingents should be avoided, because they induce competition or even conflict within the team or among several teams.

Whenever the **skills and knowledge** required to complete a task are not sufficiently developed, an *educational system* in the embedding organization has to help the team by means of offering the infrastructure and materials to acquire new knowledge and skills. Moreover, this educational system has to be easily accessible to the team.

An *information system* in the organization supports the team, among other things, during the monitoring process, the retrieval of relevant data and for the communication with other organizational units. More precisely, with an information system the performance situation can be monitored and in turn the **performance strategies** adapted if necessary. In addition, it is also of high importance to know about possible constraints that exist for the solution of the problem or information about the people who will receive, review or use the solution of the task performed. Another function an information system should provide is support for the analysis of performance strategies, i. e. for finding out what the consequences may be, if a specific performance strategy is chosen. For example, what would be the consequence in the case some subtasks of the main task are parallelized?

Group synergy is defined by Hackman as “[...] *group-level phenomena that (i) emerge from the interaction among members, and (2) affect how well a group is able to deal with the demands and opportunities in its performance situation*”. Group synergy thus does not occur in nominal teams. A nominal team is a group of people who work separately on a task and whose result is the union of the results of all individual team members. The result is thus not achieved through interaction. In contrast, group synergy refers to group-level phenomena that emerge from the interaction between team members and is said, if it is positive, to substantially improve the effectiveness of teams because it helps to overcome process losses. Negative group synergy, conversely, amplifies existing process losses and thus reduces team effectiveness.

By helping to find new innovative ways to *minimize process losses* that arise in any kind of interaction pattern, positive group synergy fosters **effort** among the team members. For example, costs that occur for managing a team or coordinating their work can be reduced. At the same time it may also mitigate the “social loafing” phenomenon [Latane et al., 1979]. “Social loafing” is defined as the effect that occurs when a team is too large for the task to be accomplished. In that case due to the high amount of team members, each single team member feels less responsible for the team’s output than if the team consisted of fewer members. Moreover, a team that is not affected by social loafing can profit from a greater *shared commitment* to the degree of real team spirit. This effect is present when individuals value their team membership and therefore work harder.

Literature collected by Hackman (for example [Johnson and Torcivia, 1967] and also [Thomas and Fink, 1961]) documents that groups often weigh the importance of contributions on the basis of task-irrelevant attributes of team members such as gender, talkativeness or verbal dominance, which clouds the really helpful contributions based on the appropriate **skills and knowledge**. In this context, interaction is a powerful way to *improve individual skills*: through interaction team members can learn from one another e.g. new concepts, theories and experiences.

As mentioned above, interaction lets new potential emerge within a team. Also new innovative **performance strategies** may be thus created that improve the effectiveness of a team. For example, through interaction a team may find out how to exploit certain resources that, up to that time, others had overlooked. Finally, problems in the implementation of strategies can be overcome through interaction, for instance, by solving emotional conflicts among the team members.

Material resources are necessary in addition to the above described variables for a team to be effective. Although a team is very efficient and does not experience any kind of

3. Team Composition

process losses, if human resources or material resources (e. g. tools, equipment, money) lack, its performance certainly will suffer.

Intervening Process Theory

Demographic diversity has been identified in literature as one of the critical factors influencing team performance. Pelled, for example, developed a theory based on several types of demographic diversities ([Pelled, 1996]).

She analyzed related work in this field (among others Ancona and Caldwell's work presented in Section 3.1.1) and reasoned that demographic differences can no longer be neglected when analyzing the effectiveness of teams. Previous work in this research field however gives ambiguous results with respect to the specific influence of diversity on the output/outcomes of a team. In Pelled's opinion the reason lies in the low discrimination of different demographic variables. For instance, research analyzing the influence of gender on team outcomes produced ambiguous results. For instance, Hoffman and Maier found evidence that heterogeneous mixed-gender groups tended to produce higher quality products [Hoffman and Maier, 1961]. In contrast, a study of Kent and McGrath indicated that homogeneous gender groups generate more original products than heterogeneous ones [Kent and McGrath, 1969]. Therefore, she developed a model which contains and analyzes a more detailed classification of demographic diversities. At the same time she tries to take into account that the opposite approach (wider classification) allows for a greater explanatory power [Bacharach, 1989]. In the following paragraphs a brief description of the model depicted in Figure 3.4 is given.

Pelled noticed that two frequent dependent variables (in this case variables that describe the output/outcome of a team) appear when investigating the effects of demography on a team: *turnover* and *cognitive task performance*. Consistently with the use in Hackman's model introduced in the first part of this section, turnover refers to the portion of members leaving a team voluntarily or involuntarily. Cognitive task performance refers to "... performance on tasks that are more mentally than physically challenging, e. g. decision-making, problem-solving, or creative idea generation". Due to their wide usage in this field Pelled investigated demographic diversity aspects in relation to these two output variables.

She then identified relevant variables regarding turnover and cognitive task performance in literature. These variables are age, gender, race, group tenure, organizational tenure, functional background and education. Among them, age, gender and race are considered as high-visible/low job-related diversity variables. Group tenure is regarded a high visible/high job-related variable. The remaining three variables are characterized by

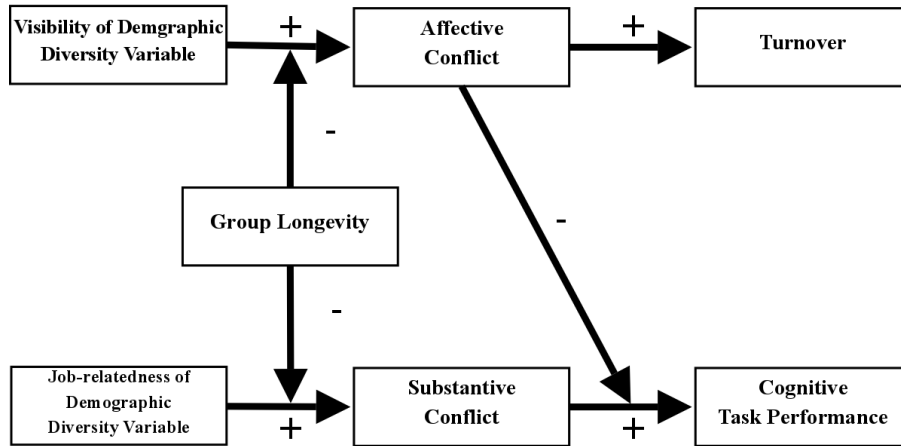


Figure 3.4.: Pelled's model on demographic diversity and work group outcomes [Pelled, 1996]

low visibility and high job-relatedness. None of the selected variables is classified as low visible and low job-related.

This categorization represents the basis of Pelled's theory. She claims that variables with high job-relatedness will foster substantial conflicts while, the more visible a particular type of diversity is, the stronger its relationship with affective conflicts will be. *Substantive conflicts* are conflicts that emerge when having different opinions with respect to the task that has to be accomplished. Up to a certain degree many researchers claim that substantive conflicts improve team performance. In contrast, *affective conflicts* are said to lower team performance and team satisfaction. However, more recent research (e.g. [De Dreu and Weingart, 2003]) shows that these implications need to be further investigated.

As a moderating variable, Pelled defines group longevity which is the amount of time, group members spend working together. She claims that on the one hand, the duration of the collaboration may reduce the negative impact caused by visible demographic diversity but on the other hand may also reduce the positive impact of substantive conflicts. This can be caused for instance, by stereotype thinking based on visible aspects such as age or race.

The last proposition in this paper states that affective conflicts within a team negatively affect the relationship between substantive conflicts and performance. Literature in that field of research states that whenever threat and anxiety are experienced, difficulties in processing new or complex information arise. This occurs, for example, when new pieces

3. Team Composition

of information are generated during substantive conflicts.

Summary

The last sections indicated that models on team behavior are less operationalized than empirical studies. Hackman's work is very valuable for identifying aspects that influence a team's output but are not directly related to the team itself. There seem to be aspects that influence a team from outside that can range from the implementation of a reward system to norms used within a team. Task specification belongs to this kind of aspect since it is not a team members' property and thus influences the team from the outside. Hackman as well as Pelled used aggregated team member properties to express their theories such as the heterogeneity of functional backgrounds or the interpersonal skills. In accordance with the previous empirical studies various different output/outcomes were used in the models of team behavior presented. These models also showed the existence of variables that are influenced by inputs and in turn influence outputs (e. g. substantive conflicts). These were not present in empirical studies. Interestingly, the investigation technique used in these models of team behavior mainly bases on literature research and chains of reasoning.

3.1.3. Analytical frameworks

Not only empirical studies (Section 3.1.1) and socio- psychological models of team behavior (Section 3.1.2) provide information on how to compose good teams. Other sources originating from behavioral sciences that contain guidelines for effective team compositions are analytical frameworks such as game theory or linear optimization models.

Mello and Ruckes, for instance, describe ideal team compositions using game theory ([Mello and Ruckes, 2006]). To be more precise, they investigate the impact of heterogeneity within a team on team performance. Performance herein is specified as the cumulated payoff of the team. The scenario they use to analyze the problem in an exemplified manner is a simple two-person hierarchy in which a superior can choose a subordinate who is similar to him (i. e. resulting in a homogeneous team) or different from him (resulting in a heterogeneous team) in terms of specific properties. As an example for these properties Mello and Ruckes indicated *training* (as a source of knowledge) and *organizational tenure* (as an indicator for the skill of correctly judging opportunities). Due to evidence found in literature they infer that a team needs a team leader such as a CEO or a division head.

In the proposed scenario the two team members (superior and subordinate) may acquire information about a prospective project's payoff in order to decide whether they

should accept it or not. Depending on their similarity, they will have access to the same information sources or not. Mello and Ruckes demonstrated that in cases of high uncertainty and rapidly changing environments a heterogeneous team is more valuable, because the variety of the information collected by a heterogeneous team may be larger than the variety collected by a homogeneous team and thus, may be more convenient to complement the team leader's information. This implies that firms operating in more uncertain environments should tend to fill managerial positions with people from outside the organization. Firms in stable environments should hire managers from inside the organization.

Mello and Ruckes's work moreover shows that if the probability that the team leader can obtain information about a prospective project's payoff is low, a homogeneous team (also by means of project preferences) may lead to a higher expected payoff. The same statement is valid if the probability that the team leader can obtain a lot of information about the project is high. In this case the principal hires the subordinate just because of the effort the subordinate will invest in this project. In the other cases (i. e. when the probability is either not very high or not very low) a heterogeneous team has a greater expected payoff.

For the detailed game theoretical modeling of these statements refer to Mello and Ruckes's article [Mello and Ruckes, 2006]. There exist also other game-theoretical models for team formation and composition. For instance, Cai and Kock present an approach based on evolutionary game theory [Cai and Kock, 2009].

3.1.4. Summary

With respect to the goals defined for this chapter this section has shown that suggestions from behavioral sciences for effective team composition can be categorized in empirical studies, models of team behavior as well as analytical frameworks. For the first category quantitative evaluations methods are often used to obtain significant indications from past team work. In models of team behavior results produced in related work are aggregated, discussed and analyzed. This analysis leads to models explaining team composition and its influence on team performance. For analytical frameworks generic mathematical models are used to explain the team composition model. As an example, a game-theoretical model has been introduced. The works presented in this section reside in the domains of sociology, psychology, management and organizational behavior sciences.

Many aspects were investigated that range from emotional factors, to demographic properties and strategies within a team. Some of these aspects can be obtained directly from a team's properties, some are part of the team work context. Moreover, some aspects

3. Team Composition

mediate between team properties and outcomes such as in the case of conflicts in Pelled's work. Some of the presented studies also tackle the problem of team output that has to be measured when assessing team performance. Various possibilities for types of outcomes have been proposed such as turnover or customer satisfaction. Despite all the indications and guidelines research provides, only some specifies more detail on operationalizations for variables and used concepts. Therefore, in order to implement them in a real world scenario some cases require additional steps in this direction. All works introduced in this section investigate how specific aspects influence the different types of team output and outcomes.

3.2. Approaches from practice

The previous sections provide an insight into current literature and research on team composition. In spite of the large amount of literature available, the team composition problem in practice is by far not solved, because precise, operationalized guidelines tailored to the needs of project and product managers who have to compose teams are lacking. For this reason, firms apply their own composition approaches that base on their own experience and tacit knowledge.

In this section these *guidelines* and *best practices* for composing teams in enterprises are presented. The first subsection introduces interviews that have been conducted for this purpose. It first presents the survey and the selection of adequate interview partners and then, the empirical results of the interviews that show which team composition approaches are utilized in practice. For details on these conducted interviews refer to [Brocco et al., 2011a].

3.2.1. Interviews

In order to analyze the team composition problem from a more practical perspective, experts in this field had to be identified, contacted and interviewed. Interviews make it possible to get an understanding of which strategies are used to compose teams and, occasionally to learn what tools are currently used to accomplish this complex task.

Unfortunately at this point of time, only very few companies declare to actively adopt the open innovation paradigm. Thus, possible participants had to be selected according to other criteria:

First, the domain of these experts had to be innovation-driven to guarantee an anal-

ogous scope as in the open innovation case (since the primary goal in open innovation is, as the name says, to innovate). Second, the companies in which these experts act, had to be familiar with interaction patterns that also occur in open innovation. This may, for example, be the case for very large companies that have to face the problem of cooperating across countries, organizational units and/or departments (also known as *company internal open innovation* [Reinhardt et al., 2010]) or smaller companies that have cooperations with freelancers, other companies, customers, etc.

Given these criteria ten experts in team composition (nine product managers and one software engineer) of different ICT companies in Germany were selected. Two of them had less than 5 years, while the others had more than 15 years of professional experience. The company sizes ranged from small (less than 100 employees) to large (up to 180.000 employees). All the companies are internationally active. The interviews were focused on the ICT sector, because it is known to be reasonably innovation-driven due to the actual high competition in the market. Companies in this sector are also more likely to be open to novel concepts, technological trends and tools. Hence, it can be expected that the trend towards open innovation will show up earlier here than in other lines of business.

To summarize, the selection of the companies was based on these two facts: First, companies from the ICT area are said to be reasonably innovation-driven. Second, large companies have to manage the same complexity as open innovation networks. For the small companies it was a priori known that they included external sources in their innovation process such as in the open innovation case. Therefore, the findings may also be applied to the open innovation networks case.

The survey was designed as a semi-structured interview [Bortz and Döring, 2006, p. 239] concerning open innovation and creativity. Several questions on the composition of teams for projects in general and, more specifically, creative teams were asked.

The first part of the interviews consisted of questions on the company and interviewees themselves. For example, it was of interest to know how long the participants had been active in this domain and how many years of experience they had had. Then, an introduction on open innovation and creative processes was given. To ensure that the interviewee understood the presented notion of open innovation and creativity, they were asked to recall related examples occurring in their daily activities. Next, three major topics were investigated: (i) creative processes, (ii) situations in which creative ideas were generated and (iii) team compositions for projects and creative processes. The results presented in this section concern the team composition part.

An overview of the interview structure can be gained in Appendix A which also con-

3. Team Composition

tains the original German statements. These statements are referenced next to the corresponding English translation listed in this section. Statements not included in full length in this section are referenced with an according key associated with the original German statements in the appendix.

3.2.2. Empirical Results

The interview results were evaluated using Mayring’s methodology [Mayring, 2007]: The interviews were transcribed and coded according to the predefined categories “team composition criteria and strategies”, “challenges in team composition”, “support tools and mechanisms”. They were conducted in German and lasted for about one hour each.

As a first step, the interviewee’s answers were paraphrased and brought to a standard written language. Through the coded sentences contained in the transcribed interviews, more generalized statements for each defined category were derived. The generalized and reduced results of this investigation presented in this section belong to the categories “team composition criteria and strategies” and “support tools and mechanisms”. Results from the other category are presented in Chapter 4.

Not surprisingly, all interviewees mentioned competencies as one of the criteria that matters when composing a team (e.g. [s11,s15]). Mainly this was mentioned because of the expectation that people from a specific domain can contribute more to solve problems in that domain due to their experience. In this context, databases (non-IT as well as IT-based) containing information about skills are often applied in order to facilitate the selection process ([s12,s13,s14,s21]). In relation to the competencies needed, experience and tenure also play a role ([s15,s17]).

The interviewed project managers also stated that they try to find people with “social competencies”, i.e. whose personality fits with the other team members in order to avoid emotional conflicts ([s15]). This is in accordance with some of the models described in Section 3.1.2. Which personalities are compatible was however ascertained based on personal experience rather than formalized theories (e.g. on personality tests such as the Myers-Briggs Test [Briggs-Myers, 1980]).

Another component that plays a role when composing teams were social networks. In this case social network does not only specifically denote social networking platforms like Facebook or LinkedIn but also implicit networks arising from collaboration or organizational structures. People are more likely to invite persons to creativity sessions that are part of their social network ([s22,s32]), the main reason being the compatibility of specific

personalities within creative team work. Social networks play a similar role when the actors are distributed across organizational units, even outside a company's boundaries i. e. in the open innovation case ([s16]). Among the interviewees several people rely on traditional non-IT support such as the combination of address books and phone to find and contact appropriate people ([s33]).

To include unknown people depending on whether they are company internal or external, brokers, curricula vitae or other sources that can assess competencies are used. For example, project managers acquiring developers from the open source field, first ascertained the quality of their contributions in the published source code and/or of their posts in forums and mailing lists ([s24]). Despite the available support systems, the last step of the "recruiting" process is usually performed without additional support in a face-to-face meeting ([s24,s34]). The reason seems to be the difficulty in assessing personality and social aspects in a formalized and operationalizable way. Some of these aspects can however be neglected for short term tasks (such as creativity sessions) as one of the interviewee stated: *"I remember a situation where I knew he [a colleague] is very uncomfortable to discuss with. Nevertheless I invited him since I knew he could significantly contribute to the solution of this problem"* [s1]. As a result it can be said that in some cases composing teams for short tasks (e. g. creative problem solving sessions) may not require aspects such as social skills as required in long term projects, especially when the quality of the result is of high importance.

Particularly of interest were answers addressing the open innovation scenario. For instance, the interviews showed that certain situations ([s25]) call for external expertise: *"I remember one occasion where we needed external help. That was a problem that we could not solve internally [...] We called an expert from outside who contributed to the solution of the problem."* [s2].

The portfolio of team composition strategies varies a lot in practice. Even within the same company several strategies to compose a team are used because of the varying contexts or task specifications. For example, the same product manager stated that for a short term project lasting 10 days: *"I would assign to this job two developers that are well attuned with each other, no matter if they are very proficient in the needed skills"* [s3], while for a brainstorming session on a new product idea that has to be performed within 30 minutes she would choose: *"[...] a senior developer, somebody thinking in a structured manner that wants to try out new things, someone that offers constructive criticism, someone that has experience with the system as a whole and finally, someone that has decision power. Moreover, all team members should be of the same language area"* [s4]. For a project across her companies' boundaries the same product manager would choose : *"... a decision maker, a potential customer, a technical and business representative of each*

3. Team Composition

partner organization, experts of the domain, one moderator and a market researcher” [s5]. Obviously, many strategies are not formalized as one interviewee stated: “I use several team composition strategies, but none is formalized. It is not something like a tool set [...], it is estimation, experience and gut feeling.” [s6].

Regarding the appropriate blend, people often seem to prefer a mix of experienced and inexperienced people in their team for the solution of creative tasks in order to balance the trade-off between new ideas and feasible ideas ([s31]). Moreover, people from inside the organization seem to be categorized as more trustworthy, regardless of the specific department they belong to. Therefore in certain cases heterogeneity in terms of external sources is avoided even though the quality of results would improve ([s26]) as recognized by the participants during the interview ([s27,s30]). Moreover, no evidence for support systems addressing this specific team composition issue could be found.

It transpired that gut feeling was often decisive for the composition of teams which is very likely tacit knowledge acquired from past experiences ([s6,s28,s29]). This aspect has been classified as very important by most of the interviewees.

Surprisingly, past output/outcomes do not seem to directly affect the project manager’s choice. If data on produced output is available, it is usually just taken into consideration in a second step to refine the search. However, this generated output/outcomes data is always associated with the environment and situation the team performed in. Thus, project managers try to create a context-aware comparison mechanism for teams. For example, one project manager stated: *“There are also many difficult projects or projects where the customer is very demanding. In that case the performance was not outstanding but they [the team members] were stretched to the limit and were very creative.” [s7].*

Apart from all the criteria specified by the team managers, organizations also heavily depend on organizational constraints such as availability ([s17,s18]), location ([s19]) and cost of people ([s20]). Therefore these rather simple criteria usually prevail against the other criteria for team composition. Because these rather simple organizational aspects most often prevail, the best composition of a team cannot always be ensured. They are usually regarded with the help of project management tools (e. g. Microsoft Project) that visualize some of the organizational constraints.

3.2.3. Summary

The experience of project managers and team composition experts working in companies has shown that many constraints influence the composition of teams. The interviews even

leave the impression that most team composition strategies cannot be adopted due to the critical impact of organizational constraints.

Nevertheless, several different team composition strategies do seem to exist in practice. Apart from considering skills which are almost always of importance, the distribution of team members' different properties appropriate to the task is considered. Sometimes, also aspects related to social networks and friendship and trust relations are regarded, i. e. interpersonal aspects concerning not only an individual team member as in the skill case. The context and constraints however, play a major role while hierarchy and personality seem to be important but not always critical.

This section also deals with the aspect of current IT support for team composition. The interviews provided evidence that only very generic tools are used such as project management tools or skill databases which confirms that the team composition strategies that are most often applied are based on availability and skills, although many others are applied depending on task and environment. It was noticeable that no tailored support for team composition is used in the companies that were interviewed.

3.3. Algorithmic Approaches

Apart from behavioral sciences (Section 3.1) and practice (Section 3.2) related work in the field of algorithmic team composition offers another set of approaches. In order to build a team recommendation system for open innovation networks, it is indispensable to analyze whether there are appropriate algorithmic approaches that can be used to solve this task. This section provides an overview of several different solutions found in a literature review that cover the team composition problem from different perspectives. The categorization represented by the subsection titles refers to different semantic perspectives on the team composition problem rather than on different algorithmic techniques.

For instance, the first subsection describes team composition algorithms that make use of semantic web technologies (i. e. ontologies) to compose teams. The second introduces approaches that exploit social networks while the third subsection comprises an agent-based methodology to model human behavior such that team activities can be simulated with the help of multi agent systems. In terms of compatibility of personalities within a team Subsection 3.3.4 describes a prototype for assembling effective agile software development teams.

3. Team Composition

3.3.1. Semantic-based

One of the most popular ways to compose teams consists in matching the skills of team member candidates with the required skill set. This can be done in several ways. The most precise is to use ontologies which describe the relations between skills. These ontologies allow for the computation of similarity between skills so that it can be expressed, to which extent a team complies with the task specification (which in this case is a set of required skills). Due to their main dependency on ontologies these approaches are labeled *semantic* technologies for team composition.

Colucci et al., describe a semantic-based, greedy approach to compose teams based on a self developed competence ontology [Colucci et al., 2005]. In a first step, they extract skills from text files and curricula vitae that match their developed skill ontology. Through a greedy algorithm, they then compose teams that cover all the requested skills while also minimizing their size.

Similarly Richter and Tolksdorf ([Richter and Tolksdorf, 2009]) attempt to minimize the number of team members. However, they set limits between 4 and 10 members. Furthermore, the level of expertise in specific skills is also considered more accurately than in Colucci's work, i. e. levels of expertise are explicitly regarded. In addition, concepts such as the heterogeneity of competencies in a team and the overlapping of skills that enable more fruitful discussions within a team are used as measure to rank a team. The similarity assessment of skills is also computed with the help of an ontology provided from external partners. Finally, the availability of single team members is incorporated. The chosen evaluated criteria are then linearly combined with parametrized weights in order to result in a final value denoted as *degree of suitability*. This measure is used to rank teams in order to support the project manager when choosing a team.

3.3.2. Social Network-based

The effectiveness of teams is also directly related to the relationships between team members. Evidence for this can be found in practice (see Section 3.2.1) as well as in sociological models (see Section 3.1.2). To address this particular aspect of team composition researchers try to compose teams using social networks and to augment them with competence profiles such that an effective team in terms of competencies as well as e. g. communication costs can be composed. The background assumption is hence that by knowing people (e. g. because of past work experiences) communication costs in terms of time can be reduced.

The problem of assembling effective teams by means of the above described criteria is addressed in [Lappas et al., 2009]. The authors show that the problem of team composition (or team formation) is NP-complete. Lappas et al. attempted to operationalize the problem of finding a team by minimizing their communication cost through two different paradigms. The first bases on the optimization of the diameter of a graph (the largest shortest path between any two nodes in the graph). The second tries to find the minimum spanning tree of the graph. The edge of the graph are communication links between team member candidates that are in turn represented by the nodes. The weights of the edges represent the communication costs. Both problems, are proven to be NP-complete. To solve the problem according to both paradigms, several algorithms are applied. Moreover, the competence matching algorithm which is also part of their approach is an instance of the Set Cover problem [Karp, 1972] and is therefore NP-complete as well.

In their evaluation they compare the results of the various algorithms by means of cardinality of the team depending on the amount of skills required, the communication costs and the number of disconnected teams composed. In terms of the latter evaluation criteria, some of the algorithms presented produce disconnected teams even if connected teams were in the solution space. The evaluation was conducted on the DBLP³ dataset where keywords of papers represented the skills and the communication costs were computed based on the number of papers written by authors in common.

Li and Shan extended this approach by allowing an arbitrary size of experts (with regard to specific skills) in a team so that a project manager can, for example, state that he needs three experts in a specific field [Li and Shan, 2010]. Nevertheless, one goal of their system is to minimize the cardinality of the team.

3.3.3. Behavior-based

A very particular approach to support decision-makers when assembling teams is to simulate a team's behavior thereby helping the project manager to decide based on this simulation whether a team is appropriate or not. Gonçalves et al. followed this approach by employing agents in a multi-agent system that simulate human behavior during a brainstorming session [Gonçalves et al., 2006].

The researchers modeled the brainstorming problem-solving process with a set of states, actions, goals, properties and classes. Regardless of the application domain, they try to simulate human behavior by first recording interactions of human teams in a case base (see Section 2.2.2.3 for an introduction on case-based reasoning) and then using this

³<http://www.informatik.uni-trier.de/~ley/db/>

3. Team Composition

case base to decide which actions a representing agent should perform. The agents have a BDI (belief-desire-intentions architecture, please refer to [Wooldridge, 2002]) based mental model that is influenced among others by perceptions and nuclear internal state variables such as fatigue or trust in a particular agent. The simulation is then started and each problem-solving process is evaluated according to the criteria:

- Shorter brainstorming sessions in terms of time are better.
- Brainstorming sessions with fewer states are better.
- Brainstorming sessions with more cooperation between the actors are better.
- Low levels of indecision, conflicts and divergence are better (computed (i) by identifying deleted states, (ii) by the amount of ramifications and reunifications in the state graph).

On the foundations of the case base and the formalized model other humans can also be simulated and results in terms of the above specified criteria can be predicted. With the support of these predictions the project leader can decide which the team to select.

3.3.4. Ideal combination of attributes

Skills and competencies can be regarded as attributes of specific team members. One way to take these into account is matching given a task that is described by means of needed competencies. Instead of only matching skills, there may be also specific combinations of skills that increase team performance. The same can also be valid for other types of attributes such as the combination of specific roles in a team, personality or perhaps nationality.

An interesting system that aims at facilitating the optimal combination of personalities comes from the domain of agile software development where team cohesion plays a major role. Conflicts among team members may lead to delays in the schedule or to poor product quality. Therefore, properties that are related to personality types play a more important role than in other teams that, for example, perform only short tasks. For this purpose Licorish et al. developed a system with the goal to identify possible personality incompatibilities [Licorish et al., 2009].

Several different theories for assessing personalities are known. Probably the most popular theory has been developed in the early 1920s by Carl Jung [Jung, 1971]. His work on personality types stated that each individual can be categorized by means of

eight psychological types which are derived from the way specific psychological functions are performed. The psychological functions in this context are categorized into perceiving functions (*sensing* and *intuition*) and judging functions (*feeling* and *thinking*). These functions can be modified by specific attitude types, namely extroversion and introversion. Thus, people can be categorized given the four functions and the associated attitudes, resulting in eight different types (e. g. introverted intuition, extroverted feeling etc.). Briggs and Myers [Briggs-Myers, 1980] elaborated and extended Jung's theory to 16 types and developed a corresponding test for assessing personality. Today this test is very popular and used in academia as well as in industry [Chen and Lin, 2004]. Given this kind of characterization compatible personalities can be found that are said to improve team performance.

Correspondingly Licorish et al. have created a software project management system in which they integrated a questionnaire that should determine software engineers' personalities. Given the theories of personality compatibility this can facilitate the task of choosing compatible team members.

3.3.5. Clustering-based

A more generic approach towards team composition has been put forward by Hlaoittinun et al. [Hlaoittinun et al., 2007, Hlaoittinun et al., 2008]. They propose to use array-based clustering in order to group aspects from the context with aspects belonging to the team itself. Array-based clustering is a clustering method that reorders rows and/or columns in a matrix according to specific criteria.

In an example Hlaoittinun et al. apply clustering to tasks and team members so that people with similar competencies with respect to a task's requirements are grouped together and then selected.

They first defined a project as the union of several different tasks. These tasks in turn are defined by a set of skills with corresponding expertise levels. In a second step a so-called "competency matrix" is computed which represents the relationships between a set of tasks and a set of actors. Necessary steps for this computation are the assessment of needed and available skills (i. e. the team member's skills).

An array clustering mechanism can then be applied to produce task-member groups that fulfill a minimum required skill expertise level.

Depending on whether this step has been performed or not, an objective function tries to minimize project costs (that comprise salaries and learning costs) within a cluster (local

3. Team Composition

optimization) or among all possible team members (global optimization). An underlying assumption is that each team member performs exactly one task.

Hlaioittinun et al. evaluated their approach with an example in the mechatronics domain. First, the algorithm with clustering and the local optimization step were tried. Afterwards only global optimization was applied (task assignment without clustering). For the first case the costs of the team (also taking into account salaries) were higher than the second case. However, in the first approach with clustering competence levels within a team increase more than in the second case without clustering. Intra-domain expertise can hence be maintained with the clustering approach. In other words the second algorithm may be chosen to decrease costs while the first one is appropriate to keep intra-domain level competencies due to the competence clusters that are created by the array-based clustering technique.

3.3.6. Latent semantic analysis

Similar to the recommendation approach elucidated in Section 2.2.2.6 latent semantic analysis is based on the fact that some “hidden” factors influence the choice or rating of specific items. This may also be the case when recruiters choose specific persons for given tasks from a set of job-applicants.

Team composition is considered by Malinkowski et al. as a two dimensional problem since it regards person-task fit as well as person-team fit [Malinowski et al., 2005]. For this reason they developed a recommender system that takes into account both elements separately.

First, with respect to the person-task fit, Malinowski et al.’s approach uses probabilistic latent semantic analysis (PLSA) [Hofmann, 1999], which is related to the latent factor approach in Section 2.2.2.6 for estimating the assessment of a recruiter. More precisely, the probability is estimated, that a recruiter x rates an attribute a belonging to a job-applicant y with the value v . Thus, the difference is that instead of matrix factorization as in Section 2.2.2.6 they use conditional probabilities and assess the latent factors through EM (Expectation Maximization, confer [Dempster et al., 1977]).

Malinowski et al. choose an arbitrary number of latent factors⁴ (say e. g. n) and train their model consisting of job profiles and people with past ratings (of different recruiters) in order to compute the latent factors. The model helps to assess how well a person fits to

⁴Note that models with a reduced number of latent factors will in general require less data to achieve a given accuracy and are less sensitive to overfitting [Hofmann and Puzicha, 1999]

a specific task whose requirements are expressed in an attribute tuple (e. g. “mathematical skills”, “diploma grade”, “1.0”, “University of Frankfurt”). The result of the application of the PLSA model is a list of recommended team members for a specific task.

In a second step trust values are integrated in the recommendation process in order to deal with the person-team fit. These values are either derived from explicit trust “ratings” or computed based on the assumption that people that have specific preferences are more likely to trust people with similar preferences to a higher degree [Ziegler and Lausen, 2004, Montaner et al., 2002]. This is again computed with the help of latent factor analysis. By applying an averaging technique to aggregate trust values (augmented by a confidence value gained from the number of co-rated jobs) with the values derived from explicit ratings it is possible to compute missing trust values among team members and afterwards generate according recommendations.

In the last step Malinowski et al. combine the two scores of the two lists (i. e. trust scores and predicted ratings) with two parametric weights α and $(1 - \alpha)$ to a final measure which is used to produce a final rating. Thus, this approach can be regarded as a weighted hybrid recommender system (refer to Section 2.2.2.1).

3.3.7. Summary

The described algorithms were developed in the field of information systems and computer science and were only partially based on empirical evidence. In fact, different assumptions derived from individual studies (such as those described in Section 3.1 and 3.2) were used to develop each system. As a consequence, they may only fit a particular scenario. Apart from this, most of the works do not contain quantitative evaluations. In contrast to approaches from behavioral science and practice presented in the previous sections, this section contains *most* of the algorithmic approaches that have been found in the literature review (not only a selection of it). Hence, it can be said that this field of research is still rather unexplored compared to research in the behavioral sciences.

The different systems described in this section each base on a specific assumption on how teams should be composed. For example, social network based approaches implicitly rely on the fact that people who know each other can communicate more efficiently and this in turn affects team performance positively. Meanwhile, less emphasis is given to the environment or context compared to the approaches of Section 3.1 and 3.2. Thus, the focus lies mainly on the team members themselves rather than the environment, task or context.

3. Team Composition

In terms of operationalization, these approaches are described in a detailed way such that a reproduction of the team composition strategies therein is possible without further effort. A simulation technique has also been found in related work that tries to model human behaviour with the help of agents (Section 3.3.3). The simulation in the end provides indications on influence factors for good teams that are similar to the indications provided by the studies in Section 3.1.

Several aspects were investigated that range from interpersonal aspects to the personality type of team members. These were computed by matching certain properties or by traversing graphs representing team members. In some cases preferences on team compositions and influencing factors were derived from past preferences of other users. It was of interest to ascertain that in some cases instead of the actual expected team performance, the user preference was used as criterion to evaluate a team (e. g. in the latent semantic analysis approach).

3.4. Summary, Analysis and Comparison of Approaches

As has been extensively presented in the previous sections, team composition is a very complex task, that is dependent on various aspects and variables.

Section 3.1 presented the current state of behavioral research on team performance and effectiveness with the help of a few selected examples. Three different types of work were distinguished: empirical studies (Section 3.1.1), models of team behavior (Section 3.1.2) and analytical frameworks (Section 3.1.3).

Empirical studies have an a posteriori perspective on team composition and team performance. In many experiments or field studies specific attributes of a team and of the environment are observed, measured and collected to find statistical indications for why a team performed as it performed. For instance, correlations between input variables such as the heterogeneity of gender and output variables (e. g. number of ideas generated) can be calculated and evaluated. In contrast, case studies often analyze team work qualitatively thereby providing insight into more hidden aspects of team composition that can be used to develop new valuable research questions.

Models of team behavior are characterized by an a priori perspective that does not make use of statistical analysis to describe team performance and optimal team composition. Instead they are based on detailed, comprehensive literature reviews, that categorize different components. The information drawn from the review is then usually argumentatively integrated into a (new) theory for team composition and new research questions in

this context.

Analytical frameworks such as the game theoretical example illustrated, try to model the team composition problem with the help of mathematical and/or probabilistic frameworks. As an example a team composition model was introduced that uses game theory to explain interactions between subordinates and principals within a team.

Section 3.2 provided insight into the best practices of team composition adopted by project managers and HR experts. In a series of semi-structured interviews with ten selected companies of the ICT sector in Germany several criteria and conditions for composing teams were determined. Unlike for the other two domains (approaches from behavioral science and algorithmic approaches) the portfolio of tools used to support the assembly of teams for different type of tasks (e. g. creative problem solving tasks, long-term projects) was also described.

Section 3.3 included current algorithmic approaches for recommending teams that were collected from informatics and information systems literature. It turned out that all algorithmic approaches discussed are based on exactly one underlying theory for team composition: One approach described uses personality types as indicators for effective teams. Other approaches focus rather on required skills or competencies as the main criterion. Some of these approaches try to limit the number of team members needed by using greedy algorithms (i. e. their team composition strategy implies the fewer team members, the better). Others, extend their approach by adding other criteria such as communication efficiency.

Despite the above summarized heterogeneity of methods to compose teams, all have one aspect in common: they try to find or use an appropriate strategy to compose a team. The strategies vary depending on the use case, the context or simply the preferences of the project manager and can be regarded as team composition strategies or **team composition models** (which is a term commonly used in team composition literature). For this reason this thesis treats the terms “team composition models” and “team composition strategies” as synonyms.

Definition 1 (Team Composition Models) *Team composition models are guidelines and strategies that manifest which aspects and variables of a team, context or environment affect specific output/outcomes of a team to what extent.*

Different abstraction levels and degrees of formalization can be recognized among the different classes of approaches. Formalization in this case is defined as the level of abstraction adopted for specifying the team composition model. To be more precise, the

3. Team Composition

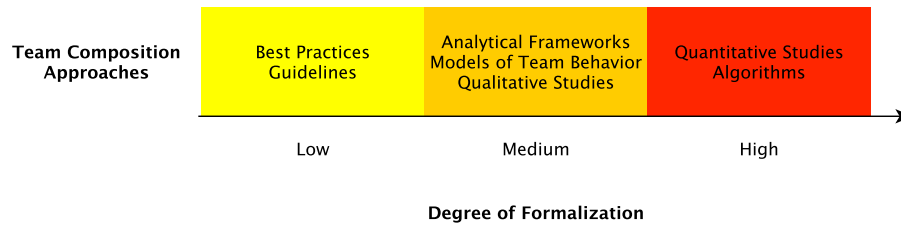


Figure 3.5.: Classification of related work based on the degree of formalization

following classes of formalization degrees in related work could be identified:

Low Level: This category comprises approaches that lack a consistent specification of variable sets i. e. only very abstract concepts are used to describe these approaches. This in turn means that the variables used to compose teams are not specified and no precise operationalization directive is defined. Relationships and dependencies between variables are also not always consistent and unspecified.

Medium Level: Approaches that have a medium level of formalization specify an exact set of influence factors and their relationship. However, these types of approaches do not define any operationalization specification and can also not be directly mapped to an algorithmic system.

High Level: Whenever an exact specification of variables, influences between variables and their operationalization is given, the approach provides a high level of formalization. In general this type of approach can be mapped more easily to an algorithmic approach (some of these types of approaches are already implemented in algorithms).

Within the high level category approaches can be distinguished that precisely specify the exact type of relationship between influence aspects. For example, it may be specified whether the team size has a more linear relationship with team performance or rather an inversely U-shaped one. In the latter case team performance increases with team size up to a certain point and decreases afterwards.

Figure 3.5 depicts the classification of the observed related work in the above described formalization degree scheme.

Algorithmic approaches and studies and experiments applying descriptive and statistical methods are formalized in great detail. Algorithmic approaches are formalized since they need to map rather abstract concepts of team composition to an IT system in order to produce results. Empirical studies based on *quantitative methods* need a precise formalization for a statistical representation and a following test of hypotheses. Thus, both

approach types are categorized as belonging to the “*High Level*” class.

In contrast to that, approaches applied in practice are almost not formalized at all since they often base on tacit knowledge and experience that is difficult to make explicit. Sometimes included aspects are not operationalizable at all because too many details are missing. The lessons learned from past tasks or projects are usually not analyzed with formal methods in companies. Thus, often the only way to profit from them is human brainpower. For this reason *best practices* and *guidelines* used in practice have a *low level* of formalization with respect to the above classification.

Models of team behavior and *qualitative studies* (e.g. some case studies) reside between the two extremes of the detailed formalization of algorithmic approaches and the best practices adopted by HR experts in companies. They provide general guidelines which however are sometimes difficult to operationalize and to formalize. Hence, they cannot inherently be used in algorithms. A further step towards a complete formalization and operationalization of the therein contained concepts is required in order to use the knowledge produced in these contributions for algorithmic team composition. Nevertheless, studies belonging to this formalization category provide information on what variables influence team output/outcomes. Moreover, they also describe which output/outcomes these variables influence. For this reason they are regarded as approaches with a “*Medium Level*” of formalization degree.

When these strategies have to be applied for algorithmic recommendation the effort for the application may vary, depending on the formalization degree. Team composition models with a high level of formalization can be integrated very easily in a team recommender, since all variables and relationships are operationalized. In contrast, best practices that are not formalized and operationalized require additional steps for this type of usage.

3. Team Composition

4. A Meta Model for Team Recommender

The previous chapter has shown the variety of team composition approaches existing in literature and practice for composing effective teams. At the same time, a more holistic understanding of the team composition problem has been revealed: Team composition is not only affected by a team's properties but is also dependent upon a number of aspects such as organizational constraints or a team's environment. The problem has been identified as a very complex and manifold one.

The objective of this chapter is to describe a concept for mapping the team composition problem to a system that considers the previous findings elaborated from related work. First, this chapter introduces algorithmic team composition with its general advantages and drawbacks. Then, related work is analyzed and classified according to so-called team composition aspect entities. Thereby, relevant components for a team recommendation system are identified.

Given the results of this analysis a concept for team recommendation is then proposed: a meta model to construct team composition strategies. A discussion of the advantages and disadvantages of the meta model solution compared to a specialized, tailored solution reveals when and why this approach is appropriate.

In the following step, the previously created categorization which serves as a foundation of the meta model approach, is explained and formalized. By doing so, the rather abstract aspects are mapped to meta model components that can be used in team recommendation support systems. These components are used to encode the knowledge of team composition models. Finally, three example meta model instantiations of team composition models found in literature are described in order to elucidate the meta model's functionality.

4.1. Advantages of Team Recommendation Systems

In order to support the usefulness of algorithmic team recommendation in the context of open innovation (short OI), a number of arguments can be asserted. First of all, OI networks can become very large [Fagerberg et al., 2006], because the set of candidate-actors for OI interaction and collaboration would increase dramatically when the restriction to company boundaries does not apply. Thus, in the case of OI, actors responsible for assembling a team normally would not know all other actors from the network personally and could benefit from an algorithmic solution.

In addition, having a system that proposes recommendations may also lower the barriers of initiating collaborations. This is attributed to the fact that by receiving recommendations for team members, project initiators can examine the possibility of whether there are appropriate teams for a task. Revealing their project idea to people from other companies in this case is not necessary.

One part of the interviews in Section 3.2.1 is dealing with the “challenges of team composition in open innovation”. Therein, experts from the team composition domain were asked about their opinion of whether an IT support mechanism for team recommendation in large innovator networks (such as open innovation networks) makes sense. Most of those experts stated, that composing a team is a very time consuming and challenging task especially when the number of possible team members is large: “*If I had, for example, to inspect 500 CVs . . . you can’t be fast in doing that*” [s8]. Another expert regarding the same issue said “*If a customer requires a team with members providing skills in, let’s say, five specific products. Finding these five people in a company that is becoming larger and larger is very challenging.*” [s9].

Some interviewees had very specific ideas about what such an IT support should entail based on the difficulties they had faced in the past: “*To assemble a team in a large innovator network such as the [name omitted by the author] I would like to have a system that tells me what I have and what I need in total, what the things are that are important to me and then produces recommendations with explanations as to why it recommended these teams.*” [s10].

Aside from the specific case of open innovation, there are a number of general arguments for the usefulness of algorithmic team recommendation. First of all, using IT support to assemble teams can avoid bias and prejudices towards specific persons guaranteeing that subjectivity does not prevent the composition of effective teams.

In this regard, past experiences by project managers can be more systematically and

objectively taken into consideration and can also be shared among project managers using the recommender system.

Often enough a multitude of variables and aspects that cannot all be considered accurately makes it impossible for a project manager to assemble a good team. In this case an IT support system that helps to extract important information about teams in a short time (e. g. how heterogeneous a team is) is of great value.

A final argument in favor of team recommendation is the fact that it facilitates leaving “well-trodden paths” in team composition: Sometimes new team composition strategies are not adopted in a company even though they could be beneficial. This can be due to different reasons such as lack of knowledge, the extra amount of effort required to develop and apply new strategies, etc. In this case a team recommender can help to bring in new team composition knowledge and insights.

4.2. Challenges of Algorithmic Recommendation of Teams

In comparison to the classical recommendation items such as POIs (points of interest), products, information items, etc., teams are far more complex to recommend. This can be attributed to two principal facts: (i) the dynamics of the items to be recommended and (ii) not just a single item, but actually a *set of items* has to be recommended. Apart from that, it is also challenging to find a way to describe and characterize the items to be recommended (the teams).

Team recommendation is dynamic in terms of characteristics that matter for the recommendation. In contrast to the recommendation of other items such as products, information items or POIs the attributes that matter for the recommendation vary very rapidly. For example, for certain recommendation tasks a team may have to fit a required skill set only. Under other circumstances the team members should also have an ideal combination of certain character types (e. g. Section 3.3.4). Moreover, the attributes of the items may also change rapidly. For instance, team members extend their skill set during their career. This problem plays a role in person recommendation systems such as expert finders (Section 2.1.4).

Recommending a set of such dynamic items results in an even more complex problem since the interdependencies between the individual items (e. g. team members) are not known. The compositional recommender in Section 2.2.2.5 is one example for this type of recommendation systems. Identifying exactly these interdependencies between items (team members) is one of the main goals and challenges of a team recommendation system.

4. A Meta Model for Team Recommender

A major challenge of solving the general problem of algorithmic team composition (or algorithmic team recommendation respectively) lies in the **identification and classification of all key components** (e.g. involved variables, dependencies, etc.). This is necessary because disregarding important components leads to imprecise or overfitted recommendations: for instance, a recommender system may be developed that only considers currently available skills. Such a recommender system can produce good results for a specific task and environment, but does not solve the problem on a more abstract, general level and can thus not be used to compose teams in other situations.

Another critical aspect concerning the algorithmic recommendation of teams is **the criterion used to rank teams**. While in recommender systems the expected, personalized utility of an item is usually represented by e.g. ratings, this may not apply to teams. Work on team composition (Chapter 3) has shown that there are several other criteria used for that purpose which comprise measures such as team performance, communication costs or turnover. Therefore, an appropriate mechanism for considering these findings in a recommendation system has to be investigated.

Unlike news or movie recommendation, team recommendation is not a very time-critical task. That means, it is not necessary to produce recommendations very fast or in real-time. Nevertheless, because of the potential high number of teams that can be generated in an open innovation network, **mechanisms for dealing with the complexity of the team recommendation process** are useful.

Many recommendation engines provide the possibility to learn from past experiences or choices. For instance, the results of collaborative filtering improve along with increasing number of items (e.g. books) a user has rated. In domains such as the movie and book recommendation examples, it is very likely that the user-item matrix needed in collaborative filtering is filled with data within a reasonable time. This is rarely the case for team recommendation. Only very few records of performed tasks would be included in the dataset. When the team performing environment, i. e. the recommendation context, has to be taken into account as well, the rating matrix becomes even more sparse, which is a problem that affects context-aware recommender systems in general [Adomavicius and Tuzhilin, 2011]. The cold start problem (cf. Section 2.2.1) heavily affects this type of approaches. This **sparseness of feedback** is a further challenge a team recommendation system has to face.

Section 3.4 reasoned that team composition approaches (and hence also algorithmic approaches) are based on underlying team composition models. It is of high importance to choose an appropriate model that fits the needed use case. The challenge resides in **finding which model(s) should be used** for this purpose.

Furthermore, investigating appropriate **interaction paradigms** between the user in charge of assembly the team and team recommender system is relevant. A traditional collaborative filter for instance, does not require any kind of particular query and automatically generates recommendations for a given target user. Context-aware, proactive recommender systems use user profiles and context information as query to generate recommendations. Depending on the concept developed for team recommendation, an appropriate interaction paradigm which accounts for the characteristics of the team composition problem has to be chosen or be developed.

Last but not least, it is hard to **determine a suitable problem class** for the problem of algorithmic team recommendation as such. The problem has characteristics of database problems, constraint solving approaches, case-based reasoning, (content-based) recommender systems, information retrieval, and statistics.

Summarizing, the seven main challenges of team recommendation are:

- 1) **Identification and classification of all key components of team composition/recommendation**
- 2) **Finding an appropriate mechanism to evaluate and compare teams with one another.**
- 3) **Keeping the computation time within reasonable limits.**
- 4) **Regarding the sparsity of data and the problems associated with it.**
- 5) **Identifying the role of team composition models.**
- 6) **Choosing or developing appropriate user interaction paradigms for the team recommendations scenario.**
- 7) **Investigating which characteristics of the team recommendation problem are related to which problem class.**

In the following, whenever design decisions or facts address one of these challenges, the relevant part in the text is labeled according to the corresponding challenge (e.g. Challenge 3, Challenge 4).

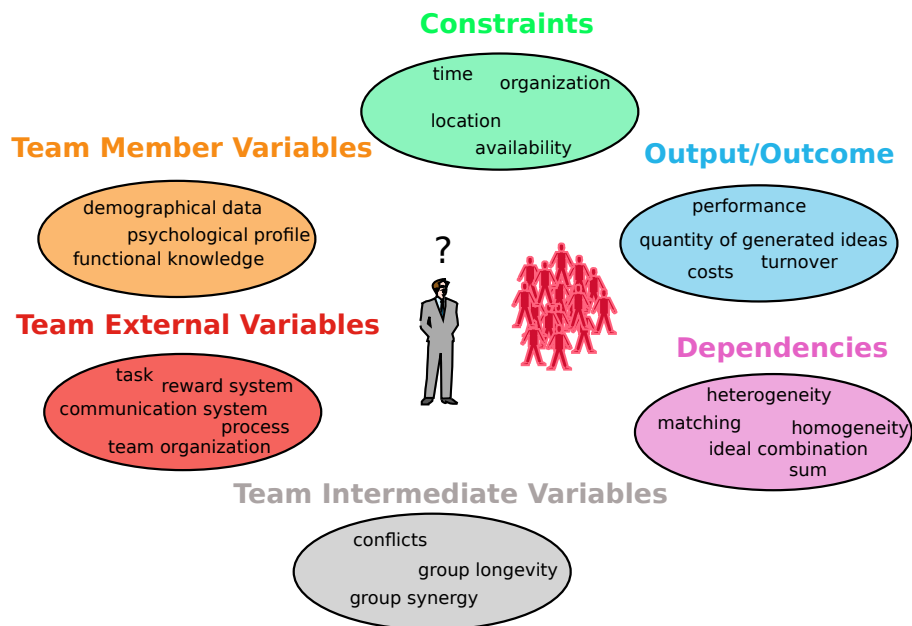


Figure 4.1.: Classification of team composition aspects

4.3. Aspect Entities of Team Composition

Collecting and classifying aspect entities for the team composition problem from existing work is the first step towards identifying which aspects have to be considered in a team recommendation system. Classes of aspect entities that have been elicited in the domains presented in Chapter 3 can be designated as team member variables, entities formalizing output/outcomes, team intermediate variables, team external variables, dependencies, and constraints [Brocco and Groh, 2009]. Figure 4.1 illustrates the identified classes and some examples for contained aspects.

The following sections provide an overview of these identified aspect entities. Where applicable, some examples for team composition approaches that include the respective aspects are quoted.

4.3.1. Team Member Variables

The class of team member variables (abbreviated as TMVs) encompasses all variables strictly related to individual team members. That means, they can be established directly from the properties of an individual team member. Examples for team member variables are skills, demographic variables, egocentric social network and psychological variables.

Functional Knowledge: Individuals can be characterized according to their functional knowledge and expertise (e. g. skills). This property can be used to ascertain whether a team member is appropriate for a specific task in terms of required skill set. This kind of matching is often used in approaches applied in practice as elucidated in Section 3.2.

Demographic Variables: Potential team members each have individual profile properties that describe them in terms of age, gender, nationality, height, language etc. Although this is very simple data, it has great impact on team output/outcome. For example: cultural background (which can be modeled in demographic variables) is useful to avoid conflicts within the team, that can sharply reduce team performance (e. g. [Bouncken and Winkler, 2008, Pelled, 1996]).

Social Network Variables: The social network between team members is important for the team composition/recommendation problem in various ways. Some researchers claim that people with strong relations with each other (i. e. social network ties) tend to work inefficiently together because of information redundancy. Therefore a team with initially weak ties may perform better (e. g. [Granovetter, 1973]). Other researchers have the opposite opinion (e. g. [Krackhardt, 1992]). In order to estimate these contradictory findings it is important to model precisely the nature of ties in these networks. It appears evident that social networks have an influence on team work while the specific kind of influence probably depends on the concrete problem. Kratzer's work presented in Section 3.1.1 for example, shows that social network ties such as the number of friendship and friendly relations can affect team performance. The interviews presented in Section 3.2 also stress the importance of the social network for team composition.

Psychological Variables: A range of variables characterize the psychological constitution or mental state of potential team members. Trower and Moore analyzed the impact of motivation on the result of group processes [Trower and Moore, 1996]. In another study Lyman and Richter ([Lyman and Richter, 1995]) suggest personality combinations for effective teams, based on the Myers-Briggs test (cp. Section 3.3.4). White [White, 1984] and Chen and Lin [Chen and Lin, 2004] presented two other approaches related to that theory.

4.3.2. Output/Outcome Variables

The central goal of team recommendation is to recommend teams with respect to maximum (predicted) performance. Performance may correspond to a high quality of proposed solutions or quantity of ideas generated in a brainstorming session. Other output variables may be the general satisfaction among team members [De Dreu and Weingart, 2003] or

4. A Meta Model for Team Recommender

the time needed to develop a solution.

It is helpful to distinguish between nominal performance and social performance of a team, although both can correlate. Nominal performance addresses the objective output of the team (mostly external criteria) while social performance is subjective from the perspective of team members and addresses interpersonal aspects (internal criteria).

Nominal Performance: A wealth of nominal performance indicators is described in Chapter 3 which includes the indicators net revenue created by the team, the time it took the team to develop the solution, customers' satisfaction or team leader's evaluations.

Social Performance: Teamwork does not only produce nominal outcomes such as customer satisfaction or other performance indicators. Teamwork also touches on social aspects such as satisfaction or trust. These output/outcome variables are social performance variables. Other examples for social performance variables are acceptance and agreement. They were investigated by Kahai and Cooper [Kahai and Cooper, 1999] in a study on the influence of the communication environment. To be more precise, the level of agreement and acceptance within a team was measured as output/outcome variable depending on the communication environment. The same could be done with other variables such as the average satisfaction of team members.

4.3.3. Team Intermediate Variables

In many cases socio-/psychological models also include intermediate variables. Pelled's approach presented in Section 3.1 for example, claims that heterogeneous teams (heterogeneous with respect to job-related demographic background) have a better performance because they produce "substantive conflicts". The number of such substantive conflicts may be considered as a team intermediate variable (short TIV). Team intermediate variables are dependent on more fundamental "prior" variables and in addition have influence on the output/outcome of a team. Listed as below are some examples for team intermediate variables:

Conflicts: Nature, number or characteristics of arising conflicts in a team are typical team intermediate variables. They often depend on other more fundamental variables such as psychological team member variables. In [Pelled, 1996] substantive conflicts (which are said to increase task performance) and affective conflicts (which increase turnover) are discussed.

Group Longevity: Group longevity (also called group tenure) measures how often and how long a specific team performed in the past. This may have influence on the

amplitude of certain effects ([Schippers et al., 2003],[Pelled, 1996]) such as conflicts, and hence on outputs.

Group Synergy: Synergy can be defined as “*the interaction of two or more agents or forces so that their combined effort is far greater than the sum of their individual efforts*” [Mears and Voehl, 1994] although other definitions also include the notion of positive and negative synergy (e. g. [Hackman, 1987]). Negative synergy defines negative effects emerging from interaction that make the collaboration less effective than the separate work of individuals. Positive synergy allows teams to outperform merely nominal teams (which are just groups of team members working alone and unifying their work at the very end) and thus may have substantial impact on team performance [Hackman, 1987]. It can be considered as intermediate variable because constellations of other variables control whether synergy in a team exists or can develop respectively. For, instance the interaction patterns or the functional knowledge of team members are variables that induce group synergy.

4.3.4. Team External Variables

Team external variables (abbreviated as TEVs) define the environment and the context of a team’s problem solving process. In other words, this category consists of variables that directly affect the output/outcome of a team from the outside. In contrast to team member variables these variables cannot be derived from the team members’ properties directly but have to be acquired from the *team’s context/environment*. This consists of dimensions such as the problem-solving process type, task type, task content, urgency, team leading policies, communication system, team climate, team structure, material resources available, whether the team is distributed or co-located, or if the organization to which the team belongs, has appropriate reward and/or feedback systems ([Powell et al., 2004],[Hackman, 1987]).

Team external variables are usually complicated to measure, model and operationalize. The team recommendation domain includes a very high number of external variables. As a matter of fact, most socio-/psychological works on team design discuss and emphasize the importance of team external variables in both, be it in general (e. g. [Hackman, 1987]) or in detail (i. e. on single variables, e. g. [De Dreu and Weingart, 2003]). The acquisition, operationalization and comparison of team external variables is probably one of the main difficulties in solving team recommendation algorithmically.

Task-related TEVs: It is obvious that a team performs differently depending on the task it has to solve. For example, when the task does not fit the team competencies the performance of the team will probably be low.

4. A Meta Model for Team Recommender

However, the problem class or task nature is also a task-related team external variable that has a substantial influence on the team's composition. Kozlowski and Ilgen [Kozlowski and Ilgen, 2006] classify task types as “team activities” including *doing things*, *making decisions* and *creating, adapting and inventing* solutions. Not included by this classification is the activity *research*.

Another important variable that influences the output of a team is the task structure. Depending on whether a task can be structured well (parallelizable, partitionable, etc.) or not, teams perform differently [White, 1984]. Other aspects influencing the team process and its output/outcome are *urgency* and *importance*.

The team composition approaches used in practice (Section 3.2) have shown that team composition is heavily dependent on the task description. Moreover, not only the set of required skills varies from one task to another. Also the optimal personality composition or the optimal role distribution can be different within the team.

Organization & Environment: The organization in which a team is embedded has a strong impact on the team: If an organization is not used to radical innovations, for example, people may (involuntarily or voluntarily) reduce their own ideation space because their organization tends to prefer less risky and less creative ideas.

Mello and Ruckes' work introduced in Section 3.1.3 shows that in environments with high uncertainty specific types of teams perform better than others. Thus, the same team could be evaluated in different ways depending on the organization. The tradeoff between risk and creativity is only one of many aspects related to organizational properties. Other influence factors include whether the organizations are nationally or internationally active or how formal/informal the working atmosphere in the organization is.

Team Organization: When starting a team project, the structure and organization of the team must be defined, for example, the type of the team (e.g. real or nominal team [Steiner, 1972]) or the type of team management style (manager-led, self-managing or self-designing) [Hackman, 1987].

Nowadays, time and space constraints are overcome by setting up virtual teams which are influenced differently by TMVs compared to co-located teams.

Reward System: A reward system is crucial for motivating team members to accomplish their task successfully or to participate actively. Its presence alone makes a difference, but its implementation is crucial. Some reward systems are based on soft rewards (e.g. reputation), some base on explicit rewards (e.g. money) [Hall and Graham, 2004]. Hybrid solutions are also possible.

Information System: The information system that an organization provides may have substantial impact on team performance. Depending on the quality of the information system, a team is more or less successful in finding the information necessary to accomplish its task. Moreover an adequate information system supports the team in choosing the best strategies for their work.

Additionally, an information system should provide feedback on the current quality of the ongoing team activities [Hackman, 1987]. Teams which have such feedback systems can correct their problem-solving strategy and direction if necessary and hence increase the quality of their results (e.g. [Hurley, 1998]). Teams without such type of system depend on themselves and on their own valuation.

Process: This category of variables contains representations of critical components in a team problem-solving process - mainly management-related issues. It addresses questions such as: What technique is used to solve conflicts among team members? Are group norms set appropriately? Do members know what the purpose of the team is? Furthermore, this category contains general information about process structure and the problem-solving approach [Scott, 2007]. Creative processes, for example, can be instantiated as one of several different creativity techniques used for creative problem solving [Forster and Brocco, 2008]. Also for decision-making processes a number of techniques has been described in literature (e.g. [Saaty, 2008],[Rigopoulos et al., 2008]), but the decision-making processes are often governed by best practices of the organization. Examples for different processes formalizations can also be found in software development (denoted as software development methodologies) [Ramsin and Paige, 2008].

Communication System: The communication system has to be chosen depending on factors such as the type of team desired (e.g. virtual or co-located team), the task and other factors. Virtual teams, for instance, mainly use an IT infrastructure to communicate similar to open innovation projects such as the Free/Open Source software projects. This includes tools for synchronous communication like chat or video conferences as well as asynchronous communication like email or forums [Kahai and Cooper, 1999]. Depending on the requirements of the process and the portfolio of tools provided by the organization, the communication system may vary and influence the outcomes [Straus and McGrath, 1994]. The communication system can be categorized, for example, regarding synchronicity and media richness.

4.3.5. Dependencies

The most important factor in modeling the support for team composition is to know the dependencies between the team member variables and output/outcome. In sociology and psychology these relations are usually defined and described very coarsely. Two scopes in the use of dependency variables have been identified. First, they describe what specific property of a set of team member variables is investigated (whether e.g. it is variety, the sum or a specific combination of specific values for TMVs). Second, they indicate whether the dependency of this property is positively or negatively related to the specific output/outcome discussed. In the following, the categories related to the first scope are described. The second scope usually included in studies on team composition is expressed in terms of positive or negative correlations (although more complex relationships are also imaginable) and is therefore not further elucidated in the following paragraphs.

The team composition approaches presented in Chapter 3 include a variety of dependencies that were applied to compute e.g. the perfect combination of specific attributes such as skills or personalities. Other dependencies indicated, for example, that cohesiveness of friendship ties has a positive linear relationship with team performance ([Kratzer et al., 2005]). Many types of dependencies can be used to express certain properties of teams for team composition, some of which are listed in the following.

Homogeneity/Heterogeneity: The concepts of heterogeneity (or diversity) and homogeneity are frequently used in combination with the topics “team composition” and “team design”. As previously mentioned, Pelled, for example, claimed that job-related demographic diversity (heterogeneity) leads to higher performance while visible demographic diversity leads to lower performance and lower turnover. This works follows the cognitive resource diversity theory from the field of management [Cox and Blake, 1991].

In contrast, the similarity attraction theory (e.g. [Byrne, 1997]) has shown that homogeneous teams “ [...] *are likely to be more productive than heterogeneous teams because of mutual attraction of team members with similar characteristics.*” [Horwitz, 2005]. In some cases these ambiguous results regarding heterogeneity and homogeneity can be explained by the use of the terms “heterogeneity”, “diversity” and “homogeneity” itself. Harrison and Klein presented in their work [Harrison and Klein, 2007] three different categories of “diversity” or heterogeneity and showed examples where diversity was used with different meanings resulting in different outcomes. They identified three types of diversity with respect to the problem of team composition:

- 1) *Separation*: team members differ from one another in their position along a single continuous attribute (e.g. endorsement about a specific type of problem-solving ap-

proach to use [Scott, 2007]). Separation emphasizes distances between team member variable values → applicable to interval scaled variables.

- 2) *Variety*: how many different classes of instantiations of a specific team member variable are present in a team (e. g. number of different nationalities in a team). Variety emphasizes the number of different team member variable values → applicable to nominal scaled variables.
- 3) *Disparity*: how the resources and decision power are distributed among the team members (e. g. one member has all the resources vs. many members have them). Disparity emphasizes ratios of different team member variable values → applicable to ratio scaled variables.

These metrics represent different diversity levels within a team, i. e. if separation, variety and/or disparity are high or low within a team. Which heterogeneity/diversity type shall be applied to which variable is mainly dependent on the team composition model and its assumptions.

For example, according to Harrison and Klein variety of tenure can be seen as each of the above-described types, depending on the desired model statement: Diversity of tenure can be seen as separation by modeling tenure on a tenure continuum. This separation is then operationalizable in a hypothesis such as “the less separated team members are on this continuum, the more similar their attitudes, beliefs and values are and the less conflicts result”.

Moreover, diversity in tenure can be also seen as variety by partitioning the representation of tenure in equivalence classes. The equivalence classes may then be associated with different classes of experiences, school of thought, etc. which, in turn, may lead to more effective decisions, more creative products etc.

Finally, diversity in tenure can also be seen as disparity by assuming a fixed total tenure for the team which is the sum of all tenure values of the individual team members. In case the team members have a high difference in tenure this can have various consequences with respect to team intermediate variables such as the type of conflicts or the impact on the decision.

Matching: A matching function is often used to match specific properties of team members to a given specification. A typical example is the matching of desired and actual skills for a given task.

Ideal Combination: In psychology, specific *combinations* of variable values are some-

times proposed as desirable. The study of Lyman and Richter [Lyman and Richter, 1995] on optimal personality types, mentioned in Section 4.3.1, is a typical example, because it describes what kind of “character types” should be present in a team and match especially well. In addition, other variable types sometimes need optimal combinations such as competence, nationality, etc. Ideal combination can also be viewed as a special type of matching.

Sum: In contrast to ideal combination which can be viewed as typically operating on nominal scaled variables only, interval scaled numerical team member variables can also be aggregated in several ways in order to be minimized or maximized in combination. One example could be the sum of these numerical team member variable values. Depending on the requirement, these sums can be used to rank teams. For example, if the tenure within the team should be high then the sum of all tenure of team members can be used to evaluate if this desired property applies.

4.3.6. Constraints

Constraints represents relations and restrictions that have to be satisfied by the team. Hence, this class does not directly represent variables such as the classes TMVs, TEVs or output/outcomes, but represents restrictions on TMVs.

Team composition is heavily reliant on constraints such as desired versus actual location of team members, workload, availability, budget etc. Instead of explicitly incorporating constraints into the theory of team recommendation, constraints are conceptually mainly understood as filters to rule out teams that are incompatible with desired specifications. If no team that fulfills specified constraints exists, these constraints can be relaxed.

Unlike team external variables, constraints do not affect the team’s output/outcomes (i. e. the recommendation system will predict the same output for a team independent of the constraints), but they will certainly affect the possible result set of the recommendation (i. e. it may be that certain teams will not be considered, because they do not satisfy the formulated constraints).

Time & Location: Constraints on time are very typical constraints for the team composition problem. In the above-mentioned sense team external variables such as the available time for a project have to be separated from the constraint that specifies absolute time intervals in which a person has to be available. This is the case because the first aspect (available time) is a TEV, since it affects how a team works (e. g. a team has two months

to complete a task) while the availability of team members is a TMV and can be restricted by constraints to be available within a specified time frame (e. g. from June to August).

Spatial constraints are usually applied for the composition of co-located teams. Virtual teams typically are not subject to these constraints as teams that work in a co-located manner. When it is required that the team members are located within a predefined spatial range, spatial constraints can be employed (e. g. [Brocco and Woerndl, 2011]).

Organization: For OI networks consisting of companies and/or organizations, sometimes also bureaucratic or legal matters play a role. For example, exclusively freelancers or certain specific organizations can be considered for team recommendation because of strategical matters or the existing partnerships. In this case constraints on partner organizations can be applied.

Availability: The availability of team members for a project is obviously a central constraint for assigning a person to a team. The conducted interviews (see Section 3.2.1) showed that this criterion is decisive for team composition in practice. This constraint specifies whether the team member has the *required resources* in terms of time for performing the task in a given time frame (which in turn can be expressed with the above-described time constraint).

4.4. Categorization Analysis

All above-mentioned aspect entities were either extracted from related work or elicited during the interviews with practitioners described in Section 3.2.1. The strategies for assembling teams derived from these works were denoted as team composition models in Section 3.4. As can be seen, each team composition model derived from related work or best practices considers only a specific subset of these aspect entities that is relevant to the respective theory. The subsets used in these team composition models are however not disjoint. For instance, Bouncken and Winkler [Bouncken and Winkler, 2008] and Pelled [Pelled, 1996] both included demographic variables (as team member variables) in their team composition model. The classification of aspect entities introduced in Section 4.3 shows that the various team composition models base their composition strategy on similar concepts, namely team member variables, team external variables, output/outcomes, constraints, team intermediate variables and dependencies. Not all of them are necessary in every team composition model. Each model uses only a selection of these aspect entities, which are summarized in the following for a better overview:

Team member variables describe properties of individual team members that range

4. A Meta Model for Team Recommender

from psychological to demographic variables. Also egocentric social networks belong to this variable class.

Team external variables represent all aspects that influence the team from the outside and cannot be acquired from the team members. For example, the task description and urgency, the communication media and the reward system applied are typical team external variables.

In most team composition models the effects of team composition and the environment are measured with respect to their influence on *output/outcomes* of a team. Popular output criteria are team performance (implemented e. g. as customer satisfaction) or social effects on a team such as resulting trust.

Dependencies relate team member variables to output/outcomes. They first aggregate team member variables to numerical values and subsequently map them to output/outcomes variables. An example for dependencies is the relation between the heterogeneity of competencies (which is a TMV) and the creativity of ideas (which is an output/outcome variable).

Certain team composition strategies also include *team intermediate variables* that mediate between team member variables and output/outcomes. As has been shown, such variables comprise, for example, group longevity or conflicts that arise because of the personality (which is considered as TMV) of specific team members.

Finally, in most of the approaches used in practice, *constraints* have to be applied. This aspect entity represents hard constraints acting to reduce the number of team member candidates. Popular constraints are the availability of team members within specific time frames or the limitation of partner organizations.

Apart from this rather structural categorization of aspect entities, a more semantic categorization can be identified among the variables (since constraints and dependencies are not variables they are not considered for this further categorization). For instance, in different meta model components aspects related to the psychology or skill domain can be found. The semantic class “skills/functional knowledge”, as example for this semantic categorization, can be used in several aspect entities depending on the team composition model: it can be used to describe actual skills of a team member (in that case “actual skills” is a TMV) or the skills acquired during the execution of the task (hence it represents an output/outcome).

As a consequence, two categorizations need to be distinguished: a vertical (structural) categorization and a horizontal (semantic) categorization. Both are involved in the aspect

categorization process, i.e. the process of identifying to which aspect entity variables described in team composition model belong. The vertical categorization for variables consists of the corresponding above-listed aspect entities, namely: TMV, TEV, TIV and output/outcomes (again: dependencies and constraints are not variables). The horizontal categorization consists of semantic domains of the variables such as skills/functional knowledge, psychological variables (e.g. self-esteem), social aspects (e.g. trust, friendship relations) or spatial aspects. In contrast to the vertical categorization, the horizontal categorization is not limited to the above-mentioned categories, but contains an unlimited number of classes representing different semantic domains.

Thus, when a team composition model contains “skills” as influence factor, this denotes only the horizontal class (in this example “skill/functional knowledge”) but does not always directly imply the vertical structural class it belongs to. This has to be identified by means of the properties and description of the aspect entities provided by the vertical, structural categorization, which specifies whether it means the TMV “actual skills” or the output/outcome variable “acquired skills”, when talking about “skills”.

For a better understanding, Figure 4.2 depicts this two-dimensional categorization with two examples for horizontal classes: skills/functional knowledge and trust.

	Team Member Variables	Output / Outcome Variables	Team Intermediate Variables	Team External Variables
Skills / Functional Knowledge	Actual Skills	Acquired Skills		Skills for Task
Trust	Actual Trust	Acquired Trust	Induced Trust	
...				

Figure 4.2.: Vertical vs. horizontal aspects of team composition

The rows in this matrix correspond to the semantic classification of team composition aspects found in existing team composition models while the columns correspond to the structural categorization.

Skills (a horizontal category) in the context of the vertical categorization TMV represents the *actual skills* of a team member. However, skills in the context of output/outcomes

represent the *acquired skills*. In terms of TEV in the semantic domain skills (horizontal categorization) the corresponding variable could be a description of the task by means of skills that describe a task (in Figure 4.2 labeled as “Skills for Task”). For the example of trust, apart from *acquired trust* and *actual trust* that are analogous to the skill example, trust can be considered as a team intermediate variable if it represents *induced trust*, i. e. trust caused by prior team member variables and that influences other output/outcomes variables. Hence, when a team composition model includes an aspect such as skill or trust, it should be determined to which structural class it actually belongs in order to distinguish and classify it clearly.

To summarize, the analysis of related work and best practices shows that although various team composition strategies exist, many of them share the same aspect entities. For instance, most of the approaches cited make use of TMVs to predict several types of output/outcomes variables. Although these approaches differ from a semantic perspective, from a structural perspective they exhibit similarities. This property can be exploited for a team recommendation system.

When classifying influence factors into aspect entities great care needs to be taken: Most team composition models found in the literature and/or practice merely name the semantic class of variables. This can lead to wrong assignments. For instance, skills can be considered as TMV or output/outcome variable or even as TEV. Thus, when assigning skills to an aspect entity it has to be carefully assessed what notion of skills is to be considered. The matrix depicted in Figure 4.2 with the aspect entities (vertical classification) and the semantic categories (horizontal classification) can be used as a conceptual tool to more clearly identify which aspect entity an aspect of a team composition model belongs to.

4.5. Solution Approach

Following the analysis of aspects influencing team composition, is the decision on an adequate approach for team recommendation. Both have to be considered: to identify the role of these aspects and how to integrate them into an algorithmic solution. Moreover, the concept for this algorithmic solution has to be elaborated.

The aspect entities found in the analysis of related work show that many team composition models rely on the same structural components. Hence, one possibility to build a team recommendation system is to combine these components (i. e. the aspect entities) depending on the team composition strategy the system uses. Figure 4.3 depicts the idea of this solution: given a team composition model description, aspect entities are utilized

to construct a representation of it. This representation is then injected as knowledge in a recommender algorithm to recommend teams.

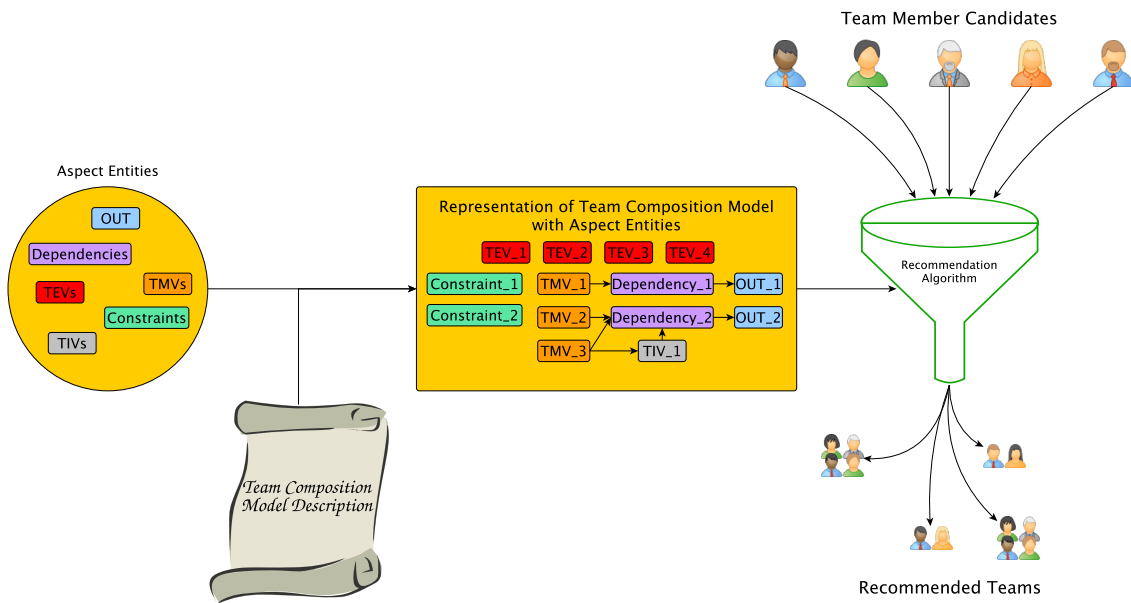


Figure 4.3.: Schematic illustration of a generic solution for a team recommender system

However, this solution requires that each aspect entity has to be modeled on a higher abstraction level: Each aspect entity is used to represent various different variables or relations. For example, the aspect entity “team member variable” should be utilizable to model “age”, “social network”, “competencies”, etc. in a team recommendation system. If the abstraction level of TMVs for example would be very low (e.g. team member variables would be modeled as data structure representing edges and nodes to perfectly model a social network), this would make it impossible to use it to represent other concepts (e.g. “competencies” or “age”). In order to address this requirement a compromise has to be done: Some particularity of the specific subvariables (e.g. “competencies”, “age” etc.) has to be disregarded in the model of TMVs. This is done to allow a generic representation and a standardized algorithmic use of them. However, this design decision may result in problems in terms of recommendation accuracy because potentially important factors for the recommendation may be excluded during the modeling.

Another possible solution similar to the approaches in Section 3.3, is to create a team recommender for a specific use case. As there is no need to compromise for the modeling of team composition aspects, this solution may be appropriate, even though it can only produce reasonable recommendations for exactly the use case it was developed for.

At this point a dilemma has to be faced since two solutions for addressing the team recommendation are possible: (i) Is it better to restrict the possible use of the recommender

system to a single use case and strategy (e.g. specific environment and/or specific task) but potentially have more accurate recommendations, or (ii) is it better to implement a generic system for team recommendation capable of using several strategies but potentially loose in accuracy? There are pro and cons for both approaches, which are discussed below.

4.5.1. Generic Solution vs. Tailored Solution

The generic solution's goal is to allow for the instantiation of several team composition models including their context and constraints. This can only be achieved by a precise modeling of influence aspects that have to be used as components to construct team composition models. As these components have to be standardized, regardless which specific composition approach is instantiated, the same recommendation algorithm can be used. Moreover, for a specific team composition approach, several different algorithms can be used to generate recommendations.

On the other hand, tailored solutions such as Colucci's [Colucci et al., 2005] or Malinowski's [Malinowski et al., 2005] works also have many advantages. First of all, they are tailored for a specific domain and concept. That means, the level of abstraction is lower than of a generic approach as this necessarily has to employ a higher level of abstraction for it to cover many team recommendation scenarios. A high abstraction level may, however, result in imprecise recommendations compared to tailored, specific solutions.

Implementing and applying several tailored recommender approaches is however time consuming and complicated in domains where a high variety of different team composition strategies is needed. Additionally, training a recommender based on a generic system can be more efficient because the underlying data structures used for all the different approaches are the same. This means that data of teams that accomplished a task in a specific scenario can be used almost independently of the actual approach used to train the recommender. In the tailored recommenders no common structure of aspect entities is usually defined. Therefore the conversion and adaption of input and output data may be complex.

Finally, extensibility with respect to team composition strategies can be achieved more easily through a generic solution since new composition approaches can be "assembled" using the provided generic components.

Table 4.1 summarizes the advantages of both approaches: generic and tailored. As can be seen, the generic solution has more advantages and in addition directly addresses some of the challenges of team recommendation listed in Section 4.2.

Generic Solution	Tailored Solution
Different strategies can be instantiated and used in the same algorithm	Tailored for a specific context
Different algorithms for same strategy usable	Higher accuracy of recommendations
Comfortable change of strategy in recommendation engine	Simpler implementation
Easier “ re-use ” of training data	
Easier comparability and extension of strategies	

Table 4.1.: Advantages of possible team recommendation approaches

For example, the fact that in related work and best practices findings on strategies to compose effective teams have already been generated should be taken advantage of. For certain configurations of team external variables, these findings should be applied when recommending teams as integrating this knowledge (such as in knowledge-based recommender systems, Section 2.2.1.3) into a recommender mitigates the cold start problem (**Challenge 4**). Thus, existing strategies already contain knowledge and do not require necessarily historical teamwork data. A generic recommender system employing these strategies can potentially generate good recommendations without input of past team activities.

In brief, the generic solution approach is capable of using produced knowledge, i. e. using various team composition strategies already developed in related work (**Challenge 5**). In contrast a tailored solution can only employ one team composition strategy, disregarding most of the previously created knowledge in this domain.

For all these reasons it makes sense to implement a generic team recommendation system rather than a tailored one in order to achieve flexibility in terms of application possibilities. Therefore, in the following sections a generic approach for team recommendation is presented, that is called meta model for team recommendation and serves as framework for mapping team composition strategies into recommender systems.

4.5.2. A Meta Model for Team Recommendation

To guarantee flexibility in terms of team composition models an approach has to be found that allows the usage of numerous team composition models in the same (generic) recommender system.

For this purpose a **meta model for team recommendation** is proposed to model team composition strategies needed for team recommendation. As every meta model it models properties of models (in this case team composition models) rather than real world artifacts, and can hence be seen as a language to create a team composition models for recommending teams.

In order to develop this meta model the problem domain of team composition has to be known, for which Chapter 3 and Section 4.3 provide a solid foundation. In addition, proper formalization and operationalization for aspect entities have to be designed so that they can be integrated into recommendation engines. The meta model has to be flexible enough to instantiate various team composition models without losing the required expressive power. Simultaneously, it has to be abstract enough for several recommendation algorithms to work on its components.

The classification of aspect entities in Section 4.3 provides a good starting point for the development of this generic approach. The following subsections build and formalize the necessary meta model constituents step by step.

4.6. From Team Composition Aspect Entities to Meta Model Components

The team composition aspect entities introduced in Section 4.3 and analyzed in Section 4.4 can be used as components for the meta model framework. To be more precise, the structural, vertical categorization (represented by the aspect entities) can be used for this purpose.

The horizontal categorization has its origin in the application domain, namely the team composition domain predominantly represented by studies and best practices on team composition. This classification is used to categorize the semantic domain of a variable. When a description of a team composition model contains a concept such as “skills” or “satisfaction”, from these labels only the horizontal category can be directly inferred. It is given by the label itself (in this example hence “skills” and “satisfaction”) . However, it is

not directly clear, to which vertical class it belongs. In order to identify to which vertical category (TMVs, TEVs, etc.) the given concept (e.g. “skills”) actually belongs, a more thorough analysis of the team composition strategy is required, as elucidated in Section 4.4. The previous sections show that while the vertical categorization mainly represents *structural* differences among the team composition aspects and is hence appropriate for developing the meta model, the horizontal categorization captures variables’ semantics. Combined with the vertical classification it is possible to disambiguate variables while modeling a team composition strategy/model.

Identification of Necessary Meta Model Components

To begin with, the vertical categories are assumed to be the needed meta model components. In other words, these two categorizations are equal: each category of the vertical classification has exactly one homonymous representant in the meta model (meta model component). This assumption is refined in this subsection.

The following paragraphs show with the help of an example, how to determine which aspects of a team composition model are really necessary to be mapped to the meta model and hence to a team recommender. This is achieved by means of an empirical study on social factors on team composition (which belongs to the category of approaches presented in Section 3.1.1) [Brocco et al., 2010b]. Team composition aspects described in this study are analyzed with the help of the vertical and horizontal categorizations in order to identify which meta model component they actually belong to. Moreover, it can be verified whether all categories inside the vertical categorization (i. e. TMV, TIV, TEV, etc.) are necessary for instantiating a team composition model with the meta model for team recommendation. Since at this point it has been assumed that the vertical categories are meta model components, these verification refines also the selection of necessary meta model components.

The team composition model examined in the mentioned study is depicted in Figure 4.4. The arrows represent hypothesized influences. For instance, the study analyzed whether affect influences reliability. The considerations that are presented in the following, help to more clearly understand the scope of the vertical and horizontal categorizations.

According to the vertical classification the depicted variables belong to the categories team member variables, team intermediate variables, team external variables and output/outcome. Clearly, “task complexity” belongs to an aspect that influences teams from outside, and cannot be acquired from an individual team member. Hence, it falls into the vertical category (aspect entity) of team external variables. The variable “project success”

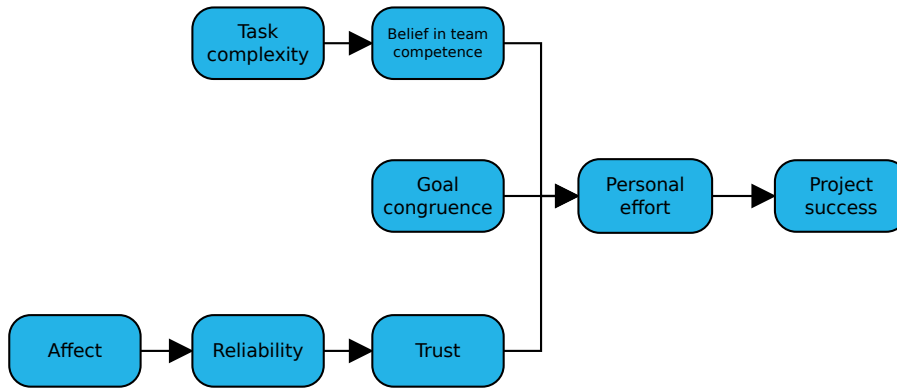


Figure 4.4.: Team composition model for social factors (cf. [Brocco et al., 2010b])

belongs to the output/outcomes class.

More challenging is the categorization of the remaining variables, namely, belief in team competence, goal congruence, personal effort, affect, reliability and trust. Again applying the vertical categorization “goal congruence” and “affect” as well as “belief in team competence” reasonably belong to TMVs since they are not dependent on prior variables. Since “Belief in team competence” is not dependent on prior team member variables (“task complexity” is a TEV) it is not a team intermediate variable, but a team member variable itself.

The categorization of aspect entities (vertical classification) would assign the variables “personal effort”, “reliability” and “trust” to the class of team intermediate variables because they seem to be dependent from prior variables. However, as mentioned at the beginning of this section, using this vertical classification only and not considering that these labels in the description of the team composition model denote only to the horizontal, semantic classification can lead to badly matched assignments. If the matrix described in Section 4.4 is used and the named variables are inserted as horizontal classes, it becomes evident that a more thorough diversification should be applied. For example, what notion of trust is used in this study? Is it actual trust, acquired, induced or resulting trust? Such questions help identify whether these variables really are team intermediate variables. The following paragraphs elucidate why this diversification is important.

The Role of Team Intermediate Variables

Section 4.3.3 defined team intermediate variables as “... *dependent on more fundamental prior variables and have indirect influence on the output/outcome of a team*”. The questions with respect to this definition and the findings based on the discussed team com-

position model is what role team intermediate variables play when recommending a team and what type of team is actually recommended. From a sociological and psychological point of view, team intermediate variables are essential, because their goal is to *explain why* certain effects can be observed in teamwork.

In the case of automated team recommendation however, the central goal is to predict team output/outcomes. When team intermediate variables influence a team member variable's impact, this impact will be reflected in dependencies used to map the influence of TMVs on the output/outcome. Thus, the function of team intermediate variables is redundant and they are not necessary for the recommendation process. For this reason three alternative approaches are proposed for the reduction of variables that, from the vertical categorization, appear to be intermediate variables.

- a) Consider excluding variables that are not **directly** necessary for your prediction goals. Some team intermediate variables are just used by team composition models to explain certain aspects that are interesting from an organizational, social or psychological perspective. For example, in Pelled's work [Pelled, 1996] visible demographic diversity is said to provoke emotional conflicts which in turn are said to decrease team performance. Emotional conflicts are only a consequence of the diversity and have just an explanatory role. In this case the variable can be neglected for team recommendation.
- b) On the other hand, if measuring group conflicts for other reasons is necessary (such as if your goal is to minimize them for training purposes), then this team intermediate variable is in fact an output you want to predict and should be regarded as *resulted group conflicts*. In that case this variable is not a team intermediate variable and should therefore be modeled as an output/outcome variable.
- c) If, in contrast to the above cases, the variable can be acquired before a project team has started (such as *actual trust* in the discussed social team composition model, cf. [Brocco et al., 2010b]), this variable has to be regarded as team member variable. This is the case, whenever it can be assumed that the given variable is not dependent on prior team member variables.

Again, the vertical/horizontal classification matrix presented in Section Section 4.4 can help choose from the above approaches. If the supposed team intermediate variable is actually a TMV (e.g. actual trust instead of induced trust) then approach c) should be chosen. In case the supposed team intermediate variable is an output/outcome variable (e.g. resulting trust) then the second approach should be applied. Only if by means of the vertical and horizontal classification the variable is identified as a true team interme-

4. A Meta Model for Team Recommender

diate variable the first approach should be employed and the variable should be excluded from the meta model instance that is to represent the team composition model used for recommendation.

Summary

In summary, the elements of the vertical classification (structural classification) presented in Section 4.3 suffice for constructing the base components of the intended meta model. Team intermediate variables are the exceptions, since they do not directly contribute to the performance prediction of a team because they are influenced by previous variables. However, variables that from the perspective of the vertical categorization may appear as team intermediate variables can have a different purpose. This can be identified with the help of the horizontal classification. For instance, when “conflicts” have to be mapped, it should be considered whether the conflicts are “past conflicts” (and hence TMVs) or “generated conflicts” (output/outcomes). If they are “induced conflicts” the first reduction alternative should be used since that would mean that they are not needed in the recommendation process.

Thus, the components that should be taken into account for instantiating team composition models by means of the meta model are:

- Team Member Variables
- Output/Outcome Variables
- Dependencies
- Team External Variables
- Constraints

The following sections more thoroughly describe and formalize these aspects and present some examples for the usage of these meta model components.

4.7. Meta Model Formalization

The previous sections delimited and characterized the problem of team composition. In addition, a concept for team recommendation, based on a meta model that models the aspect entities of the team composition domain, has been proposed. The meta model can

then be used for instantiating team composition strategies in recommendation algorithms. To be able to integrate this solution into a recommender system, the concept has to be formalized and further operationalized where necessary. The goal of this section is to provide mathematical structures that model the previously introduced meta model concept which consists of:

- Input Space, consisting of team member variables
- Output Space, consisting of output/outcome variables
- Dependencies that map the Input Space to the Output Space
- Team external variables that influence a team from the outside
- Constraints that have to be satisfied before a team is composed according to a model

Given these constituents, four main elements that make up a team recommendation system can be derived:

- a) *Team composition core models*: strategies and heuristics used to compose a team through the usage of team member variables, output/outcome variables and dependencies (these can, for example, be derived from the approaches introduced in Chapter 3).
- b) The *context* in which the team performs its actual task defined by the team external variables.
- c) Organizational *constraints* that may affect the set of possible teams available for a recommendation.
- d) *Algorithms* for actual recommendation using the above three elements (or a subset).

This section focuses on the first element, namely team composition core models. Thus, context, constraints and algorithms are not considered in this part. The goal of this restriction is to more clearly demonstrate how team composition models can be mapped using the meta model, irrespective of the actual context and constraints.

Below, the elements input space, output space and dependencies are examined first. Later parts of this thesis discuss in detail how team external variables and constraints can be integrated into an algorithmic solution for recommending teams.

4. A Meta Model for Team Recommender

Input Space: Every team member can be described by a set of team member variables (TMVs)

$$TMV = \{age, gender, number\ of\ contacts, competencies, \dots\}$$

which is operationalized, for instance, as:

$$TMV = \{[0, 120] \in \mathbb{N}, \{m, f\}, \mathbb{N}, \\ \{“programming”, “marketing”, \dots\}, \dots\}$$

Hence, a selection of such team member variables is defined as $TMV' \subseteq TMV$, e. g. $TMV' = \{age, gender\}$. Defining a Cartesian product space Q over TMV' as:

$$Q = tmv'_1 \times tmv'_2 \times \dots \times tmv'_{|TMV'|}$$

where $tmv'_i \in TMV'$, each team member is described by a tuple of properties $t_i \in Q$. In the above example, $Q = age \times gender$. An element of Q could for example be $(35, m)$. A team of N members is described by a set of N such tuples

$$t = \{t_1, t_2, \dots, t_N\}$$

The space of such team descriptions is denoted as T . This means that, for instance, $\{t_1, t_2\} \in T$ and also $\{t_4, t_2, t_7\} \in T$. In case TMV' consists of age and gender, a team member can therefore be described through the vector $t_1=(36,m)$. The whole team may then be described as $t=\{(36, m), (24, f), (54, f), (36, f)\}$.

A specification $q_i \subseteq Q$ defines a single team member's desired properties. E. g. a single team member desired to be in the age range from 20 to 40 where gender does not matter could be specified as $q_i = ([20, 40], \{m, f\})$. More complex specifications can be formalized using simple first order logic expressions or set operations respectively (the algebras are isomorphic): $(\vee, \wedge, \neg) \leftrightarrow (\cup, \cap, \bar{})$. A desired exact specification for a team can then be defined as

$$q = \{q_1, q_2, \dots, q_N\}$$

The space of such (exact) specifications is denoted as \tilde{Q} . This means that, for instance, $\{q_6, q_7\} \in \tilde{Q}$ and $\{q_4, q_2, q_8\} \in \tilde{Q}$. This type of specification is e. g. used when the project manager exactly knows what the properties of individual team members should be. For example, if members in a team should be at least 35 years old and two of them females, this (exact) specification can be expressed by the following set: $q = \{([35, 120], \{f\}), ([35, 120], \{f\}), ([35, 120], \{m, f\}), \dots\}$.

As will be shown in the paragraph “Dependencies”, it is not always necessary to specify q : particular requirements for a team can be modeled also without. For instance, if heterogeneity of competencies is desired, q is not needed. For such requirements q can be an empty set.

Output Space: The output space consists of the possible output/outcome variables of a team and can be formalized as such:

$$\begin{aligned} OUT &= \{ \textit{average satisfaction}, \textit{produced money}, \\ &\quad \textit{judgment of team leader}, \dots \} \\ &= \{ [0, 10] \in \mathbb{R}, \mathbb{N}, [0, 10] \in \mathbb{N}, \dots \} \end{aligned}$$

A selection $OUT' \subseteq OUT$ and an according Cartesian product space

$$O = out'_1 \times out'_2 \times \dots \times out'_{|OUT'|}$$

where $out'_i \in OUT'$, can be used to specify the outcome to be predicted (e.g. average satisfaction, produced money, judgment of team leader).

Dependencies: Dependencies are mapped in two steps:

a) *An analytic mapping*

$$\alpha : \tilde{Q} \times T \rightarrow \mathbb{R}$$

maps an optional desired team specification and an actual team description (both according to a selected set of team member variables) to a real value. The semantic of the analytic mappings α is dependent on the specific mapping which can, for example, be *heterogeneity*, *sum* or *exact* and *fuzzy matching*.

The *exact matching* analytic mapping was already briefly introduced in the paragraph “Input Space”. It can be used in team composition to set up a team with specific properties. An example for it could be a team consisting of exactly one manager. Q for this example is chosen as $Q = orgposition \times tmv'_2 \times \dots \times tmv'_{|TMV'|}$. The set $orgposition = \{ \textit{“manager”}, \textit{“secretary”}, \textit{“housekeeper”}, \dots \}$ consists of simple tags describing the organizational roles. According to that, a desired specification for a team consisting of two members, one manager and one non-manager role would be $q = \{q_1, q_2\}$ where $q_1 = (\{ \textit{“manager”} \}, tmv'_2, \dots, tmv'_{|TMV'|})$ and $q_2 = (orgposition \setminus \{ \textit{“manager”} \}, tmv'_2, \dots, tmv'_{|TMV'|})$. For a team $t \in T$ with $t_1 = (\textit{“housekeeper”}, x_2 \in tmv'_2, \dots, x_{|TMV'|} \in tmv'_{|TMV'|})$, $t_2 = (\textit{“manager”}, x_2 \in tmv'_2, \dots, x_{|TMV'|} \in tmv'_{|TMV'|})$ the specification is satisfied, i.e. $\exists t_i \in q_1$ and $\exists t_{j \neq i} \in q_2$. In this case, the *exact matching* analytic mapping α would return the value 1. If this is not the case, 0 is returned. A *fuzzy matching* analytic mapping can

4. A Meta Model for Team Recommender

compute a similarity between the desired specification and the actual team member variable provided by means of e. g. a taxonomy of organizational positions.

An example for an analytic mapping without exact specification is *heterogeneity*. Heterogeneity of nationalities, for example, can be mapped (operationalized) by Blau's index [Harrison and Klein, 2007], which in this case represents the α function. Accordingly in context of this model the query specification $q \in \tilde{Q}$ is an empty set $\{\}$ and the space Q over TMV' is $Q = \textit{nationality} \times \textit{age} \times \textit{gender} \times \dots \times \textit{tmv}'_{|TMV'|}$. E. g. the description of team member 1 could therefore be $t_1 = (\textit{British}, 28, \textit{male}, \dots)$, that of team member 2 $t_2 = (\textit{German}, 48, \textit{female}, \dots)$ and that of team member 3 $t_3 = (\textit{German}, 32, \textit{female}, \dots)$. For this team α would map the team $t = \{t_1, t_2, t_3\}$ to the value: $1 - \sum p_k^2$ (which is the formula for Blau's index) where k is the number of represented nationalities (in this case two: German and British) and p_k their proportion in the team t . In this case the result of this analytic mapping (Blau's index) would be $1 - ((\frac{1}{3})^2 + (\frac{2}{3})^2) = \frac{4}{9}$ which indicates a moderate heterogeneity (minimum would be 0, maximum 1) with respect to nationalities within the team. As has been said, this example for α does not require any query specification q .

b) *Heuristic mappings*

$$\chi : \mathbb{R}^i \rightarrow O_j$$

use the result values from i different analytic mappings to predict output/outcome variables, i. e. they relate the result of the analytic mappings to an output/outcome j of the Cartesian product space O . E. g. a heuristic mapping could map the heterogeneity (analytic mapping function) of team member nationality (team member variable) to the team performance (output/outcome). In this example, because there is only one analytic mapping (heterogeneity of nationality), $i = 1$.

The overall mapping $\alpha \circ \chi$ (where α could also represent a vector of i analytic mappings) describes the dependency of the predicted output/outcome from properties of a team (analytic mappings computed on TMVs). Hence, the analytic mappings used in a heuristic mapping are first computed in parallel and then passed as arguments to the corresponding heuristic mapping.

4.7.1. Formalization Summary

To summarize the various formalizations in this section, a team recommendation system S is defined by using the described components of the team composition meta model in

the following tuple:

$$S = (CON, TEV, TMV, A, H, OUT)$$

where CON is the set of available constraints (e. g. on time, organization, location, etc.), TEV , the set of available team external variables and TMV the set of available team member variables. A is the set of available analytic mappings $\alpha : \tilde{Q} \times T \rightarrow \mathbb{R}$ and H the set of the $\chi : \mathbb{R}^i \rightarrow O_j$ mappings that relate the results of the different analytic mappings to an element of the Cartesian vector space O consisting of a selection of elements of the output/outcome selection $OUT' \subseteq OUT$. OUT is the set of available output/outcome variables. Depending on the application domain and the algorithms applied on the meta model instantiations to recommend a team, not all the tuple elements have to be specified. For instance, team recommendation algorithms that do not consider the context (i. e. team external variables TEV) may omit this aspect. In this case the set TEV does not need to be specified and can be replaced by an empty set.

4.7.2. Meta Model Benefits

The above proposed meta model directly addresses some of the challenges described in Section 4.2. First of all, it allows for the modeling of the most important aspects that influence a team (**Challenge 1**). Recommender systems that make use of the meta model are not limited to particular variables or influence aspects.

Based on a diversification of team output/outcomes through corresponding variables of the set OUT , the meta model enables algorithms to evaluate teams accordingly (**Challenge 2**): For each instantiated team composition strategy a clear specification of a set OUT' including the evaluation criteria is required. The precise mechanism for ranking teams with more than one output/outcome variable is then dependent on the specific implementation (e. g. weights that can be parametrized by the user for linear combination).

As another advantage, depending on the algorithm used, specific combinations of this meta model components can be used to instantiate team composition models (**Challenge 5**) if required.

The following section demonstrates this latter part by instantiating some of the team composition models found in the literature review.

4.8. Modeling Team Composition Strategies

In this section the usage of the presented meta model is sketched by instantiating models for team composition found in the literature review presented in Section 3.1. Since the level of detail in the following models from literature is not always adequate enough for an algorithmic use (for details refer to Section 3.4), an operationalization of their suggested concepts is proposed where necessary. The focus for the example instantiations shown in this section is on the aspects belonging to the *team composition core model*. Where necessary, the specifications of the remaining meta model components (i. e. team external variables and constraints) are described in an informal way.

Pelled's model [Pelled, 1996] is the first example. It deals mainly with demographic diversities. The model is particularly suited for the mentioned focus, because of its emphasis on team member variables, dependencies (i. e. analytic and heuristic mappings) and output/outcomes.

The second team composition model discussed is a study on multicultural innovation teams performed by Bouncken and Winkler [Bouncken and Winkler, 2008]. Besides of its value in the more practical orientation, this article is also very suitable for the meta model approach due to the domain in which the study took place which relates well to the open innovation scenario described in Chapter 2.

The third example is the game theory approach to team composition developed by Mello and Ruckes [Mello and Ruckes, 2006]. It shows that the meta model can also be applied to theories beyond the classical statistical-based approaches used, for instance, in quantitative studies conducted in behavioral sciences.

4.8.1. Pelled's Model

As already described in Section 3.1.2, Pelled's Model shows how teams perform depending on demographic diversity. The approach is an extension of Ancona's work on demography and team performance [Ancona and Caldwell, 1989]. Pelled differentiates between two types of demographic diversities: visible and job-related.

Visible diversity can be assessed by observing the appearance of team members. Variables with high visibility for example are age or gender. In contrast, job-related diversity comprises all diversity aspects that can not be inferred easily from the appearance such as skills or educational background.

Pelled claims that visible demographic diversity negatively influences people’s satisfaction and team performance because it fosters affective conflicts. Furthermore, she claims that job-related demographic diversity has a positive influence on performance because it fosters substantive conflicts. Substantive conflicts can be seen as an expression of the friction between different mindsets. The effects of diversity on affective and substantive conflicts are dependent on how often and how long a team has worked together. This latter aspect is denoted as “team longevity”. According to the previously introduced classification team longevity is a typical team intermediate variable (see Section 4.3.3).

Team Member Variables and Output/Outcomes

To operationalize this model using the meta model, first, team member variables and the output/outcome of the team process have to be chosen. According to Pelled’s work, functional background (*functionalBG*), educational background (*educationalBG*) and organizational tenure (*orgtenure*) are selected as team member variables for job-related demographic diversity.

For the visible demographic diversity variables in turn *age*, *gender* and *race* are selected as team member variables. Pelled also identified the variable *group tenure* (team longevity), which by means of the reduction approach proposed in Section 4.6 is included as team member variable in the meta model instance. It can be pictured as a social network graph whose ties are working relationships in a team: the stronger the tie, the more times team members worked together in the past. The set TMV' for Pelled’s model is therefore

$$TMV' = \{functionalBG, educationalBG, orgtenure, \\ age, gender, race, grouptenure\}$$

where $TMV' \subseteq TMV$. Q as previously described is the Cartesian product space over TMV' and thus:

$$Q = functionalBG \times educationalBG \times orgtenure \times \\ age \times gender \times race \times grouptenure$$

As output/outcome the variables “judgment of team leader” (representing the “cognitive task performance” measure stated in the article) and “turnover” are selected.

$$OUT' = \{judgment\ of\ team\ leader, turnover\}$$

where $OUT' \subseteq OUT$.

Analytic and Heuristic Mappings

The dependencies respectively map functional background, educational background, organizational tenure and group tenure to judgment of team leader, and age, gender, race and group tenure to turnover and judgment of team leader (note that group tenure is used in both dependencies). A general overview of the resulting model is given in Figure 4.5.

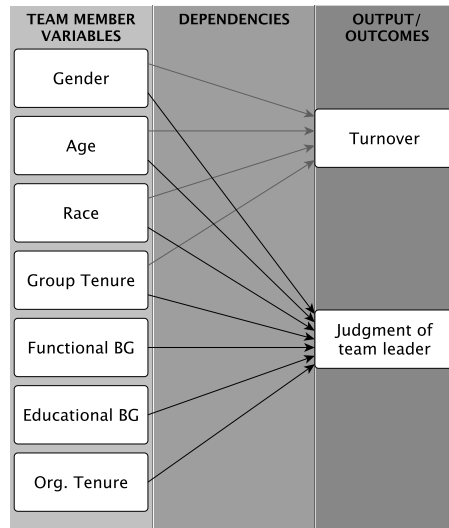


Figure 4.5.: Pelled’s model implemented by the meta model (dependency specifications omitted for clarity)

As mentioned in Section 4.7, dependencies are operationalized by two kinds of mappings: analytic mappings α and heuristic mappings χ .

In Pelled’s case two different types of analytic mappings can be identified. Both mappings are based on the concepts of diversity (or heterogeneity) mentioned in the specification of this composition model. The first stands for diversity as variety, such as in the case of educational background (α_1), functional background (α_2), gender (α_3) and race (α_4) in Pelled’s model. It can be operationalized by Blau’s index (*blauIndex*) as described in Section 4.7 in the paragraph “Dependencies”.

The second represents diversity as separation and is usually operationalized with the standard deviation (*stdDev*) [Harrison and Klein, 2007]. This type of analytic mapping is applied to the team member variables age (α_5), organizational tenure (α_6) and group tenure (α_7). In the following, these analytic mappings are labeled with the corresponding

TMV they are acting on:

$$\begin{aligned}
\alpha_{1,educationalBG}(\{ \}, t) &: blauIndex(t^{(2)}) & (4.1) \\
\alpha_{2,functionalBG}(\{ \}, t) &: blauIndex(t^{(1)}) \\
\alpha_{3,gender}(\{ \}, t) &: blauIndex(t^{(5)}) \\
\alpha_{4,race}(\{ \}, t) &: blauIndex(t^{(6)}) \\
\alpha_{5,age}(\{ \}, t) &: stdDev(t^{(4)}) \\
\alpha_{6,orgtenure}(\{ \}, t) &: stdDev(t^{(3)}) \\
\alpha_{7,grouptenure}(\{ \}, t) &: stdDev(t^{(7)})
\end{aligned}$$

where t is the set of the team to be evaluated. The second index of alpha is only added for explanatory reasons to emphasize in the notation the TMV to be used for the analytic mapping. This is actually represented in the index in brackets of t , which corresponds to the team member variable position in the Cartesian product space Q of TMV' .

In this model two heuristic mappings χ_1 and χ_2 can be identified: χ_1 maps the numeric results of α_3 , α_4 , α_5 and α_7 to turnover. This is formalized as described in Equation 4.2:

$$\chi_1(\alpha_{3,gender}, \alpha_{4,race}, \alpha_{7,grouptenure}, \alpha_{5,age}) = turnover \quad (4.2)$$

χ_2 takes the same analytic mappings as χ_1 plus α_1 , α_2 and α_6 and maps it to the judgment of the team leader as depicted in Equation 4.3.

$$\begin{aligned}
&\chi_2(\alpha_{3,gender}, \alpha_{4,race}, \alpha_{7,grouptenure}, \alpha_{5,age}, & (4.3) \\
&\alpha_{1,educationalBG}, \alpha_{2,functionalBG}, \alpha_{6,orgtenure}) = judgment\ of\ team\ leader
\end{aligned}$$

From Pelled's model only the sign of the slope of the arguments within the heuristic mappings χ_1 and χ_2 can be acquired: In χ_1 all analytic mappings except for α_7 have a positive slope. In χ_2 elements contained also in χ_1 have the opposite slope (except for α_7 on group tenure which still has a negative slope), while α_1 , α_2 and α_6 on organizational tenure have a positive slope. This results into the following equations for χ_1 and χ_2 :

$$\chi_1(\alpha_{3,gender}, \alpha_{4,race}, -\alpha_{7,grouptenure}, \alpha_{5,age}) = turnover \quad (4.4)$$

$$\chi_2(-\alpha_{3,gender}, -\alpha_{4,race}, -\alpha_{7,grouptenure}, -\alpha_{5,age}, \quad (4.5)$$

$$\alpha_{1,educationalBG}, \alpha_{2,functionalBG}, \alpha_{6,orgtenure}) = judgment\ of\ team\ leader$$

4. A Meta Model for Team Recommender

Since the type of relationship is not specified, the functions χ_1 and χ_2 have to be configured with a default mathematical function (linear, quadratic, etc.). E. g., χ_1 could be a linear function of the analytic mappings: $\chi_1 : a*\alpha_{3,gender} + b*\alpha_{4,race} - c*\alpha_{7,group\ tenure} + d*\alpha_{5,age}$. By applying the model consecutively and monitoring their output/outcome (thus creating training data for the model) the parameters of these heuristic mappings can be adjusted accordingly. However, this is not part of the meta model itself but part of the recommendation algorithm acting on the meta model (cf. Section 5.3).

4.8.2. Bouncken and Winkler Model

Another theory for composing teams was investigated by Bouncken and Winkler in their work about global innovation teams (Section 3.1.1, [Bouncken and Winkler, 2008]). They conducted a longitudinal study on five large multicultural innovation teams (some of them consisting of three sub-teams) over two years. They attempted to investigate how communication styles affect innovativeness of teams. The teams' domain was food and cosmetics. They focused on teams of a firm that was pursuing radical innovations. This could be translated in the meta model through appropriate team external variables (e. g. variables *domain*="food or cosmetics" and *innovativeness*="radical"). Although the strings themselves in the absence of a formal ontology do not appear to carry any formal semantics, these team external variables can be used for improving context awareness in the recommendation process. For example, they may be used to indicate to the project manager for which environment a specific model has been tailored. Moreover, algorithmic support for context awareness using team external variables is an interesting research question and can be addressed with appropriate approaches (e. g. the team recommendation algorithm proposed in Section 5.2).

Bouncken and Winkler's approach can be formalized through the definition of the meta model components team member variables, analytic mappings, heuristic mappings and output/outcomes. To identify which aspects have to be regarded in this model the main propositions of Bouncken and Winkler's work are summarized as follows:

- a) Diversity in *time perception* leads to lower performance.
- b) In teams which are not bi-national and not completely mixed (i. e. not 100% multi-national): If English *language skills* and *intercultural experience* of the national minority of a team are good, this improves acceptance within the team and thus improves *team performance*. A low *cultural distance* between the national minority and the majority of the team also improves team performance.

- c) In bi-national teams differences in *decision power* distance lower team performance.
- d) In bi-national teams differences in *cultural context degree* (i.e. whether context or explicit information plays a major role in the communication) negatively influence team performance.
- e) *National* diversity improves team performance.
- f) In completely mixed MCIT (multicultural innovation teams) strong English *skills* improve team performance.

Mixed teams in Bouncken and Winkler's nomenclature refer to multinational teams. These are teams whose members all belong to different nationalities. Thus, some of these propositions are valid in general, some for completely mixed teams, some for minority/majority teams and some others for bi-national teams. Team performance in the above case refers to "creativity and innovation" as stated in Bouncken and Winkler's work. Given these propositions, the team member variables are identified.

Team Member Variables and Output/Outcomes

One topic tackled by Bouncken and Winkler was diversity in terms of time perception. They distinguished "monochronic" and "polychronic" cultures. Monochronic cultures (e.g. German, French) prefer to perform jobs sequentially. Thus, they prefer to terminate a task before executing a new one. Moreover, they feel disturbed when interrupted while performing a task, unlike polychronic cultures (e.g. Egyptian). People belonging to this latter type of culture like to perform several tasks in parallel. Therefore a team member variable "time perception" has to be included whose domain comprises the values "monochronic" or "polychronic".

Intercultural experience is another essential team member variable and has to be included in the meta model instantiation. It can be operationalized in different ways. On the one hand it can be represented as the number of years of intercultural experience. On the other hand also a skill with the label "intercultural experience" could be used as TMV. In this example the first notion is employed.

The team member variable "language" is also included with an attached weight that indicates the level of proficiency in the corresponding language on a scale of 1 to 5 (e.g. "English",4).

Cultural distance is defined by the distance in terms of *time perception*, *space* and

4. A Meta Model for Team Recommender

cultural context. The first aspect is described above.

The aspect *space* refers to different requirements in terms of minimal required physical distance to a face-to-face communication partner to still feel comfortable (either “high” requirements, i. e. much space, or “low”). This indicates what the typical distances for communication of a person and culture are. These boundaries are necessary to identify for example at what physical distance from each other persons start to feel uncomfortable because of an insufficient distance from their communication partners. For instance, Hall identified that for western countries the intimate zone for a comfortable communication with close friends and family is 20 inches ($\approx 0,5$ meters).

As another team member variable, the degree of *cultural context* (compare Proposition d)) is indicated according to Hall and Hall’s specification [Hall and Hall, 1990]. This variable can assume the ordinal scaled values “*low*” and “*high*”. “Low” context indicates that the number of written rules in this culture is rather large, while “high” expresses that there are few written rules. “Low” context describes whether people belonging to this culture feel more comfortable to be direct and explicit. For instance, by specifying as precise and explicit as possible the information needed by the communication partner. In contrast, “high” context describes cultures that tend to prefer to build up a familiar context with their communication partner, making it therewith unnecessary to clearly and explicitly specify the information that has to be communicated. In “high” context cultures the contextual factors (e. g. mimic, environment of communication) often carry more information than the information itself. Cultures with a “low” cultural context are for instance the German, US American and Scandinavian. Cultures with a “high” context are, for example, the Italian, Indian or Latin American [Copeland and Griggs, 1986].

Summarizing, cultural distance is an aggregated aspect that can be computed by the three subaspects: time perception, space and cultural context. For this reason this aspect does not require to be modeled as a separate team member variable.

Decision power is a dimension defined by Hofstede [Hofstede, 1983] in a study about national cultures. He defines power distance as the team members’ perception of a team leader’s leading style. In some cultures strong hierarchical leadership with the team leader making autocratically decisions dominates. In contrast, other cultures are used to a more consultative way of making decisions, sometimes driven by majority vote within a team. Hofstede assigns different power distance scores depending on the nationality. For this reason this value could be directly derived from the TMV nationality. In this example a new TMV “power distance” is defined that can assume, according to Hofstede’s work, values between 11 (low power distance, consultative way) and 104 (high power distance, autocratic way). This range is defined by the maximal and minimal power distance he

determined for selected cultures.

Finally, diversity of nationality was said to improve team performance. Hence, “nationality” is added as team member variable of this team composition model.

The set of team member variables TMV' and corresponding vector space Q can thus be specified as:

$$\begin{aligned}
 TMV' &= \{degree\ of\ cultural\ context,\ time\ perception,\ space,\ intercultural\ experience, \\
 &\quad\ decision\ power,\ language,\ nationality\} \\
 Q &= degree\ of\ cultural\ context \times time\ perception \times space \times \\
 &\quad\ intercultural\ experience \times decision\ power \times language \times nationality
 \end{aligned}$$

where degree of cultural context and space are measured on ordinal scales that specify the degree of cultural context and the need for space respectively (i. e. “low” or “high”). Time perception is a nominal scaled variable that can assume the values “monochronic” or “polychronic”, decision power is an ordinal scaled variable ranging from 11 to 104, language a set consisting of string-value tuples (e. g. (“English”, 3), (“German”, 5)) and intercultural experience a variable representing the years of intercultural experiences (e. g. a positive number). The TMV nationality contains a string denoting the nationality of the team member candidate (e. g. “Italian”).

Even though Bouncken and Winkler mentioned creativity and innovation as output variables in their work, these were never treated separately. For this reason it is not required to have them separated in the set OUT' . Instead an aggregated output/outcomes variable is proposed that includes both outcomes. This resulting variable is denoted in this case as *team performance* and is hence the only element of the set OUT' .

Analytic and Heuristic Mappings

Bouncken and Winkler’s propositions can be adequately mapped to the meta model by using analytic and heuristic mappings.

Proposition a) can be mapped using the notion of variety. In case the team variety of time perception is close but not equal to 0 (0 means that all team members have the same time perception, i. e. all polychronic or all monochronic) team performance is negatively impacted. The variety can be operationalized with the index of quality variation which normalizes Blau’s index between 0 and 1 for an arbitrary number of value-classes.

4. A Meta Model for Team Recommender

This prevents having low values of heterogeneity in the binary case were only the value classes “monochronic” and “polychronic” are used for computation. For instance, for a team consisting of 2 people belonging each to the “monochronic” and “polychronic” class, Blau’s index would result in the value 0.5. The index of quality variation on the other hand will regard that there are only 2 classes available and thus result in the value 1 (maximum heterogeneity), because it divides the result of Blau’s index with its theoretical maximum [Harrison and Klein, 2007]. This represents the scenarios with the highest variety of time perception. Thus, the first analytic mapping α_1 is the variety of time perception operationalized for example with the index of quality variation (*IQV*).

The first two aspects of Proposition b) can be expressed as: the better the English skills and the intercultural experience the better the team performance. As a consequence, these can be used by creating two analytic mappings. The first analytic mapping “average matching minority” (*avg_matching_minority*) extracts the English skills of the team members that do not belong to the national majority of the group and averages their proficiency levels in the English language (α_2): The better their proficiency, the higher the value of this analytic mapping. This minority corresponds to the members which do not belong to the majority. The majority nationality in turn can be computed by using a majority algorithm (e. g. [Rothstein, 1999]). The second analytic mapping α_3 averages the years of intercultural experience of the part of the team which is not within the national majority (*avg_minority*).

Cultural distance as last influence factor of Proposition b) can be measured as the average of the pairwise distances of the TMVs cultural context, space and time among team members. These analytic mappings α_4, α_8 and α_9 could hence be denoted as “pairwise average” (*pairwise_avg*) and act respectively on the three TMVs *degree of cultural context*, *time perception* and *space*.

In Proposition c) the topic of power distance is addressed. In this case, the power distance TMV encodes the power distance index according to Hofstede [Hofstede, 1983]. Therefore the analytic mapping “separation” is used that computes the standard deviation of power distance within the team (*stdDev*). High values for this analytic mapping represent very different attitudes concerning the style of team leading and decision making.

In a bi-national team the difference in cultural context can negatively influence team performance (Proposition d)). To operationalize this type of difference there are several possibilities. On the one hand, if it is known that the cultural context of a bi-national team has to be investigated, it can easily be assessed by an analytic mapping whether the context values “high” and “low” are both present within the team which demonstrates that there are differences in this regard. Otherwise, using Blau’s index it can be ensured

whether the variety of cultural context α_6 is 0.5 which indicates that half of the team members have the cultural context value “high” and the other half “low” (*blauIndex*). Again: a value smaller than 0.5 would indicate that there is a majority in the team as described for the for the TMV *time perception* in Proposition a). This can be combined with an analytic mapping α_7 (e. g. again Blau’s index) that measures the variety of the nationality of a team to measure whether the team is bi-national or not. Moreover, this latter analytic mapping can also be used to map Proposition e).

The last proposition of Bouncken and Winkler’s work is that obviously in MCIT team members have to speak English very well. This can again be operationalized with the analytic mapping α_2 which can be reused for this proposition. For this reason no additional analytic mapping is needed. In summary these are the used analytic mappings:

$$\begin{aligned}
\alpha_{1,timeperception}(\{\}, t) &: IQV(t^{(2)}) & (4.6) \\
\alpha_{2,language}(\{q_1\}, t) &: avg_matching_minority(t) \\
\alpha_{3,interculturalexperience}(\{\}, t) &: avg_minority(t) \\
\alpha_{4,degreeofculturalcontext}(\{\}, t) &: pairwise_avg(t^{(1)}) \\
\alpha_{8,timeperception}(\{\}, t) &: pairwise_avg(t^{(2)}) \\
\alpha_{9,space}(\{\}, t) &: pairwise_avg(t^{(3)}) \\
\alpha_{5,decisionpower}(\{\}, t) &: stdDev(t^{(5)}) \\
\alpha_{6,degreeofculturalcontext}(\{\}, t) &: blauIndex(t^{(1)}) \\
\alpha_{7,nationality}(\{\}, t) &: blauIndex(t^{(7)})
\end{aligned}$$

where

$$q_1 = (\text{degree of cultural context, time perception, space, intercultural experience, decision power, (“English”, [1,5]), nationality})$$

As has been said, in α_2 and α_3 it is necessary to identify the minority of the team. This is done by ascertaining who does not belong to the majority of the team (the nationality of the majority of the team can be computed with Rothstein’s algorithm [Rothstein, 1999]). Note that these two analytic mappings act on the complete team member vectors of the set t instead of on a specific team member variable $t^{(j)}$, $j \in [1, |TMV'|]$ since they require more than one team member variable (namely nationality and language) to operate.

4. A Meta Model for Team Recommender

This team composition model is quite complex to construct since it does not provide details on the operationalization (cf. Section 3.4). This is especially the case for the construction of heuristic mappings. Since only one output/outcome variable is provided, only one heuristic mapping χ_1 is required that maps the results of all mentioned analytic mappings $\alpha_1 - \alpha_9$ to the output/outcomes variable *team performance*:

$$\chi_1(\alpha_1, \text{timeperception}, \alpha_2, \text{language}, \alpha_3, \text{interculturalexperience}, \\ \alpha_4, \text{degreeofculturalcontext}, \alpha_5, \text{decisionpower}, \alpha_6, \text{degreeofculturalcontext}, \alpha_7, \text{nationality}, \alpha_8, \text{timeperception}, \alpha_9, \text{space}) = \text{team performance}$$

Apart from that, it is also clear that α_7 should be used to moderate the influence of the propositions on the evaluation of the teams since some propositions apply for bi-national, some for completely mixed and some for teams whose majority of team members has the same nationality. In the case of α_7 , for example, results in the value 0.5 indicating a bi-national team, Proposition c) and d) should be used in the heuristic mapping to evaluate team performance. If a mixed team is recognized, Proposition f) and the according analytic mapping α_2 have to be weighted more. Bouncken and Winkler's work does not indicate whether in teams with minorities the proposition concerning bi-national teams should also be somehow regarded. For this reason assumptions have to be taken to complete the operationalization of this model.

Thus, in terms of heuristic mappings, the propositions themselves can be mapped using linear functions in absence of a more detailed description. For instance, for Proposition c) a linear function with negative slope can be used which relates higher power distance to lower team performance. All propositions have to be integrated in this manner in the heuristic mapping χ_1 with the goal to weight the corresponding propositions depending on the distribution of the nationalities. These aspects are not further developed in this section for reasons of clarity. Chapter 5 provides many other precise examples for such operationalizations.

4.8.3. Mello and Ruckes Model

A different approach described in Section 3.1.3 for the analysis of the impact of heterogeneity on performance has been applied by Mello and Ruckes [Mello and Ruckes, 2006]. They use a game theory model to analyze how far heterogeneous teams affect team performance. This section instantiates the meta model in an iterative manner by adding and integrating subsequently aspect entities of this team composition model.

As mentioned in Section 3.1.3 performance is specified as the cumulated payoff of the team. The team members are described by a set of team member variables consisting of the variables *training* (as a source of knowledge) and *organizational tenure* (as an indicator for the skill of correctly judging opportunities). From their literature review Mello and Ruckes infer that a team has to have a team leader such as a CEO or a division head, i. e. another according team member variable specifies their organizational *role*.

The TMV' set hence consists of these team member variables, each containing structured or unstructured elements according to their specification. The TMV “attended trainings” can for instance be a set containing the name of all training courses offered by the organization. A subset of it in the team member vector t_i expresses all training courses the team member i attended. The set “role” could be a hierarchical structure of roles. The organizational tenure set “orgtenure”, which measures the tenure in years, is equal to a suitable subset of \mathbb{N} .

$$TMV' = \{attended\ trainings, orgtenure, role\}$$

The first analytic mapping α_1 is defined as follows: it is an *exact matching* mapping over the role within the company. The related query $q \in \tilde{Q}$ is:

$$\begin{aligned} q &= \{q_1, q_2\} \\ q_1 &= (attended\ trainings, orgtenure, \\ &\quad “CEO” \cup “division\ head”) \\ q_2 &= (attended\ trainings, orgtenure, \\ &\quad role \setminus \{“CEO” \cup “division\ head”\}) \end{aligned}$$

This analytic mapping filters all teams consisting of exactly one division head or CEO. All other teams are not going to match and could return for instance the value 0 instead of 1. In this example and in the following a team comprises only two team members (for reasons of clarity).

Mello and Ruckes assume that both team members may acquire information about the prospective project’s payoff. Depending on the similarity of knowledge, organizational tenure and attended trainings, the information sources will be very similar or very different. The authors showed that in cases of high uncertainty and rapidly changing environments a heterogeneous team is more valuable, because of the high variety of the information collected. Mello and Ruckes derived that firms operating in more uncertain environments should tend to fill managerial positions with people from outside the organization. In

4. A Meta Model for Team Recommender

contrast to that, firms in stable environments should hire managers from inside the organization.

This again leads to the addition of another team member variable and to an extension of the parameter q in the analytic mapping α_1 .

$$TMV' = \{attended\ trainings, orgtenure, role, inside\}$$

where *inside* is a set of the logical values “true” and “false” that defines whether the person is inside or outside the organization (also other operationalizations are possible). The above paragraph raises another important issue with respect to the meta model concerning *team external variables*. Mello and Ruckes distinguish environments that their respective findings can be applied to. Depending on the TEV “environment”, that can take the values “certain” or “uncertain” a heuristic mapping χ_1 changes the sign of the result of the analytic mapping α_1 .

This in turn could be modeled through the meta model by instantiating two different models that have to be applied according to the TEV “environment”.

The first model instantiation for uncertain environments would then implement the following query q in α_1 :

$$\begin{aligned} q &= \{q_1, q_2\} \\ q_1 &= (attended\ trainings, orgtenure, \\ &\quad “CEO” \cup “division\ head”, inside) \\ q_2 &= (attended\ trainings, orgtenure, \\ &\quad role \setminus \{“CEO” \cup “division\ head”\}, “false”) \end{aligned}$$

the second model instantiation for certain environments would be:

$$\begin{aligned} q &= \{q_1, q_2\} \\ q_1 &= (attended\ trainings, orgtenure, \\ &\quad “CEO” \cup “division\ head”, inside) \\ q_2 &= (attended\ trainings, orgtenure, \\ &\quad role \setminus \{“CEO” \cup “division\ head”\}, “true”) \end{aligned}$$

As described before, the only output variable of the set OUT' in this team composition model is the overall payoff. Hence, the model instantiation consists of the previously declared team member variable set TMV' , the dependency $\alpha_1 \circ \chi_1$ (χ_1 maps the result of α_1 to the payoff: the better the matching in α_1 the better the payoff), the TEV “environment” and the output/outcome variable overall payoff.

4.9. Summary

This section analyzed the team recommendation problem and motivated when and why team recommendation makes sense. It has been discussed that team recommendation is a more challenging recommendation problem than the traditional item recommendation. This can primarily be accounted to the fact that the items are actually **sets** of items and that these sets are very **dynamic** in nature.

In order to find an appropriate solution for algorithmically recommending teams in the domain of open innovation, several studies from literature and approaches from practice have been analyzed from which a set of aspect entities could be identified that characterizes the team composition problem.

Section 4.4 emphasized the fact, that there are actually two categorizations that have to be taken into account: a vertical categorization describing structural properties of team composition aspects and, a horizontal categorization describing their semantic class and domain.

After this refinement, an adequate approach for team recommendation had to be developed. Two general possibilities were discussed. The first possibility was consistent with current work on algorithmic team composition found in related work and presented in Section 3.3: developing a team recommender system which is tailored for a specific use case. Since in open innovation this approach would be rather restrictive, the second possibility was chosen: developing a generic framework that allows for the instantiation of different strategies to compose teams. Advantages and drawbacks of both approaches were also discussed.

Section 4.6 showed which team composition aspect entities are finally used as components to develop this generic approach. It analyzed the vertical categorization in terms of importance for team recommendation tasks and showed an example that demonstrates that team intermediate variables (which was one of the identified aspect entities) do not contribute directly to the recommendation process and they should hence be disregarded for team recommendation. Therefore they are not included as component in the meta model in contrast to the other aspect entities contained in the vertical categorization.

Afterwards, these selected meta model components were utilized to formalize the components of this generic system which was denoted as **meta model** for team recommendation. With the help of this formalization three team composition models presented in Chapter 3 were instantiated to demonstrate the meta model's functionality.

4. A Meta Model for Team Recommender

5. Application in Recommendation Algorithms

The previous chapter provided the basics and principles behind the idea of the meta model for team recommender systems and demonstrated how the meta model can be used to instantiate team composition strategies. As stated in Section 4.7, a team recommender system needs four main ingredients to recommend a team. The first three are strictly related to the meta model itself: First, an instantiation of a team composition strategy, denoted as *team composition core model*, is required (that may consist of the specification of team member variables, analytic and heuristic mappings and output/outcomes). Second the environment in which the team has to perform must be specified through the definition of appropriate team external variables. Last, organizational constraints can be integrated. At this point, a common structure is provided which, however, is insufficient to recommend teams. For that purpose appropriate recommendation *algorithms* have to be applied. The use of the meta model in recommendation algorithms can demonstrate that the meta model is a valid framework on which to base team recommenders. The meta model alone cannot address all team recommendation challenges described in Section 4.2 to demonstrate its validity. For this purpose the meta model has to be employed in algorithms.

Which recommendation algorithm is appropriate for a given task, depends on many factors and variables. In many cases the team composition model influences the choice of the algorithm used. For instance, if the strategy requires that similar problems should be solved by similarly good teams, a case-based recommender may be well-suited. In other cases project managers define strategies that take into account when team member candidates are available and what skills they should have. According to this they directly choose appropriate candidates to be contacted. In this case no learning is required and a constraint-based approach can be the correct choice.

To show the usage of the described meta model for team recommendation in recommender systems, three different prototypes for team recommender have been developed. The following sections present these recommender systems and the meta model components selected for each of them. The first section presents a constraint-based recommender that implements the team composition model derived from the interviews in Section 3.2.1.

5. Application in Recommendation Algorithms

The second section discusses a case-based recommendation approach that accounts for past projects (i. e. especially different team external variables). The third approach can be regarded as hybrid recommender with switching technique that accounts for the peculiarities of the team recommendation problem and its associated challenges and requirements described in Section 4.2.

The first two algorithms show how the meta model can be used in state-of-the-art approaches with only minor adaptations, i. e. : only adequate mappings needed to be developed to adopt the meta model in these systems. These mappings are presented in the following section as one contribution of this thesis.

The final algorithm is a new approach developed as another contribution of this study that highlights the advantages of the meta model. This recommender makes use of team composition models and/or existing machine learning and data mining techniques to recommend teams. Similar to the other two sections describing the above-mentioned algorithms, the section concerning this algorithm also presents how the meta model components can be mapped to the learning techniques used in the developed algorithm.

5.1. Constraint-based

The team composition/recommendation problem can be considered as a problem belonging to the constraint satisfaction problem (CSP) class (**Challenge 7**): Resources (team members) have to be assigned to job positions with specific requirements and constraints. For this reason modeling the team composition problem as a constraint satisfaction problem is promising for certain scenarios. Therefore, a constraint-based recommender system has been developed as one of the prototypes presented in this thesis that solves the team recommendation problem with the help of constraint reasoning (Section 2.2.2.4).

Constraint-based team recommender systems directly address some of the requirements derived in Section 4.2: First of all, being a knowledge-based recommender system, a constraint-based recommender is not affected by the cold start problem. Thus, sparsity of data does not affect the quality of recommendations (**Challenge 4**). Second, the interaction patterns of a constraint-based recommender systems are very similar to database queries which is an advantage in terms of acceptance for real world scenarios (**Challenge 6**). In the following the essential parts of this recommender system are presented. Details on this approach can be gained from [Brocco et al., 2011b, Andergassen-Sölva, 2010].

5.1.1. Selected Meta Model Components

A constraint-based recommendation approach demands that constraints are formulated on specific variables with corresponding domains (refer to Section 2.2.2.4). Hence, in general constraints can be applied to all the meta model components represented by variables with standard type of scale (nominal, ordinal, interval, ratio). Thus, in the meta model case, constraints of a CSP can in principle make use of the meta model components team member variables TMV , analytic mappings A , team external variables TEV and output/outcome variables OUT as variables and domains of a CSP or constraints CON , as the name says, as constraints of the CSP. Heuristic mappings are used in the meta model to predict outcomes. Therefore they can not be *directly* applied to this approach. Therefore, at first sight, it seems that they are inappropriate for constraint-based recommendation. For this reason it is challenging to find an adequate representation of heuristic mappings in a CSP. Where and how heuristic mappings are nevertheless included in this approach is explained later in this section.

Despite the fact that all other meta model components can generally be employed in constraints for constraint-based recommendation, it is beneficial to restrict the application of certain meta model components to certain parts of the recommender only. For instance, it rarely makes sense to formulate constraints on variables of the set OUT . To analyze which meta model components should be used in which part of the recommender system some of the results of the interviews presented in Section 3.2.1 were used. More precisely, it was of interest to find out which meta model components are used by project manager of the ICT domain to formulate requirements for a team. On the basis of their best practices the consideration of certain meta model components for the recommendation approach has been restricted (**Challenge 1**). Following is a brief overview of these design decisions.

Team member variables are the basis of the presented recommendation approach. The interviews showed that in practice it is very common to express specific requirements on properties of the team members to compose a team. Generally TMV are directly used by two other meta model components: constraints (CON) and analytic mappings (A). The constraints represented by the set CON are very easy to map to the constraint-based recommendation system since they are hard constraints themselves by definition. Analytic mappings such as the fuzzy matching of competencies within a team cannot be mapped to hard constraints. In this case soft constraint reasoning should be used to assess the similarity between the specification q and the corresponding TMV in a team.

Regarding the output/outcomes variables OUT and team external variables TEV , as the interviews in Section 3.2.1 showed, these are not directly used to do a first selection of possible team members out of the pool of possible candidates. For this first step only

5. Application in Recommendation Algorithms

the components *CON*, *TMV*, and *A* are required. *TEV* and *OUT* are used in a second step when first results are already provided. They are used for the inspection of all past project data (specified through an arbitrary set *TEV'* and *OUT'*) of recommended teams.

This team recommendation system does not directly make use of heuristic mappings, since it completely relies on the knowledge encoded in similarity measures and the constraints specified by the project manager or user as queries. However, the semantic role of heuristic mappings can be recognized in other parts of the recommender system. The following section discusses also this topic among other things.

To summarize, the meta model components explicitly used to generate recommendations are the team member variables *TMV*, the analytic mappings *A* and the constraints *CON*.

5.1.2. Recommendation Approach

As a first step towards a possible mapping from the above described meta model components to the constraint-based recommender system it is advantageous to revisit the notion of hard and soft constraints described in Section 2.2.2.4.

Then, an appropriate classification of hard and soft constraints has to be constructed. Finally a more detailed concept for the combination of hard and soft constraint components can be developed.

Constraints representation

In principle, all *TMV* are appropriate candidates to be utilized in constraints in the constraint-based recommender. However, it has to be considered that **hard constraints** definitively filter out teams or team members that do not satisfy them. It therefore rarely makes sense to map aspects such as the matching of skills (which is an analytic mapping) as hard constraint. It only makes sense when it is a priori known, that there are enough assignments that can satisfy the constraint. Otherwise it will lead to overconstraintment. In contrast to analytic mappings, the meta model component *CON* (constraints) is always modeled as hard constraint. As previously mentioned *CON* is a set of organizational constraints (e.g. availability, limitation of partner organizations). In accordance to the project managers' needs, in most of the cases these constraints must be met (e.g. availability). Thus, they are convenient to be mapped as hard constraints in contrast to the other selected meta model components (e.g. analytic mappings).

For the part concerning the **soft constraints** of the recommender it is advantageous to use aspects whose degree of satisfaction can be used as criterion for the quality of the recommendation, such that the teams can be ranked accordingly. Analytic mappings are a suitable alternative, as they express desired properties of teams. The combination of the values resulting from analytic mappings can represent the satisfaction degree of the soft constraint problem. For instance, skill matching can be implemented as soft constraint with the use of the analytic mapping “fuzzy matching” that returns a value between 0 and 1 that specifies how well a specified set of skills matches the skill profile of a member or team. As another example, if a team has to be heterogeneous in terms of nationality, this can be expressed by an analytic mapping variety (nationality) that returns values between 0 and 1 representing the degree of heterogeneity of the team. All analytic mappings can be aggregated (in a parametric manner) to a single value that can be used as final value to rank the teams.

The next step consists in modeling the above described meta model components as constraint satisfaction problem. The computation of a solution for this resulting CSP is the main task of this team recommender system.

According to Meseguer et al. [Meseguer et al., 2003] a constraint satisfaction problem consists of a triple (X, D, C) where

- $X = \{X_1, \dots, X_n\}$ is a set of variables
- $D = \{D_1, \dots, D_n\}$ is the domain the corresponding variables X_i belong to
- $C = \{C_1, \dots, C_r\}$ is the set of constraints that have to be satisfied

The solution of such a triple is a value set for the variables in X (known as *assignment*) that satisfies this problem specified with the constraints in C . The meta model components can be mapped as follows using two different CSPs. The first (denoted as CSP_1) by using hard constraints, the second by soft constraints (CSP_2). For CSP_1 the corresponding mapping from meta model components to the CSP is as follows:

- $X = \{tmv'_1, \dots, tmv'_{|TMV'|}\}$ is the set of variables TMV'
- $D = \{D_1, \dots, D_{|TMV'|}\}$ is the domain the corresponding variables TMV'_i belong to
- $C = \{CON'_1, \dots, CON'_r\}$ is the set of organizational constraints that have to be satisfied

For each team member candidate CSP_1 is evaluated. Team member candidates that satisfy CSP_1 are forwarded to the soft constraint part CSP_2 :

5. Application in Recommendation Algorithms

- $X = \{tmv'_1, \dots, tmv'_{|TMV'|}\}$ is the set of variables TMV'
- $D = \{D_1, \dots, D_{|TMV'|}\}$ is the domain the corresponding variables TMV'_i belong to
- $C = \{\alpha_1^*, \dots, \alpha_r^*\}$ is the set of analytic mappings A

where α_i^* is a representation of the analytic mapping α_i in constraints. For example, for the analytic mapping α_i describing the variety of nationality a corresponding α_i could be Blau's index *blau* (cf. Section 4.7) applied on the nationalities $n \subset nationalities \in TMV'$ (thus $\alpha_i = blau(n)$). To use this notion as constraint, it is necessary to map α_i to a constraint. The constraint, in the case high variety is desired, is $\alpha_i^* : \alpha_i = \max_{x \in 2^{nationalities}} blau(x)$ or correspondingly $\alpha_i = 1$. Note that X in CSP_1 and CSP_2 are the same since it is possible to apply hard and/or soft constraints on the same TMV s if required.

The CSP declaration above is incomplete, because in order to be considered as soft constraints, an optimization step is required. For this purpose *valued constraints* can be used [Živný and Jeavons, 2009]. Valued constraints assign costs to express the degree of satisfaction for each constraint:

$$\phi : C \mapsto P$$

where the cost function ϕ maps a constraint $c \in C$ to a valuation $p \in P$. The valuations P must belong to the same valuation structure $S(P, \oplus, \succ)$ whose properties are described in [Schiex et al., 1995].

Finally, the costs for each single soft constraint ϕ_c can be aggregated in a global cost function

$$V_{CSP_2} = \bigoplus_{c \in C} \phi_c(c)$$

For instance, in case the soft constraint α_1 is a matching function that matches required skills to the skills of the team members. Accordingly, a cost of 0 is applied, if the skills match perfectly. If not, the cost function must be adapted according to a similarity function: the more similar the skills are, the less costs are computed by the cost function. In the variety example presented in the above paragraph, for each co-occurrence of nationalities a penalty value can be added to the total costs of the constraints given an assignment a of the variables $var(\alpha_i^*) \subset X$. $var(\alpha_i^*)$ denotes the ordered set of variables $\subset X$ that appear within the constraint α_i^* (also known as constraint *scheme* or *scope*).

The aggregation in the global cost function V_{CSP_2} as well as the exact transformation from analytic mappings to soft constraints (see above) should be parametrized according to the desired team composition model/strategy. It is interesting to observe that despite heuristic mappings and output/outcome variables are not explicitly used in this approach,

they are nevertheless implicitly utilized for those purposes: The parametrization in the global cost function implicitly takes the role of **heuristic mappings**, since it specifies which analytic mapping affects the output and to what extent.

For example, if the desired heuristic mapping is a linear function of two analytic mappings α_1 and α_2 , V_{CSP_2} linearly and inversely combines the cost functions $\phi_{\alpha_1}(\alpha_1)$ and $\phi_{\alpha_2}(\alpha_2)$ leading to costs that are anti-proportional to the output/outcome variable to be maximized. Thus, in this case $\chi(\alpha_1, \alpha_2) = -V_{CSP_2}(\phi_{\alpha_1}(\alpha_1), \phi_{\alpha_2}(\alpha_2))$. The output/outcome variable to be maximized depends on the team composition model. However, in this constraint-based recommender it is only possible to use one output/outcome variable (e. g. user preference) which is used for ranking the results.

Also the choice whether a mapping should be an exact matching (or maximal heterogeneity, minimal amount, etc.) is usually part of a heuristic mapping which can express whether a high or low value is beneficial for the corresponding output/outcome.

Furthermore, when a project manager requires a team from the recommendation system, she already implicitly decided upon which output/outcome variable to evaluate. This (perhaps aggregated and/or fuzzy) output/outcome variable is also not explicitly specified.

Interaction Paradigm

Given the above-mentioned mappings from meta model components to a constraint-based recommendation algorithm, it is necessary to use queries in order to communicate with the recommender system (**Challenge 6**).

On the one hand the queries must contain the hard constraints defined in CSP_1 , and, on the other hand the soft constraints of CSP_2 . The association of query terms to CSP_1 or CSP_2 is made according to the team composition strategy: *CON'* aspects are already encoded as constraints, and can be directly used as constraints for CSP_1 . All analytic mappings are part of CSP_2 . For each a constraint has to be specified according to the properties of the mapping. For example, the “matching” mapping or the “ideal combination” mapping need parameters to be inserted in the constraint while other analytic mappings such as the homogeneity of a specific TMV do not need additional parameter specifications.

After inserting the parameters required for the hard constraints in CSP_1 and soft constraints in CSP_2 , the constraint-based recommender first filters out all persons and/or teams whose properties do not satisfy the constraints in CSP_1 . Then, all analytic mappings are transformed into constraints that expect maximum/minimum values, i. e. maxi-

5. Application in Recommendation Algorithms

mum/minimum similarity, maximum/minimum heterogeneity, maximum/minimum homogeneity, maximum/minimum sum, etc. When these constraints are not satisfied, penalty values for each analytic mapping $\phi_{\alpha_x^*}(\alpha_x^*)$ are applied and finally combined in a global cost function V_{CSP_2} . Finally, the constraint solver returns the top t teams with the lowest cost values associated together with their past history data as a team (TEV' and OUT') if available.

Analytic mappings as hard constraints

In order to reduce the search space, in certain cases it may be advantageous to map analytic mappings (which are usually mapped to soft constraints) to hard constraints. The advantage resides in the fact that hard constraints filter team member candidates and therefore improve the speed of the recommendation generation (**Challenge 3**). However, in certain cases it is not inherently clear how to map analytic mappings to hard constraints.

This can be accomplished for instance by introducing threshold values. Thresholds can be used to limit the size of the result set or to guarantee a certain quality. For example, competence matching implemented as “fuzzy matching” returns a value that expresses how well the competencies of a certain team member candidate match the requirements of a specified job position. Here one type of threshold may set the minimum compatibility value so that team members exhibiting lower compatibility to the job position are excluded from the result set of CSP_1 . The other threshold option is to limit the result set to the best N team members for each job position in terms of competency compatibility.

However, this transformation of analytic mappings to hard constraints only brings benefits, when the analytic mapping can be decomposed into analytic mappings that can be computed for each team member separately, as in the case of competencies: for each job position one “fuzzy matching” on competencies can be performed. In the case of heterogeneity, optimal combination or other similar analytic mappings, no performance advantage can be gained since the combination of team members is performed on the whole team. Therefore, all teams instead of single team members have to be analyzed.

5.1.3. Algorithm Summary

Summarizing, to generate recommendation with this recommender system the following steps are required:

- a) Assume that TMV' , CON' and dependency functions (especially desired analytic

mapping values) are already specified i. e. according to the team composition model/strategy chosen. Furthermore, costs for all soft constraints α_i^* are configured (also according to the team composition model).

- b) The constraint-based recommender eliminates all team members candidates and teams for which no assignment can be found that satisfies CSP_1 .
- c) For different team compositions the analytic mappings “masked” as CSP_2 are evaluated with the results of b).
- d) Costs for each constraint that could not be completely satisfied are assigned in CSP_2 for each team and then combined in a global cost function V_{CSP_2} .
- e) The top N teams with the smallest cost values are returned and if past projects performed by exactly the same team can be found in the database, these are also returned attached to the team recommendations.

Certainly, it could be possible to tailor the last step to allow for the inspection of past projects performed by only a certain part of the team. Furthermore, also projects could be retrieved where all team members were part of a larger team. However, this may lead to wrong conclusions about the team. For this reason, it is better to present only projects that were performed by the entire team rather than a subset. As feature of a corresponding graphical user interface however, it could be provided by clicking on the individual team members to view the projects they participated in.

Example

The above described steps are more clearly elucidated with the help of an example for composing a team consisting of three members. As a first step, the definition of TMV' and CON' is required. This has to be performed according to the team composition strategy. For instance, as set TMV' the team member variables *Belbin's team role* (for details on these roles please refer to [Belbin, 2010]), *age* and *organizational tenure* in years and *affiliation* (denoting the company or institution the candidate is member of) are selected.

As constraints for CSP_1 the set C of the constraint satisfaction problem is defined as:

$$\begin{aligned}
 CON'_1 &: && \text{organizational tenure} \geq 5 \\
 CON'_2 &: && \text{affiliation} = \text{"TU Munich"} \text{ OR } \text{affiliation} = \text{"ETH Zurich"} \\
 CON'_3 &: && \text{Member}_1ID \neq \text{Member}_2ID \\
 CON'_4 &: && \text{Member}_2ID \neq \text{Member}_3ID \\
 CON'_5 &: && \text{Member}_1ID \neq \text{Member}_3ID
 \end{aligned}$$

5. Application in Recommendation Algorithms

This selects all candidates that are affiliated to TU Munich or ETH Zurich and have been working in their company for at least five years. Apart from that, it is ascertained that all team members are different such that a team member cannot be assigned twice to the same team.

The example team composition strategy devised for this example implies that the smaller the age differences in the team are, the better the outcome is. Moreover, in order to balance the strengths and weaknesses of the team, the chosen team composition strategy indicates to have different Belbin roles in a team rather than having a team composed of members with the same team role.

For this purpose two analytic mappings α_1 and α_2 have to be defined. α_1 is applied to the age and computes the standard deviation among the team members. The value for α_1 can be normalized for a more comfortable combination with other analytic mappings. α_2 is applied on Belbin's roles within a team. The operationalization can again be done using Blau's index, Teachman Entropy or the Index of Qualitative Variation. For clarity in this example the Index of Qualitative Variation (IQV) is used which returns a normalized value between 0 and 1. As described above, in a second step the analytic mappings have to be mapped to soft constraints:

$$\begin{aligned}\alpha_1^* &: \alpha_1 = 0 \\ \alpha_2^* &: \alpha_2 = 1\end{aligned}$$

where α_1^* denotes that age variation within the team should be 0 and hence minimal. In contrast to that α_2^* which represents the variety of roles should be maximal and hence equal 1.

Together with the soft constraints specification also the costs of the valued constraints have to be defined. This can happen according to the team composition strategy. Since the analytic mappings were normalized simple cost functions can be defined:

$$\begin{aligned}\phi_{\alpha_1^*}(\alpha_1^*) &: \alpha_1 \\ \phi_{\alpha_2^*}(\alpha_2^*) &: 1 - \alpha_2\end{aligned}$$

that assign 0 in case the constraint is totally satisfied and higher values up to 1 for a partial satisfaction.

Hereafter, V_{CSP_2} can be specified. An easy way to aggregate the costs is to linearly combine them. At the same time, weights can be applied to each analytic mapping in order to implement the team composition strategy as needed. For example, the costs resulting

from a not very heterogeneous distribution of roles is weighted double.

$$V_{CSP_2} = \frac{a_1 \phi_{\alpha_1^*}(\alpha_1^*) + a_2 \phi_{\alpha_2^*}(\alpha_2^*)}{a_1 + a_2} \quad (5.1)$$

where $a_1 = 1$ and $a_2 = 2$ are the weights configured for this example.

Now that the first specification step has been performed, the recommender can do a first filtering of team member candidates. Table 5.1 depicts an example pool of persons that are candidates for the teams to be composed.

MemberID	Age	BelbinRole	Tenure	Affiliation
1	24	Plant	3	CompanyA
2	38	Implementer	7	CompanyB
3	25	Specialist	4	TU Munich
4	63	Coordinator	25	ETH Zurich
5	54	Coordinator	12	ETH Zurich
6	52	Monitor Evaluator	5	ETH Zurich
7	32	Teamworker	3	CompanyA
8	38	Plant	7	TU Munich
9	42	Shaper	14	TU Munich
10	45	Specialist	20	TU Munich

Table 5.1.: Example set of team member candidates

The user of the recommender system requires a team consisting of three persons. Hence, theoretically 120 teams are possible. After solving CSP_1 the resulting set of teams is reduced to 20 teams (**Challenge 3**). Then, CSP_2 is performed that works on the prefiltered set of teams. The best teams are then evaluated according to the cost function V_{CSP_2} . To continue the above described example in Table 5.2 the evaluated and normalized analytic mappings for the top 5 teams are listed in separate columns.

MemberIDs	stddev(ages)	α_1	α_2	V_{CSP_2}
5, 9, 10	5.09	0.47	0.75	0.32
6, 9, 10	4.18	0.39	0.75	0.29
4, 5, 6	4.78	0.44	0.5	0.48
5, 6, 10	3.85	0.36	0.75	0.28
8, 9, 10	2.86	0.27	0.75	0.25

Table 5.2.: Best teams and evaluated analytic mappings

The results of the first analytic mappings regarding the homogeneity of a team with

respect to the age of its members show that the last team composed by team members 8, 9 and 10 seems to be the most suitable team. Its α_1 value is 0.27 only.

As can be seen except for the third team, all teams are very heterogeneous in terms of Belbin Roles. The index of qualitative variation accounts also for the fact that just three out of nine Belbin Roles (in the example presented in this section only seven were used) can be present in a team consisting of three people. For this reason the value 1 can never be reached with these requirements.

In the last step the costs associated with each assignment listed in Table 5.2 were computed with the aforementioned weights as denoted in Equation 5.1. The results are also presented in Table 5.2 under the column V_{CSP_2} .

Among these five teams, the last team is preferable in terms of the defined team composition strategy, because it has the least associated costs (0.25). The third team composed of members 4, 5 and 6 is ranked last, because the presence of two coordinators in the team has a high impact on the overall cost of the team ($a_2=2$).

5.2. Case-based

During the conducted interviews presented in Section 3.2.1, it was interesting to ascertain that most project managers have a clear picture on the criteria that matter when assembling their teams. Due to their experience, they know for which type of project, environment etc. which criteria matter. Having carried out projects in the past they are able to apply their experiences for similar cases to compose an appropriate team. This finding is related to the paradigm used in case-based reasoning to solve problems with the (adapted) solution of similar past projects (**Challenge 7**). For this reason, a case-based recommender approach is a promising algorithm for team recommendation.

In the following the design of such a recommender system for teams is described (for more details consult [Asikin et al., 2010]). As foundation for its development, the four main tasks of a case-based recommender described in Section 2.2.2.3 (retrieving, reusing, revising and retaining) are used as guidelines.

5.2.1. Selected Meta Model Components

As in every recommendation paradigm that makes use of the meta model, a mapping of meta model components to the aspects used in the recommender is needed for the CBR

recommender. For CBR, most of the *team member variables* (e.g. age, gender, etc.) and *team external variables* (e.g. reward system, task) selected according to the team composition strategy, can be directly used as fundamental features of *cases*. *Output/outcomes* have an important role as a part of fundamental features and the ranking mechanism of recommended past *cases*.

Dependencies, and more precisely analytic mappings, can be used to recommend a team even if the case base does not contain enough similar *cases*. Desired analytic mappings (e.g. all team members should be experts in different functional domains), that can then be used as references in the similarity assessment, can be either learned from other less similar cases contained in the case base, or can be defined according to the team composition model (see Section 5.2.2). Figure 5.1 depicts the general idea for the usage of the team composition core model and the context (cf. Section 4.7) in this team recommender system.

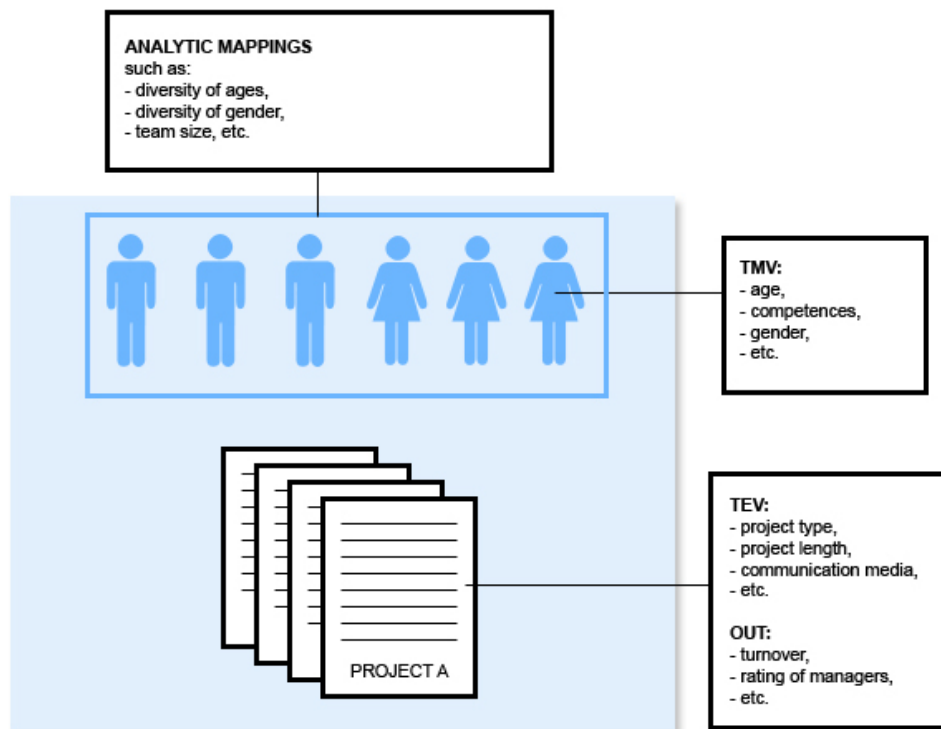


Figure 5.1.: Specification of team composition core model and context (TEV) used

Finally, *constraints*, as last possible item for a team recommendation system (the used algorithm is specified in the following sections), are especially important in the query phase of the recommendation process. The existence of constraints reduces the number of recommended teams because members or teams which do not satisfy the constraints are

eliminated directly before the *case* retrieval process (**Challenge 3**).

5.2.2. Recommendation Approach

The four phases covered in the problem solving cycle of CBR described in Section 2.2.2.3 contribute as the basis of the recommendation approach. This section proposes how the concepts of CBR can be applied to team recommendation and then describes how the meta model can be incorporated into the proposed CBR approach. Moreover, the essential CBR components are defined.

Case Representation and Definitions

The *problem* and *solution* spaces can be defined as follows: “Problem” is the description of a project (given task) consisting of (i) a specification of the project, (ii) a specification of the required members, and, (iii) constraints. All these specifications are configurable depending on the team composition model that has to be applied. For instance, if for specifying the project, its length and the presence of a reward system is indispensable then corresponding team external variables have to be defined as features (e. g. *ProjectLength-InMonths=6, RewardSystem=true*). The case-based reasoning dimension “Solution” is embodied by a team (i) whose past project specifications are similar to the specification of the current problem; (ii) whose members’ specifications are similar to the required members (positions); (iii) whose members fulfill the constraints. For (i) the same configuration aspects apply as for the above described “problem” space.

In comparison to case-based recommendation in other domains, e. g. a product recommendation with a number of static features, the *case* representation of a case-based team recommender is more complex. This is attributed to the fact that (i) there are two kinds of fundamental variables, namely the variables of individual team members and the variables of the team itself that are the result of specific analytic mappings of the individual team members (e. g. variety of gender in a team). The team member variables alone are, as mentioned earlier, not sufficient as characterizing factor for a *good* team, because the members have not only to fulfill the specifications but also to be adequate for the required context (e. g. a project). (ii) There are outcomes to be maximized and dependencies between the team member variables and the output to be considered.

To solve these issues, a CBR approach with two levels of *cases* is proposed. The **first level** or the main *case* is denoted as TEAMCASE and consists of a specific set of team members (*a team*) and another set of *contexts* (e. g. projects). Through appropriate queries

(introduced in the next section), cases are **retrieved** from the case base. **Reusing** of a team directly in a new context (project) results in the addition of the project into the set of contexts in the TEAMCASE. **Revising** the team of a TEAMCASE through addition/replacement of members or elimination of a team member yields a **new** TEAMCASE (**retaining**). Members are regarded as **second level cases** (denoted as MEMBERCASE) because a member consists of features (team member variables) and these features are required in the similarity assessment as well.

The following example depicts the approach more clearly: Suppose that a best team for a “web portal” project is already recommended by the system, but the user prefers to eliminate the “web designer” for some reason and demands new recommendations for this specific position in the current team. This elimination is enabled by the conversational component of the recommender and regarded as user feedback. The CBR system looks first in the TEAMCASE base for teams who performed tasks in a similar context. Within these teams the system then tries to find persons with properties similar to the “web designer”. In case not enough appropriate TEAMCASEs are available, the system searches in the database containing all available candidates (hence also candidates that are not in the case base) for person with similar properties.

According to the notation introduced in Section 4.7 the described *cases* are defined as follows:

$$\begin{aligned}
 TEAMCASE & : (mc, (ctx, out)) \\
 mc & = \{mc_1, mc_2, \dots, mc_N\} \text{ a set of } N \text{ MEMBERCASEs,} \\
 (ctx, out) & = \{(ctx_1, out_1), (ctx_2, out_2), \dots, (ctx_M, out_M)\} \\
 & \text{a set of tuples of M past projects (contexts) with outcomes,} \\
 mc_k & \text{ is a vector of features defined by } TMV', \\
 ctx_k & \text{ is a vector of features defined by } TEV', \\
 out_k & \text{ is a vector of features defined by } OUT'.
 \end{aligned}$$

Interaction Paradigm

From the formalized *case* representation, the problem specification (QUERY) can be derived. The query specifies how a user interacts with the system in order to retrieve the

5. Application in Recommendation Algorithms

recommendations for a desired team (**Challenge 6**).

$$\begin{aligned} \text{QUERY} & : (q, eq, con) \\ q & = \{q_1, q_2, \dots, q_N\} \text{ denotes the specification of required members} \\ eq & = \text{vector of specifications in } TEV' \\ con & = \text{vector of constraints in } CON' \\ q_k & = \text{vector of member specifications in } TMV'. \end{aligned}$$

The TMV' specification in the variable q_k corresponds to the variables mc_k defined in the *case* representation. E. g., for $TMV' = \{age, gender\}$, $q_k = ([25, 45], \{m, f\})$. The notation $[25, 45]$ is an example of an age range that specifies that the age of the member must be between 25 and 45 years. The second notation in q_k ($\{m, f\}$) says that both, *male* and *female* are accepted as gender of the team member. The specification of required team external variables eq that defines the project can be specified in the same way according to the TEV' in ctx_k . E. g., $TEV' = \{projecttype, projectlength\}$, $eq = \{\text{"news portal"}, -\}$. In this definition, the variable eq specifies the project type as "news portal" and does not limit the project length. The inclusion of constraints regarding TMV of team members in the algorithm requires the availability of members' information that is part of the constraint scope (recall Section 5.1). For example, for $CON' = \{time, location\}$ and accordingly the constraint $con = \{\text{"June-August"}, \text{"Munich"}\}$, the information on the availability of a team member candidate has to be provided. In this example, the member candidates have to provide data on the variables for *time* and *location* availability. The constraints are satisfied by using hard constraints, in contrast to other entities, where the similarity assessment is performed with the use of soft constraints.

Similarity Assessment

As mentioned earlier a similarity assessment defines the degree of similarity between past problems and the current problem in the case-based reasoning process. A solution of the problem in a more similar *case* is assumed to be a better solution than solutions from other less similar *cases*. The performance of CBR mostly depends on the suitability of an underlying similarity measurement [Avramenko and Kraslawski, 2006].

The similarity can be computed partially, thus not all features have to be considered. Generally, similarity assessments use the concept of *nearest neighbor*. Computing the distance between a target query t and a *case* c with n features can be acquired through

the L_k norm (Euclidean distance if $k = 2$):

$$L_k(t, c) = \left(\sum_{i=1}^n |t_i - c_i|^k \right)^{1/k}.$$

The L_k norm distance measure can be used, when it is known that the features of *cases* are regarded with the same importance and can therefore be computed in a uniform way. This is apparently not the case for a complex problem such as team recommendation with different kinds of features. More sophisticated similarity metrics have to be used by case-based recommenders that are based on an explicit mapping of various *case* features and the availability of specialized feature level similarity knowledge. A standard similarity function of the two *cases* can be defined as the following weighted sum metric:

$$Similarity(t, c) = \frac{\sum_{i=1..n} w_i * sim_i(t_i, c_i)}{\sum_{i=1..n} w_i}. \quad (5.2)$$

In this equation, w_i denotes the weight encoding the relative importance of a feature in the similarity assessment. The assessment involves combining the similarity measurement of individual features defined by $sim_i(t_i, c_i)$. This individual feature level similarity is calculated in various ways depending on the type of the feature, e. g. *numerical*, *hierarchical* or *categorical* (cf. Section 2.2.2.3). The value of each similarity measurement is assumed to be normalized into the range of $[0, 1]$. If the similarity is assessed through a distance measure, then consequently 0 has to be regarded as the highest similarity value across all features.

Equation 5.2 can be applied directly for measuring the similarity between two vectors of features in the spaces of TMV' and TEV' . The equation to assess the similarity between a member case mc_k and a specification vector q_k for a member with $n = |TMV'|$ can be rewritten as:

$$Similarity_{TMV}(q_k, mc_k) = \frac{\sum_{i=1..n} w_i^{TMV'} * sim_i(q_i^k, mc_i^k)}{\sum_{i=1..n} w_i^{TMV'}}. \quad (5.3)$$

Likewise, the similarity between a vector of TEV' specification eq and a project ctx_k is measured with ($n = |TEV'|$):

$$Similarity_{TEV}(eq, ctx_k) = \frac{\sum_{i=1..n} w_i^{TEV'} * sim_i(eq_i, ctx_i^k)}{\sum_{i=1..n} w_i^{TEV'}}. \quad (5.4)$$

5. Application in Recommendation Algorithms

At this point, it is possible to formalize the first global similarity measure between a *QUERY* and a *TEAMCASE*. Let *query* and *tc* be instances of *QUERY* and *TEAMCASE*, respectively and let N be the number of specified members in *q*:

$$\begin{aligned} \text{Similarity}(\text{query}, \text{tc}) = & \frac{1}{2} \left(\min_k (\text{Similarity}_{TEV}(eq, ctx_k)) \right. \\ & \left. + \frac{1}{N} \sum_{l=1}^N \text{Similarity}_{TMV}(q_l, mc_l) \right). \end{aligned} \quad (5.5)$$

where it is assumed that the set *mc* is already ordered such that the l^{th} -element of *mc* is the most similar team member to the l^{th} -element of *q*. The first term represents the similarity between the query and the most similar project within the teamcase *tc*. In other words, this is the similarity of the most similar project performed by a team represented within the teamcase *tc*. Thereby it can be assessed whether the team has experience with similar projects. The second term assesses how similar the team member mc_l are with respect to the requirements q_l in case a team specification has been defined.

If a member specification is not given by the user, the similarity is measured only with the first function and the division by 2 will naturally be neglected. Note that the *min()* function is used since the distance measure regards 0 as the maximum similarity.

Analytic Mappings as Features

As already mentioned at the beginning of this section, analytic mappings of the meta model can be used as special features. According to the team composition model chosen, reference analytic mapping values can be computed to produce an artificial *case* in the case base. If, for example, a team composition model states that high heterogeneity in terms of competencies is beneficial, a reference value of 1 using Blau's index can be chosen.

It is also possible to acquire these reference values through appropriate learning mechanisms in the case base. The second approach however, is only promising for situations in which enough similar cases are provided in the case base. Otherwise, optimal values for analytic mappings are derived from not similar cases which very likely results in erroneous assumptions (since related work in Chapter 3 has shown that teams performing in one context in general do not perform equally well in a very different context).

The utilization of the values resulting from these analytic mappings depends on the specific configuration of domain knowledge in the recommender system. To make use of

the knowledge encoded in the reference values for analytic mappings in a CBR manner, a similarity assessment can be performed: If the complete dependency consisting of analytic mappings and heuristic mapping is known (for example, it is known that the heuristic mapping states that a higher value, even over the reference value, for an analytic mapping is better), an asymmetric similarity function can be chosen in order to select good teams: higher levels for an analytic mapping have to be interpreted as favorable fact. Otherwise, a symmetric similarity function can be applied. In that case no matter whether the analytic mappings of a team are higher or lower than the reference value, the teams are estimated as comparably good. Thus, given n features on which the outputs are dependent, the similarity function in Equation 5.2 between two *TEAMCASEs* tc_a and tc_b with respect to the analytic mappings can be written as:

$$Similarity_{DEP}(tc_a, tc_b) = \frac{\sum_{i=1..n} w_i * sim_i(\alpha_i(tc_a), \alpha_i(tc_b))}{\sum_{i=1..n} w_i}$$

where α_i defines the analytic mapping function of feature i and w_i represents the configured importance weight for this function. In this regard, a team composition model can specify an ideal *TEAMCASE* which can then be compared using *Similarity_{DEP}* with real *TEAMCASEs* contained in the case base.

This approach addresses **Challenge 4** and thereby the cold start problem since by using adequate team composition models, teams can be recommended even if the case base is empty or no appropriately similar cases can be found.

Improving the Ranking Mechanism

The previously described similarity assessments between member specification and *TMV'*, project specification and *TEV'* and eventually analytic mapping reference values and analytic mappings respectively, only provide a value for the similarity. Ranking by similarity is not always satisfactory in team recommendation: The basic idea of case-based team recommendation is to suggest solutions for a given project by selecting the best teams that have performed similar tasks in the past. The most similar teams resulting from the similarity assessment however do not necessarily represent the best teams. If the difference between similarity levels of two teams to the given query is very small, it is not immediately clear which team might perform better. As an additional heuristic in this kind of situations it is favorable to take the output/outcomes into consideration. The output/outcomes are used as additional ranking criteria and to support the user's choice (**Challenge 2**).

However, a problem arises when the output is not available: The output of a team is incorporated into the *case bases* after the project is finished and analyzed. The process

would normally take a long time and most of the time the *TEAMCASEs* remain in the *case bases* without the output/outcomes information. This situation also occurs for newly formed teams. These teams are composed for comparison with the members' specification but have not participated in any projects previously.

In these situations ranking can be improved by considering the novelty of the recommendations. More novel recommendations can be achieved by improving the diversity of *cases*. Various methods to achieve diversity are already investigated and can for instance be found in [Smyth, 2007]. The diversity enables a greater variety of teams to be drawn from the set of *cases* so that they obtain more opportunities to take part in projects.

5.2.3. Algorithm Summary

The here proposed case-based team recommendation approach can be summarized as follows:

- a) Assume that TEV' , TMV' , OUT' , CON' and dependency functions (especially desired analytic mapping values) are specified according to the team composition model chosen. Additionally, features' weight in all of the spaces and a similarity threshold (denoted as $simThreshold$) are configured by the user or a domain expert.
- b) A user inputs the specification of a project, the optional members' specification, the optional constraints, and specifies $TeamsReturned$ denoting the number of team recommendations to be returned by the system.
- c) If constraints have been specified, eliminate *TEAMCASEs* and *MEMBERCASEs* from the search space which do not satisfy these (hard) constraints.
- d) Use Equation 5.5 to assess the similarity of available *TEAMCASEs* with respect to the query. The *TEAMCASEs* whose similarity value is lower than $simThreshold$ are stored into a return set $tcReturned$.
- e) If the number of teams in $tcReturned$ is larger or equal to $TeamsReturned$, rank the list with respect to the output/outcomes of the teams and return the $TeamsReturned$ best ranked teams. Otherwise, members can be retrieved and composed by using the similarity assessment between the member specification and the existing members. The members can be saved in a combination space for further combination in new created *team cases*.

The designed steps define the algorithm of the proposed recommendation approach. There is obviously a vast number of interactions and case possibilities that are mostly dependent on the specific system implementation. As described in Section 5.2.2, for instance the analytic mappings can be used as features in another similarity assessment within the

algorithm steps.

Example

In order to demonstrate the general usage of this recommender the previously described configuration and computation steps are performed with the help of an example. The

Table 5.3.: Example of features configuration for *computer science* teams.

Entity	Variable	Feature Type	Similarity Measure
TMV	age	Numeric	Absolute Distance
	gender	Nominal	Conceptual
	competencies	Set	Levenshtein
	years of experience	Numeric (asym.)	Absolute Distance
TEV	project type	Hierarchical	Levenshtein
	project length	Numeric	Absolute Distance
	communication type	Hierarchical	Levenshtein
OUT	manager rating	Numeric (asym.)	Absolute Distance

first step consists in configuring the required parameters. Table 5.3 shows the observed variables in this example and their corresponding similarity measures. In order to apply the Levenshtein similarity measure a corresponding data structure has to be provided. Figure 5.2 provides the taxonomy representing the relationships between the competencies.

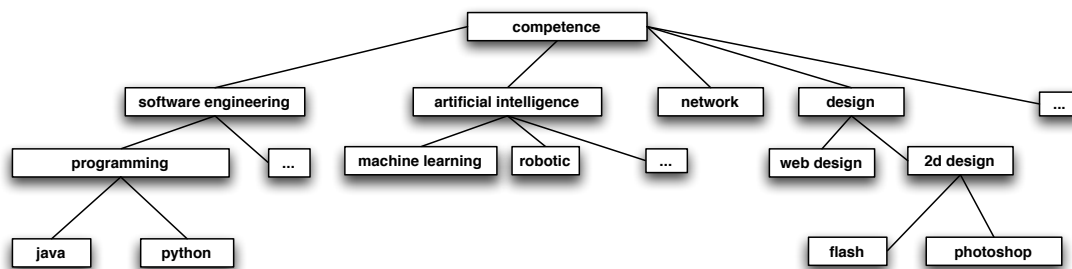


Figure 5.2.: Competence taxonomy used in the example

Figure 5.3 accordingly reflects the relationships between different problem types:

Finally, Figure 5.4 depicts how communication types can be structured.

In all three taxonomies the distances between a “parent” and a “child” is 1. Therefore,

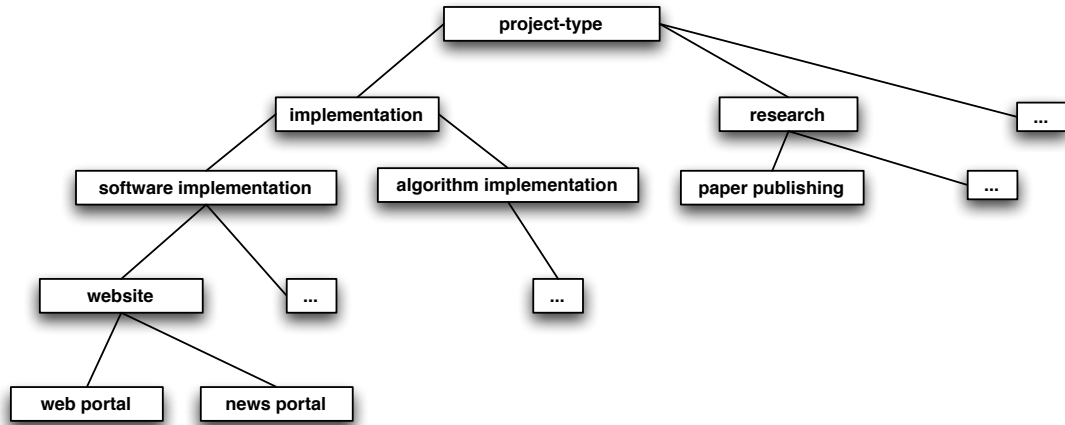


Figure 5.3.: Taxonomy of problem type for the example

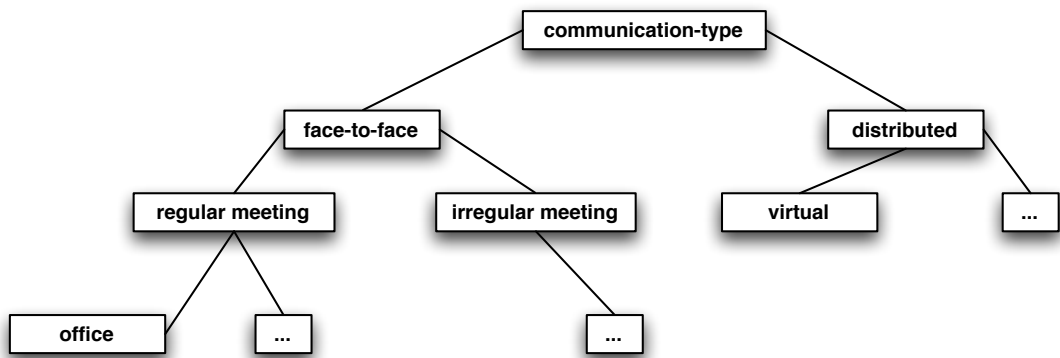


Figure 5.4.: Taxonomy of communication types in the example

for example, the distance between the competencies “competence” and “robotic” is 2. The *case* bases for this example are created adequately to help comprehension:

MEMBERCASE:

$$\begin{aligned}
 mc_1 &= (25, m, \{\text{java, photoshop}\}, 3) & mc_6 &= (33, f, \{\text{web design}\}, 6) \\
 mc_2 &= (27, f, \{\text{java, python, network}\}, 2) & mc_7 &= (40, m, \{\text{machine learning}\}, 15) \\
 mc_3 &= (34, m, \{\text{progr., network}\}, 2) & mc_8 &= (19, m, \{\text{java}\}, 1) \\
 mc_4 &= (28, m, \{\text{java, python, network}\}, 5) & mc_9 &= (29, f, \{\text{progr., web design}\}, 5) \\
 mc_5 &= (32, f, \{\text{photoshop, flash}\}, 6) & mc_{10} &= (37, m, \{\text{artificial intelligence,} \\
 & & & \text{robotic}\}, 10)
 \end{aligned}$$

As stated in the previous description of this algorithm, mc_k is the definition of a team member consisting of values of a subset of TMV. For example, mc_1 describes a team member that is 25 years old (TMV age), male (TMV gender), knows about “java” and “photoshop” (TMV competencies) and has 3 years of experience (TMV years of experience).

TEAMCASE : ($mc, (ctx, out)$)

$$\begin{aligned}
 team_1 &= (\{mc_4, mc_{10}\}, \{((\text{algorithm implementation}, 3, \text{irregular meeting}), 8.5)\}) \\
 team_2 &= (\{mc_2, mc_5, mc_1\}, \{((\text{website}, 2, \text{office}), 9.5), ((\text{website}, 5, \text{virtual}), 9.0)\}) \\
 team_3 &= (\{mc_7, mc_{10}\}, \{((\text{paper publishing}, 6, \text{virtual}), 9.0)\}) \\
 team_4 &= (\{mc_4, mc_6\}, \{((\text{web portal}, 8, \text{meeting}), 9.0), ((\text{website}, 1, \text{virtual}), 8.0)\}) \\
 team_5 &= (\{mc_8, mc_9, mc_7\}, \{((\text{web portal}, 12, \text{meeting}), 10.0), \\
 & \quad ((\text{web portal}, 6, \text{office}), 10.0)\})
 \end{aligned}$$

A teamcase $team_k$ consists as already mentioned, of a set of team members/member cases mc_k , and their past projects and outcomes. For example $team_1$ consists of the team members mc_4 and mc_{10} and in the past performed as a team a project concerning the implementation of an algorithm. The project lasted for 3 years in which they had irregular meetings to communicate. Their manager rate the success of the project with a value of 8.5 out of 10.

A query for the recommendation is for example:

QUERY : (q, eq, con)

$$\begin{aligned}
 query_1 &= (\{([25, 200], \{m, f\}, \{\text{programming}\}, [1, 100]), \\
 & \quad ([25, 200]), \{m, f\}, \{\text{webdesign}\}, [2, 100]\}), (\text{newsportal}, -, -, \emptyset)
 \end{aligned}$$

This query indicates that two team members with specific properties are required for a news-portal project. It does not matter for how long the project was and what communication type has been used. Moreover, no organizational constraints are specified in this query. For the application in the recommender system according sets of feature

value spaces for the set, hierarchical, and nominal features were created. All features were considered as equally important (weight = 1). The result of the similarity assessment between the query and *team cases* through Equation 5.5 is illustrated in Table 5.4.

Table 5.4.: Result of similarity assessment for the given query.

Team Case	Similarity	Output
<i>team</i> ₁	0.251	8.5
<i>team</i> ₂	0.184	9.5
<i>team</i> ₃	0.326	9.0
<i>team</i> ₄	0.151	9.0
<i>team</i> ₅	0.109	10.0

According to the algorithm, the set of teams being recommended consist of the three most similar teams for *TeamsReturned* = 3. Thus, *team*₂ (Sim=0.184), *team*₄ (Sim=0.151), and *team*₅ (Sim=0.109) are the provided recommendations. The ratings (outputs) can then be used in a second step to support a simple ranking mechanism which then ranks according to the output. The resulting ranking in this example is *team*₅ (Output=10), *team*₂ (Output=9.5), and *team*₄ (Output=9). Note that the number of team members is only significant to increase the likelihood to have a good matching of the QUERY with the team.

5.3. Model-based

As last recommendation algorithm a system has been developed that extensively relies on the knowledge encoded in studies on team composition. The *model-based* recommender system (cf. [Brocco and Asikin, 2011]) that is described in this section was designed for addressing an important issue related to team recommendation, namely the sparsity of data and the related ramp-up (cold start) problem. The term “model-based” has been chosen to label this recommender because team composition models are directly used to recommend teams. On the other hand “models” produced by machine learning methods are also used.

Team recommendation introduces a problem which is not common to other recommender systems (e.g. recommender systems in taste-related domains) and is related to **Challenge 2** presented in Section 4.2. To be more precise, team recommendation opens new issues regarding the output to be maximized. Is it a user’s preference as in e.g. col-

laborative filtering systems? Is it a team’s expected performance? In this regard most of the existing recommender systems were investigated in Section 3.3 rely on the explicit matching of self-defined queries (e. g. desired competencies, roles etc.) and thus maximize a user’s preference. A team recommender could on the other hand also be used to predict a team’s actual output and recommend teams accordingly. This usage of the team recommender differs from classical recommendation problems since they rarely deal with actual outcomes. Instead they deal with the output “user preference” which is often represented by an implicit or explicit rating.

The output prediction (no matter if the output is a user’s preference or an actual outcome) is challenging since team recommendation is not only heavily affected by the cold start problem but also influenced by many factors. Finding out which factors matter for which tasks and environments is very difficult. To address these problems the knowledge derived from investigations conducted in social sciences and in practice (i. e. team composition models) that provides indications on the influence factors to consider can be employed. Furthermore, also the effects of the cold start problem (**Challenge 4**) can be reduced.

For this purpose, it should be possible to instantiate several different team composition strategies in a recommender system so that a selection of team composition strategies for several environments and tasks (specific configuration of TEV) can be provided. With its use a comfortable switching of team composition strategies should be possible (**Challenge 6**). In addition to that, when data on past teamwork is provided, it should be possible to adapt or refine team composition models or potentially even create new team composition models to add to the repository.

In summary, the following aspects are not yet covered in team recommendation: (i) Considering unconventional *recommendation criteria* (i. e. instead of taste as criterion, which is usually expressed by ratings in recommender systems, using teams’ actual output/outcomes); (ii) Employing knowledge of *various* investigations and best practices on team compositions. This recommender system aims at providing contributions in this direction.

5.3.1. Selected Meta Model Components

This section describes the selection of needed meta model components for this team recommender system. Since this recommender system relies heavily on team composition models, this task is accomplished with the help of a team composition study, namely the work of Kratzer et al. introduced in Section 3.1.1, as representative for empirical studies

5. Application in Recommendation Algorithms

on team composition.

In their field investigations on innovative teams the authors focused on *cohesiveness of informal contacts* within a team and its influence on *team performance*. Here, team performance was assessed by the team members themselves (self-report). The report specified whether team members were willing to put effort into the development of new products by applying new knowledge. In addition, the generated number of ideas and other innovative aspects (e. g. creativity) were reported.

Informal contacts were categorized into *friendly communication* (informal contacts via email, phone etc.) and *friendship relations* (informal meetings for free-time activities). Firstly, they found evidence for an inversely U-shaped relationship between cohesiveness of friendly communications in the team and team performance. Next, they observed that cohesiveness of friendship relations in a team positively (and linearly) affects team performance as well. This cohesiveness is high when all members of a team meet outside the context of their job. Additionally, Kratzer et al. also used *team size* and *team tenure* as control variables.

As properties of team members are analyzed in Kratzer’s study (e. g. a list of friendship relations with other member candidates), *TMV* have to be used to model these properties. “Cohesiveness” of friendship relations can be regarded as an analytic mapping (similarly team size also belongs to this class). Hence, this meta model component is also present in mappings of team composition strategies. The teams were evaluated using the measure *team performance*. This can be regarded as variable belonging to the *OUT* set. As final component it is necessary to indicate what influence the input has on the output. E. g. it could indicate that the smaller the team, the better the team performance. In that case teams of two members (per definition an individual is not a team) would be the best team. This is done by applying heuristic mappings *H*.

In principle, these are the only ingredients needed to map team composition strategies for the model-based team recommender. They correspond exactly to the elements of the *team composition core model* elucidated in Section 4.7. In order to improve the team recommendation quality and performance of the system, also *TEV* (to improve context/task awareness) and constraints *CON* (to reduce the search space) could be optionally used. If the recommender has to be tailored for one scenario (i. e. a specific set of *TEV*), this is however not strictly necessary. The minimal set of meta model components required for mapping a team composition strategy in this recommender, thus comprises the classes *TMV*, *A* (analytic mappings), *H* (heuristic mappings) and *OUT* (as already mentioned in Section 4.7).

5.3.2. Recommendation Approach

The main idea behind model-based team recommendation is to directly use team composition models to predict a team's output/outcomes. The type of output/outcomes (e. g. team performance, costs, customers satisfaction, number of created ideas) is arbitrary as long as it is specified in the team composition model. In order to do this prediction the recommender system in the simplest case makes use of dependencies and relations that have been found in existing studies. For example, in case a study assigned a very high importance to the matching of skills within a team this is used in a corresponding team composition model in the algorithm to predict team performance. In Chapter 3 many examples for studies on team composition were presented. The more operationalized they are, the easier it is to map them to the meta model and use them for team recommendation (cf. Section 3.4). Because of the high variety of team tasks and the context of some domain, it is necessary to provide a team composition model repository where the team composition models can be stored and are ready for use.

For the algorithmic implementation of these strategies it has to be assured that all TMV, analytic mappings and heuristic mappings are operationalized in the same or similar way as in the study they were derived from. In this way, reproducibility of results generated in the study the team composition model was derived from can be ensured. When this is provided, analytic mappings can be connected in heuristic mappings as

$$\chi_k(\alpha_i(q_i, t), \dots, \alpha_j(q_j, t)) = O_k$$

By applying heuristic mappings χ_k to an arbitrary number of analytic mappings α_i the variables within the set O_k (remark that O is the Cartesian product space over OUT') can be predicted. The better the predicted output is, the higher the corresponding team is ranked. In case more output/outcomes are defined in the team composition model, an appropriate prioritization or combination of them has to be specified.

The shape of χ is determined by either the team composition strategy (e. g. a linear relationship implies a linear function), or eventually by learning algorithms in case the team composition model can be adapted when data is available. In the next paragraphs the recommendation process is illustrated in more detail.

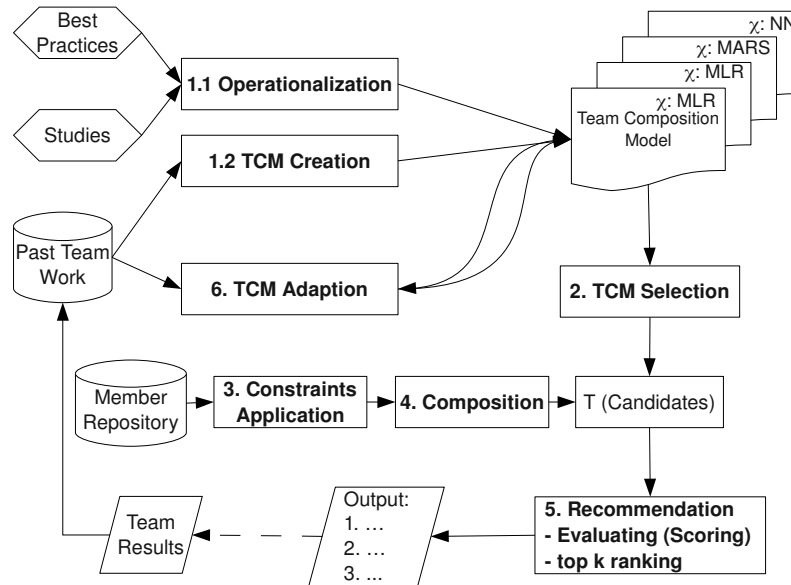


Figure 5.5.: Model-based recommendation process (Flow Chart).

Recommendation Process

The whole model-based recommendation process is depicted in Figure 5.5 and comprehends several phases: *Finding suitable models* (through either “operationalization” or “team composition model creation”), *model selection*, *constraints application*, *composition* and finally *recommendation*. When past data is available moreover existing team composition models (abbreviated with TCM) can be adapted in the *adaption* phase. These phases are described below.

Finding Suitable Models Team composition models used for recommendation can be acquired in two ways:

- 1) *Operationalization*. A model can be operationalized or constructed based on existing studies in *social sciences* (e.g. Section 4.8). A verified and accepted hypothesis implying dependencies between team performance and certain variables is already sufficient for building a team composition model. Furthermore, project managers can implement their own model according to their preferences and experiences (*best practices*). As already indicated, using these models addresses the cold start problem since no additional data about past projects in the recommendation is required.
- 2) *TCM Creation*. Given past observations, a best set of independent variables (analytic mappings) that explains the variation of a dependent variable (output/outcomes)

can be learned by certain algorithms, thereby creating a new team composition model. Examples of such algorithms include stepwise regression and MARS (which is introduced later in this section).

TCM Selection Created team composition models are stored in a repository. Depending on the given task, context, or user preferences, a suitable model is selected in this phase for recommending teams. Typically the user chooses her preferred model based on experience or the description of the models (**Challenge 6**). For instance, a study may suggest a TCM suitable for creative tasks. Apart from the model preference, this phase enables the user to enter a desired team specification q (cf. Section 4.7).

Constraints Application In addition to selecting the team composition model, the user can set constraints for the team composition before initiating the composition process. Given a pool of K team member candidates, the complete search space of a team recommendation algorithm consists of $2^K - K$ teams. Even for moderately sized values of K , it becomes impractical for the algorithm to go through the search space in reasonable time and in most cases searching the whole space is not necessary. One way for limiting the search space are constraints. The number of potential team members can be reduced before the composition is performed when constraints are available. A directly applicable hard constraint is, for example, the availability of team member candidates or the desired size of teams.

Composition Composing a team is a combinatorial problem since N out of K_{CON} individuals are selected where K_{CON} is the number of individuals that satisfy the constraints from the previous phase and N is the desired size of a team. This process is typically still challenging in spite of the reduction of the search space. Moreover, when a desired team specification q is defined, the best overall person-specification fit within a team has to be achieved by permuting team members on different member specifications.

Recommendation The recommendation process is concluded in two steps: evaluating (scoring) and top k ranking. The set T of teams composed from the previous steps has to be evaluated, in order to calculate the expected team performance. In general, the parameters encoded in the heuristic mapping are used to predict teams' outcomes according to the heuristic mapping's shape. This shape depends on which algorithm was used to train the parameters in the TCM as described in the next phase. For instance, in case multiple linear regression was used to train the model, the heuristic mapping is a

5. Application in Recommendation Algorithms

linear combination of weighted analytic mappings. Apart from multiple linear regression a variety of other learning algorithms can be used for that purpose.

However, the prediction procedure is uniform for all applied learning algorithms. Output variables always depend on the predictor variables configured with parameters \mathbf{w} . The learning algorithms attempt to predict these parameters by learning from output of past teamwork. For example, if the team performance is measured by a variable *innovative products quantity*, the actual number of resulting innovative products needs to be stored in the system after the recommended team has completed a real task. The teams are then sorted by the expected performance and the top k teams are then recommended. In case more than one outcome is expected by the user to be predicted, a linear combination of the outcomes where the user is allowed to define the weight of each output variable, can be applied.

TCM Adaption Like common recommender systems, the quality of recommendations should improve after each feedback (input) from the user. Users can evaluate (rate) teams or enter their actual outcomes after the teams have really performed the given task. This information is stored as *past teamwork* and this data is again learned, in order to improve the performance of a particular model. In the following, a number of learning algorithms that can be used to perform the model adaption phase are introduced.

Adapting models

The parameters in the models instantiated from best practices or social sciences studies that are needed by heuristic mappings to predict a team's output/outcome are tailored to the investigated case and are not always appropriate for other cases. Furthermore, often the relationships in these studies are represented with the use of standard statistical methods such as linear regression to map relationships between a set of predictor variables and one or more response variables. Consequently, the majority of team composition models only provides information on linear relationships. This may be inadequate for some cases. Studies found evidence that the results obtained via standard statistical methods are not always sufficient to define complex relationships in a team [Grzner et al., 2007]. The unsatisfying results mostly occur due to the existence of outliers and the nonlinear nature of many relationships among variables. For instance, sociological studies show that team effectiveness only increases with team size up to a certain level, and decreases for higher team sizes. A simple linear regression cannot accurately model this relationship. In consequence, it would not be possible to predict a team's output accurately for most of the team composition models used as input. Moreover, many studies from social sciences

do not provide any operationalization of these relationships at all, so that this step has to be performed before using it in the recommender system (e. g. specifying the relationship as heuristic mapping function).

For this purpose the instantiated model can be adapted or even refined when data on past teamwork can be provided. A number of learning algorithms that are able to *determine* the type of these relationships and are able to *model* these kinds of relationships can be used.

In regression tasks (such as in the team recommendation problem), a learning algorithm principally learns how an **explanatory variable** “explains” some of the variation in another **response variable**. Regression has the goal to predict the value of one or more continuous *response* variables t given the value of a D -dimensional vector \mathbf{x} . Given a training data set encompassing N observations $\{\mathbf{x}_n\}$ and corresponding response values $\{t_n\}$, where $n = 1, \dots, N$, regression predicts the value of t for a new value of \mathbf{x} . Values learned are stored in a parameter vector \mathbf{w} with the length depending on the applied learning algorithm. Three different popular learning algorithms that can be easily integrated into the recommender system were selected for adapting team composition strategies:

Linear Basis Function Models (LBFM) For modeling nonlinear relationships the standard multiple linear regression models are insufficient, because they impose significant limitations on the model. For this purpose linear basis function models (LBFM) can be utilized instead [Bishop, 2006]. LBFM consider linear combinations of fixed nonlinear functions of the input variables, given by

$$t = y(\mathbf{x}, \mathbf{w}) = w_0 + \sum_{j=1}^{m-1} w_j \phi_j(\mathbf{x})$$

where $\phi_j(\mathbf{x})$ are called *basis functions* and m is the amount of parameters in this model. Parameters \mathbf{w} can be learned by using Maximum Likelihood Estimation (MLE).

Neural Networks (NN) LBFM can model arbitrary nonlinearities occurring between response and explanatory variables. However, since the basis function is fixed, the number of basis functions needs to grow rapidly with the dimensionality D of the input space. To resolve this problem, *neural networks* allow basis functions to be adaptive. This means, it uses parametric forms for the basis functions in which the parameter values are adapted during training. Neural networks are assumption free. They can model data without a priori knowledge on the nature of the considered relationships (linear or nonlinear). Moreover, they perform well with missing or incomplete data, are robust against changes in sample size, number of variables, and against type of data distribution [Grzmar et al., 2007].

Multivariate Adaptive Regression Splines (MARS) MARS is an adaptive procedure for regression (e.g. in [Hastie et al., 2009]). It can be viewed as a generalized stepwise linear regression, i.e. it can select independent variables that significantly contribute to the regression. Moreover, as an extension of piecewise regression it is able to divide input space into regions and fit a different local function in each region. MARS uses expansions in piecewise linear basis functions of the form $(x - t)_+$ and $(t - x)_+$ where x denotes an independent variable, t has a value in the range of x and “+” means positive part¹. Example functions can be $(x - 0.18)_+$ and $(0.18 - x)_+$ or $(x - 100)_+$ and $(100 - x)_+$. These types of functions are known as *hinge* functions. Their so-called knot is at point t . In the given example at 0.18 and 100. A mirrored pair of hinge functions creates a linear piecewise function. Since a hinge functions yields 0 for a certain range of arguments ($x > t$ for $(t - x)_+$ and $x < t$ for $(x - t)_+$) they are very suitable for partitioning the data: In the ranges where hinge functions result in 0, other hinge functions can be used to fit that specific range of data, resulting into a piecewise linear function. Due to these characteristics, MARS is well suited for high dimensional problems (i.e. a large number of inputs) and large data sets.

5.3.3. Algorithm Summary

Summarizing, this approach comprises the following steps from a user perspective:

- a) Specify CON' in case organizational constraints have to be met.
- b) Select a team composition model from a model repository. Alternatively, experts can associate team composition models to tasks such that the user can indirectly choose a model by selecting a similar task.
- c) Combinatorially compose all possible teams (an integration and evaluation of CON' before this step can be very beneficial or even necessary for restricting this possibly large search space)
- d) Evaluate all heuristic mappings (and previously analytic mappings) for each team.
- e) If the team composition model is composed of more than one heuristic mapping, rank the teams according to the specified importance of the corresponding output variables predicted.

The repository of team composition models can e.g. also contain models that have been adapted or refined by using learning techniques as described in the above section “TCM Adaption”. As the parameters in that case are tailored for a specific task, the selection of such models should only be possible in an indirect manner by choosing similar tasks that were used to train these models.

¹also the notation $\max(0, t - x), \max(0, x - t)$ is popular

Figure 5.6 depicts the above listed steps in the user interface of the recommender system prototype that has been developed in the context of the KoPIWA project introduced in Section 2.1.3. The recommender uses the above-described approach to recommend teams for different tasks. Since this user interface is not part of the evaluation concept for this recommender system (cf. Section 6.3.3) its concept and look is presented in this chapter.



Figure 5.6.: Wizard interface used to configure the team recommender implemented

The first icon (Allgemeine Einstellungen), represents the first step that is responsible for setting general parameters and hard constraints CON' such as the ranges for the team size or whether the user has to be included as fixed team member (Figure 5.7).

Allgemeine Einstellungen

Aufgabenname:

Anzahl der Teams zu empfehlen:

Teamgröße: -

Ich als gesetztes Teammitglied:

Figure 5.7.: General settings for team recommender

In the second step depicted in Figure 5.8 the team composition strategy can be chosen according to the model description (Strategiemodus) or the task that has to be performed by the team (Aufgabenmodus). Thus, the user interacts with the system primarily by choosing a team composition strategy (**Challenge 6**). Users can enter as comments whether the team composition model was appropriate for past tasks they performed or not. Additionally, in case no suitable team composition model or task type can be found, the user can request for a specific new model to be included in the repository (Figure 5.8).

The third step consists in the specification of other hard constraints CON' that in the KoPIWA case were constraints on partner organizations among which team members can be selected. At the same time by selecting partner organizations the team recommender shows how many possible teams can be generated given the constraints CON' . This value can be used to predict the runtime for the generation of recommendations. It has

5. Application in Recommendation Algorithms

Wählen Sie bitte die Strategie für Team Recommendation

Aufgabenmodus ⓘ

Aufgabentyp	Beschreibung
<input type="text" value="Creativity Task"/>	<p>Quelle: Team Empfehlung beruecksichtigt Variablen, die im Zusammenhang mit Kreativitaet stehen, wie z.B. Alter und Geschlecht.</p> <p>Geeignet fuer: Eine Gruppe von Personen arbeiten zusammen, um neue Innovationen oder Ideen zu generieren.</p> <p>Kommentare:</p> <ul style="list-style-type: none">Max Mustermann - Ich habe diesen Aufgabentyp verwendet, um ein virtuelles Team zur Entwicklung einer neuen Produktidee zusammenzustellen. Wir haben zum Ideenaustausch ein Forum verwendet. Die Ergebnisse des mir empfohlenen Teams waren gut, und ich habe neue interessante Leute kennengelernt.

Haben Sie keine passende Aufgabentyp gefunden? Klicken Sie bitte hier.

Strategiemodus ⓘ

Team Empfehlungsstrategie	Beschreibung
<input type="text" value="(select model)"/>	<p>Quelle:</p> <p>Geeignet fuer:</p> <p>Kommentare:</p>

Haben Sie kein passendes Model gefunden? Klicken Sie bitte hier.

Figure 5.8.: Team composition strategy window

to be recalled that the configuration of the number of possible team members has also to be regarded as constraint belonging to the class CON' . However, for reasons of user acceptance, an additional view in the user interface was included.

The fourth view was included to enter the optional specifications q required by some of the analytic mappings. For instance, the competencies required for specific job positions can be entered. In the KoPIWA portal for example, team composition strategies using competence matching as analytic mapping, allow for the selection of competencies labeled by competence tags in contributions done in the open innovation platform (cf. Section 2.1.4, Competence Management). The screenshot depicted in Figure 5.9 shows a “matching” analytic mapping that matches ages of members within a team.

Once all the configuration steps were performed a final summary on the most important parameters is given including the team composition model used. The constraints are applied such that the amount of teams to be evaluated can be computed. Figure 5.10 shows a screenshot of this view.

Even though this chapter is concerned with the design and not the evaluation of team recommendation algorithms, it is interesting to notice from this figure that for about 1 million teams to be evaluated about 6 minutes are needed for the computation. This

Operator	NOT Variable	Variable	Wert
	<input type="checkbox"/>	Age	> 25
AND	<input type="checkbox"/>	Age	< 50

Figure 5.9.: Configuration of analytic mappings in the user interface

Zusammenfassung:

Aufgabenname: Neue Produktidee

Modelltyp: FameModel

Ich als gesetztes Mitglied: true

Anzahl der gewünschten Empfehlungen: 10

Verfügbarkeit:

Anzahl der Mitglieder: 66

Progress:

Anzahl der bearbeiteten Teams: 9703408

Die Restdauer: 4 Minute(n)



Figure 5.10.: Summary of configured parameters and progress bar for the computation of team recommendations

indicates that the approach is promising in terms of run time (**Challenge 3**) for this number of teams.

After the computation, the recommended teams are ranked and listed in a result window according to normalized output scores that are combined given the team composition model (**Challenge 2**). If a ranking by single output scores is desired, an a posteriori ordering can be achieved by clicking on the corresponding output variable on the top of the according score column without requiring a recomputation. This result window is illustrated in Figure 5.11.

Rank	Teamname	Mitglieder	Anzahl der Ideen	Umsetzbarkeit
1	Team1		1.00000	1.00000
2	Team2		0.99998	0.99998
3	Team26		0.99993	0.99990

Figure 5.11.: Window with team recommendations ranked by output scores

Example

To demonstrate the main concept of this recommender system the example strategy introduced in [Kratzer et al., 2005] and Section 3.1.1 is recalled. The aspects “cohesiveness of friendly communications”, “cohesiveness of friendship relations”, “team tenure” (control variable), “team size” (control variable) contain two different meta model components namely team member variables and analytic mappings. The other control variable “phase of innovation process” is not used here since it is a *TEV* which are omitted for reasons of clarity. Cohesiveness, in this context, is represented by Kratzer et al. as the mean of edge weights in a simple undirected weighted graph (social network) of individuals (team members). The first two aspects use this concept and only differ in the definition of edge (relation) among the nodes (individuals), i. e. friendly communication and friendship rela-

tion, respectively. Let M^1 be a $K \times K$ adjacency matrix representing the friendship relation network where K is the number of all available individuals in a recommendation system. A matrix cell $M_{i,j}^1$ denotes the edge weight between two individuals t_i and t_j , which is 1 if a friendship relation exists between these two persons and 0 otherwise (according to Kratzer’s operationalization). Since the graph is undirected, $M_{i,j}^1 = M_{j,i}^1$. The first TMV is denoted as *frel* which is a list of edge weights to other individuals in M^1 . For a team member t_i , the value is defined as a vector $m_i = (M_{i,1}^1, M_{i,2}^1, \dots, M_{i,K}^1)$ from row i in matrix M^1 . The next TMV can similarly be called *fcom*: the friendly communication network. It is represented by another equally structured matrix M^2 where $M_{i,j}^2 = 1$ means that a friendly communication relation between individuals t_i and t_j exists (0 otherwise). Team tenure *tt* is a variable containing a list of teamIds the team member was member of with the amount of years he worked in that team in the past. For instance, a team composed of two members could have worked in the past on a project concerning the development of a software and might therefore be an appropriate candidate for the new project. Finally, another TMV *person entity* can be defined which is “1” for all team member candidates and is used for the team size computation. Thus,

$$\begin{aligned}
TMV' &= \{fcom, frel, tt, person\ entity\} \\
Q &= fcom \times frel \times tt \times person\ entity \\
frel &\in \{M_i^1 = (M_{i,1}^1, M_{i,2}^1, \dots, M_{i,K}^1)\}, i \in [1, K] \\
fcom &\in \{M_i^2 = (M_{i,1}^2, M_{i,2}^2, \dots, M_{i,K}^2)\}, i \in [1, K] \\
tt &\subseteq \{(teamId, years) | teamId \in \mathbb{N}, years \in \mathbb{N}\} \\
person\ entity &\in \{1\}
\end{aligned}$$

In order to acquire the “cohesiveness of friendly communication”, “cohesiveness of friendship relation”, “team size”, “average team tenure” four analytic mappings α_1 , α_2 , α_3 and α_4 were applied respectively. As elucidated before, the means of all edge weights among team members can be used to measure the “cohesiveness of friendly communication” in a team. Let TM^1 be an adjacency matrix representation of the induced subgraph of the friendly communication network consisting of members of a team to be evaluated for the recommendation, the “cohesiveness of friendly communication” of this team is the first analytic mapping α_1 given by

$$\alpha_1(\{\}, t) = \frac{2}{|t|(|t| - 1)} \sum_{i=1}^{|t|} \sum_{j=i+1}^{|t|} TM_{i,j}^1$$

5. Application in Recommendation Algorithms

where $|t|$ is the number of members in the team. The analytic mapping α_2 can be defined in the same way by constructing a matrix TM^2 for the friendship relation network. Next, team size is given by

$$\alpha_3(\{\}, t) = \sum_{i=1}^{|t|} t_i.person\ entity$$

where t again denotes the set of team members t_i in the space Q (cf. Section 4.7). Finally, α_4 is defined as the average over the team tenures tt . To be more precise, the years members spent pairwise together in the same team are summed and averaged. The set $L_{u,v}$ for two team members u, v defined in Q is defined as:

$$L_{u,v} = \{(x, y) | x \in u_3 \wedge y \in v_3 \wedge x_1 = y_1 \wedge u \neq v\}$$

which contains pairwise tt (u_3 and v_3 , not that the index 3 represents the position of tt in the team member vectors $u, v \in Q$) tuples of team member u and v with the same *teamId* ($x_1 = y_1$). Example tuples for $L_{u,v}$ are $((2,3),(2,7)),((321,1),(321,3))$ meaning that team member u and v were both members of the teams with *teamId* “2” and “312” for 3 and 7 years and 1 and 3 years respectively.

With the help of this set the α_4 analytic mapping can be defined as

$$\alpha_4(\{\}, t) = \frac{1}{2|t|} \sum_{u \in t} \sum_{v \in t} \sum_{l \in L_{u,v}} \min(l_{1_2}, l_{2_2})$$

which sums up and averages the minimum co-occurrences of team members in a team. Notice that the sum is divided by $2|t|$ instead of only $|t|$ because of the symmetry between $L_{u,v}$ and $L_{v,u}$. The function $\min(a, b)$ returns a if it is smaller than b , otherwise b . In the above example of the two team members in team “2” and “312” α_4 would result in $\frac{\min(3,7)+\min(1,3)}{2} = 2$. The further example further illuminates the definition of the set TMV' : A team $t = \{t_1, t_2\}$ is a team composed of two team members t_1 and t_2 which have the following properties for the set TMV' $t_1 = ((0, 1, 1), (0, 1, 0), ((2, 3), (321, 1)), 1)$, $t_2 = ((1, 0, 1), (1, 0, 1), ((2, 7), (321, 3)), 1)$ denoting that the first team member worked for 3 years in team “2” and 1 year in team “321” while t_2 worked in team “2” for 7 years and team “321” for 3 years. Both team members are person entities (person entity=1). For t_1 the (egocentric) friendly communication network can be formalized as $fcom = (M_{1,1}^1, M_{1,2}^1, M_{1,3}^1) = (0, 1, 1)$: there are 3 individuals in the recommendation system, and t_1 has friendly communications with both t_2 and t_3 .

The aspects represented by the above formalizations, all influence team performance which is the only output mentioned by this team composition strategy. Thus, the out-

put/outcomes subset OUT' consists of the set *team performance* only. Aspects and output/outcomes are brought together with the help of heuristic mappings χ .

For the cohesiveness of friendly relation (α_2) an inversely U-shaped relation was hypothesized. In contrast, team size was confirmed to affect team performance negatively. Team tenure had a negative impact on team performance. The resulting operationalizations are rather intuitive: e. g. $y = b * \alpha_2(\{\}, t)^2 + c * \alpha_2(\{\}, t)$ for the first, $y = d * \alpha_3(\{\}, t)$ for the second and $y = e * \alpha_4(\{\}, t)$ for the third analytic mapping. Note that the intercepts of all mappings have been omitted since they are not needed to rank only. For convenience the vector α of analytic mappings $\alpha_i(q, t)$ is mapped to a corresponding vector \mathbf{a} that represent the computed values of each α_i mapping:

$$\mathbf{a} = (a_1, a_2, a_3, a_4)$$

By combining the above functions describing the influence of analytic mappings on team performance a prediction for the configured output “team performance” can be obtained:

$$\chi_1 = b * a_2^2 + c * a_2 + d * a_3 + e * a_4$$

b and c can be set according to Kratzer et al.’s regression analysis to -2.35 and 10.04. d and e are -0.12 and -0.45 respectively. This heuristic mapping can be used to compute a “team performance” value for each team. Since only one single output is used, no further ranking function is required to provide e. g. the top N recommendations: these N teams are the teams with the best predicted “team performance value”. No additional data except for the parameters included in the team composition study are required. The operationalization for the “cohesiveness of friendly relations” (α_1) is done in an analogous manner and results in:

$$\chi_2 = f * a_1^2 + g * a_1 + d * a_3 + e * a_4$$

Both, χ_1 and χ_2 predict the same output/outcome variable *team performance*. The study does not specify how their interplay is. Therefore they cannot be aggregated in a single heuristic mapping.

As can be seen, this mechanism is not affected by the cold start problem and is aimed at addressing the corresponding challenge listed in Section 4.2. A similar procedure for the instantiation of other team composition strategy can be performed with other best practices or studies.

Team Composition Model	Section
Pelled	Section 4.8.1
Bouncken and Winkler	Section 4.8.2
Mello and Ruckes	Section 4.8.3
Example Model 1	Section 5.1
Example Model 2	Section 5.2
Kratzer et al.	Section 5.3

Table 5.5.: Instantiated team composition models and section of occurrence

5.4. Summary

In order to show that recommendations can be generated with the help of the meta model, three algorithms were developed and presented in this chapter. For each one, a mapping of meta model components to recommendation technique constituents was proposed. By means of this mapping, examples for the functionality of each algorithm were elucidated. Table 5.5 gives an overview on the team composition models that were instantiated through the meta model in this and the previous chapter which includes four models from literature and two models that were arbitrarily created to illustrate the functionality of the algorithms. In the evaluation below two additional instantiations are used to demonstrate the applicability of the meta model approach.

The next chapter tackles the evaluation of the meta model as well as the algorithms employing it to generate team recommendations.

6. Evaluation

The previous chapters introduced the concept and design of the meta model for team recommendation and presented some team recommendation algorithms making use of it. General arguments were provided to motivate why the meta model solution is advantageous compared to tailored solutions. This chapter's goal is to evaluate in more detail the produced artifact. In this context, this chapter first elucidates, which criteria are used to appraise the effectiveness of the meta model. Then, the evaluation methodology is described which relies on a classification of evaluation approaches and methodologies of the information system sciences. The three algorithms presented in the previous chapter are then used to assess different aspects of the meta model. Finally, the design decisions and components developed for addressing the challenges listed in Section 4.2 are summarized. Given the results of each section, this chapter ends with an overall discussion of the approach including its advantages, drawbacks and limitations.

6.1. Evaluation Criteria and Approach

Evaluating models (and hence also meta models) is a demanding and complex task. This can be attributed to the fact that there are no well-known and popular metrics to show that the current model is “better” than other models. Moreover, the term “better” is very fuzzy and has to be defined before the evaluation. Certainly, one could use objective metrics such as the number of entities used to model the application domain, the number of attributes included in the entities, the number of dependencies etc. However, these measures do not have an impact on the quality of results of a team recommender system. In addition, an evaluation with respect to “other” models cannot be performed in this case since no other meta model for modeling the team composition scenario exists.

One possibility for evaluating the developed meta model is to compare it with respect to a traditional, tailored approach instead of a generic one for the team recommendation scenario. As this discussion is already conducted in Section 4.5.1 it is not used as evaluation criterion again.

6. Evaluation

A central evaluation for a meta model consists in showing that it is capable of instantiating several different models. In other words, to evaluate its *expressiveness*. In the case of the presented meta model this results in the instantiation of several different team composition models. It is clear that it cannot be demonstrated that the meta model is capable of instantiating *all* existing team composition models. However, the meta modeling can be regarded as useful in case numerous models can be instantiated.

The second important criterion for evaluating the meta model presented in this thesis refers to its *applicability*. To be more precise, it is necessary to elicit the consequences of applying the meta model in practice. On the one hand, the run time behavior of the recommender system could be affected. On the other hand the meta modeling could have implications on the recommendation quality.

To summarize, for evaluating the meta model developed in this thesis two main criteria are chosen: **expressiveness** and **applicability**. To evaluate the first criterion several instances of the meta model were created during this thesis and are summarized in a corresponding section.

The second criterion (applicability) is subdivided in the criteria **run time** and **recommendation quality** which are evaluated with the help of recommender prototypes developed for this purpose.

According to [Riege et al., 2009] the artifact produced in this work is evaluated against both evaluation approaches, *research gap* and *real world scenario*, using different evaluation methodologies. The evaluation against the research gap is performed through the following methodologies listed by Riege et al.:

- **Example** instantiations of Section 4.8
- **Development** of three **prototypes** using the meta model
- **Application** of these three prototypes
- **Simulation** of a team recommendation scenario (see constraint-based recommender evaluation)
- **Experiment** conducted within the evaluation of the model-based recommender

The latter evaluation is also considered for the evaluation against a real world scenario if the meta model has to be applied for that specific part of the open innovation process (idea generation).

The next two sections describe in more detail how these evaluations of the meta model were performed and what the main findings and results are.

6.2. Expressiveness

Expressiveness is a very important aspect to evaluate a meta model. A meta model's task is to instantiate the underlying models as possible. The common structure of the meta model can then be employed as explained in Section 4.5.1. For this purpose several different team composition models are realized with the help of the meta model and summarized in Table 5.5 in Section 5.4:

Section 4.8 demonstrated how models from literature can be instantiated by using the meta model components. Team composition models of Pelled [Pelled, 1996], Bouncken and Winkler [Bouncken and Winkler, 2008] and Mello and Ruckes [Mello and Ruckes, 2006] were instantiated step by step with meta model components as examples for team composition models from the behavioral sciences and game theoretical approaches.

Chapter 5 showed that the recommendation of teams through team composition models is not only possible in theory by presenting the usage of meta model components in three state-of-the-art recommendation algorithms. Three examples were introduced that included at least a meta model instance each.

In total eight different team composition strategies were instantiated: three from social sciences, three as examples for demonstrating the functionality of the algorithms in Chapter 5 and two are used in the next section to demonstrate the applicability of the meta model approach. In addition to these two, one algorithm presented in the next section employs a team composition strategy also used in Chapter 4 to demonstrate the functionality of the model-based recommender.

6.3. Applicability (Evaluating Algorithms)

When applying meta modeling, it has to be ensured that its application does not negatively influence the results. The results produced by a function or system using meta model instances has to be similar to the results produced without. For example, if a team recommender with a tailored model produces recommendations in a given time, a team

recommender using a meta model should behave similarly in terms of recommendations and run time. Another valid evaluation would be to apply the meta model instances to the original data of the user studies and compare whether it produces the same results. Since this data is not available a slightly different evaluation has been performed that should indicate that a team recommender using a meta model produces, from an **absolute** perspective, reasonable recommendations. This cannot compare directly to the recommendation systems applying one model only but nevertheless, in case the team recommender produces good recommendations, the meta model approach can be regarded as a valid approach in terms of result quality.

The three recommendation algorithms presented in Chapter 5 are subsequently used to show how the meta model performs in terms of applicability.

In a first evaluation the constraint-based recommender was analyzed. A team composition strategy derived from interviews was used to show the run time behavior of a team recommender system using the meta model.

Then, a case-based recommender was implemented that was evaluated in terms of recommendation quality in a qualitative manner. For this purpose a team composition strategy of an expert in creativity research is modeled using the meta model and subsequently used in the recommender system. The results are then compared to the expert's team rankings in order to show whether it is feasible and valuable to use a case-based approach for team recommendation based on the proposed meta model.

On top of the knowledge acquired from the previous recommender systems evaluations, the last evaluation in this section shows that by using team composition models derived from literature or best practices, the cold start problem can be significantly mitigated (which is one argument for using a meta model approach). Moreover, when enough data is available, the flexibility of the meta model can be utilized by changing team recommendation strategy according to the collected data. A second evaluation shows that the team recommender built on the meta model is capable of generating recommendations with a satisfying accuracy when data on past projects is provided. This evaluation is one of the few quantitative evaluations with real world data available in the team recommendation research so far.

6.3.1. Constraint-based

In Section 5.1 a recommender system was designed that solves the team recommendation problem by applying constraint reasoning. In this section the algorithm is used with a

meta model instance derived from the best practices of team composition experts. This is valuable for evaluating how a recommender system based on the meta model performs when mapping real life requirements. More details on this evaluation can be gained in [Brocco et al., 2011b].

6.3.1.1. Evaluation Approach

Runtime performance is used as criterion for evaluating this recommender system. It is important to evaluate performance issues in terms of computation time for practical use. The possibility of NP completeness [Lappas et al., 2009] of the problem of team composition makes it important to evaluate whether team recommendation can be performed in practice though the number of possible teams suggests that the computation is very time consuming.

For this purpose a constraint-based recommender system, including a graphical user interface has been developed that implements the constraint-based approach described in Section 5.1. As back-end the Choco [Team, 2010] constraint reasoner was used. The approach and the computation time of this prototype were evaluated for different team sizes and team composition models.

For this recommender the interviews presented in Section 3.2 were employed to create a team composition model for the team recommender. They represent the best practices of practitioners and hence provide a realistic scenario for this evaluation.

In terms of team member variables the project managers stated that *competencies* are the most important factor when choosing appropriate team members. Therefore, it is strictly necessary to support skill matching in the recommender. Then, variables representing the availability of team members are needed. This comprises availability in terms of time (denoted as *availability*) and *location*. As additional *TMV* very common demographic data such as the *languages* spoken, *nationality*, *gender* and *age* are added. These variables are included to allow the optional heterogeneity/homogeneity requirements expressed by some of the project managers. Language and skills moreover can be weighted by a degree of experience, i. e. whether the team members are proficient in that language or skill. As last *TMV* the overall years of *working experience* are selected.

Thus, the set $TMV' \subseteq TMV$ consists of:

$$TMV' = \{competencies, age, gender, nationality, \\ languages, workingexperience, \\ affiliation, availability, location\}$$

6. Evaluation

As mentioned in Section 3.2 *TEVs* are not actively used in the search process, but are used for refinement after the recommendation process. When people work together in a team, the circumstances under which they performed and the output are analyzed to make a final choice. *TEVs*, such as *task description*, *client type*, *project urgency* may prove useful candidates for that purpose: since they are not algorithmically used to recommend teams, and different environments only expose specific *TEVs*, this set of variables is not restricted a priori to some specific set *TEV'* for this implementation.

The same applies to output/outcomes variables that may differ according to the measured project outputs (e. g. costs, customer satisfaction).

As mentioned previously, the interviewees stated that they occasionally also make use of the heterogeneity and matching concepts to compose teams. These are typical analytic mappings. It proved to be necessary to include the “matching” mapping to compute a fitting value between required and actual skills, languages etc.. As other analytic mappings separation and variety were defined as described in Harrison’s and Klein’s work [Harrison and Klein, 2007]. These mappings are applied in order to compute tenure separation, variety of nationalities and variety of genders.

Finally, the interviewed project managers mentioned availability in specific time frames and locations as a critical factor influencing the choice of team members. People in charge of assembling teams only want team members who have the required capacity in a given time frame and are easy to contact face-to-face whenever necessary. This can be regarded as a typical constraint. They also indicated that companies opening their innovation process only choose to open it to certain partner organizations, thus constituting another typical constraint. This constraint must also be included in the recommender system. The time constraint is expressed by specifying a time frame in a date format, and the capacity in man-hours. In order to guarantee that team members are situated within a predefined spatial range, a list of cities and a maximum distance allowed can be defined. This way, project managers can specify that they require team members to be located within a certain range specified in kilometers. The affiliation constraint is addressed by matching the affiliation name. The corresponding query for the specification of constraints *CON'* hence contains a city name *c*, a range *r*, an affiliation name *a* (this could also be a list of possible affiliations; for reasons of clarity just one affiliation name is allowed), a time frame *tf* and required man-hours (*tf.hours*). According to the *TMV'* specification described in the above paragraphs, the constraints can be formalized as:

$$\begin{aligned}
CON' = \{ & distance(location, c) \leq r, \\
& affiliation = a, \\
& availability.begin \geq tf.begin, \\
& availability.end \leq tf.end, \\
& availaiblity.hours \geq tf.hours \}
\end{aligned}$$

where $distance(x,y)$ is a function computing the distance between two locations x and y in kilometers. Following the meta model formalization of Section 4.7, to complete a team recommendation system a team composition (core) model, context, constraints and an algorithm are needed. The first three points were addressed by specifying the set of variables TMV' , TEV' (which is arbitrary), OUT' (which is arbitrary), analytic mappings A and CON' . Heuristic mappings (H) could also be elicited for the matching of skills: the better the skills match, the better the output. Other strategies used by project managers also require the heuristic mapping “the more heterogeneous the age, the better the output”. A corresponding graphical user interface for specifying these requirements was developed in order to facilitate the project manager.

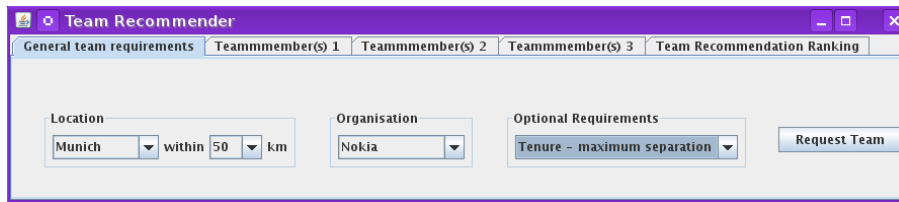
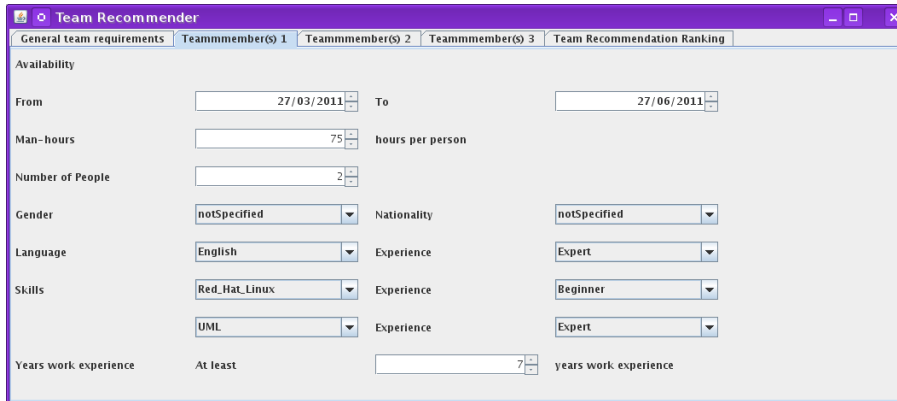


Figure 6.1.: General settings of the constraint-based team recommender

Figure 6.1 depicts the first panel of the implemented user interface containing general settings such as two of the hard constraints (affiliation/organization and location) and the choice of optional requirements (analytic mappings) to be applied. After setting these general requirements, the project manager can enter specifications for different job positions and indicate how many team members are needed for each job position. This comprises again hard constraints of CSP_1 as well as soft constraints for CSP_2 . Figure 6.2 demonstrates the representation of these aspects in the prototype. The tabs “Teammember(s)X” correspond to specifications of job positions. For each job position the number of persons needed with the specified profile can be entered.

The matching of skills is the only soft constraint which is configurable in this view. All other configuration parameters are hard constraints that can eventually be solved by CSP_1 by including corresponding constraints in the set CON' . Herewith also constraints

6. Evaluation



The screenshot shows a software window titled "Team Recommender" with a purple title bar. It has four tabs: "General team requirements", "Teammember(s) 1", "Teammember(s) 2", and "Teammember(s) 3". The "Teammember(s) 1" tab is active. The interface is a form with the following fields:

- Availability:** "From" (27/03/2011) and "To" (27/06/2011) date pickers.
- Man-hours:** A spinner box set to 75, followed by the text "hours per person".
- Number of People:** A spinner box set to 2.
- Gender:** A dropdown menu set to "notSpecified".
- Nationality:** A dropdown menu set to "notSpecified".
- Language:** A dropdown menu set to "English".
- Experience (Language):** A dropdown menu set to "Expert".
- Skills:** A dropdown menu set to "Red_Hat_Linux".
- Experience (Skills):** A dropdown menu set to "Beginner".
- Skills:** A dropdown menu set to "UML".
- Experience (Skills):** A dropdown menu set to "Expert".
- Years work experience:** The text "At least" followed by a spinner box set to 7 and the text "years work experience".

Figure 6.2.: Tab for setting the requirements for specific job positions

on the TMV involved in CSP_2 can be specified (e. g. age, gender).

Following the design of this recommender system presented in Section 5.1, CON' together with the TMV needed to solve the contained constraints, represent the hard constraints of CSP_1 . The soft constraints of CSP_2 are defined according to the analytic mappings previously described, namely: skill matching, separation of tenure, variety of nationality and variety of gender. If the user wants to consider also other matchings of individual team member variables the system includes them as hard constraints of CSP_1 . The operationalizations are the same as described in Section 5.1: for separation a normalized standard deviation was used, while for the variety of nationality and gender the index of quality variation was applied. Similarly, the costs associated are also the same as described in the example in Section 5.1. The skill matching is operationalized with Wu and Palmer's method for concept similarity [Wu and Palmer, 1994] and normalized in the range of 0 to 1. Additionally, skill matching is used to generate a list of the top n candidates for each job position in order to reduce the problem space before composing teams (**Challenge 3**).

After setting up the configuration, the recommender generated the top N recommendations for the entered query. If past data is available, TEV (describing the task and context) and OUT variables (describing the produced and generated output/outcomes of the team) are also listed in the result set so that the project manager can further investigate the results. The prototype includes just an example implementation of this feature which is illustrated in Figure 6.3.

Team	Team members	Team compa...	Teamoutput from old projects
1	P3, P137, P109,	23%	No past outputs available
2	P3, P133, P109,	20%	Output: 4 TEV: Organisation=informal Team Organisation=nominal Team Reward System=soft_rewardsOutput: 2 TEV: Organisation=international Team Organisation=self-managi...
3	P3, P103, P109,	23%	Output: 1 TEV: Organisation=national Team Organisation=manager-led Reward System=soft_rewardsOutput: 2 TEV: Organisation=national Team Organisation=self-managing Re...
4	P3, P102, P109,	23%	No past outputs available
5	P3, P80, P109,	23%	Output: 1 TEV: Organisation=national Team Organisation=manager-led Reward System=soft_rewardsOutput: 2 TEV: Organisation=national Team Organisation=self-managing Re...
6	P3, P65, P109,	23%	No past outputs available
7	P3, P15, P109,	23%	Output: 1 TEV: Organisation=national Team Organisation=manager-led Reward System=soft_rewardsOutput: 2 TEV: Organisation=national Team Organisation=self-managing Re...
8	P3, P17, P109,	20%	Output: 4 TEV: Organisation=national Team Organisation=manager-led Reward System=soft_rewardsOutput: 9 TEV: Organisation=international Team Organisation=self-managing...
9	P3, P16, P109,	20%	Output: 1 TEV: Organisation=national Team Organisation=manager-led Reward System=soft_rewardsOutput: 2 TEV: Organisation=national Team Organisation=self-managing Re...
10	P3, P10, P109,	20%	No past outputs available

Figure 6.3.: Result window with team recommendations and data on past projects of the teams if available

6.3.1.2. Evaluation Results

In order to test the run time performance of the team recommender, a test database with 500 candidates with different skills and properties was generated in order to simulate a user database of an open innovation community. In addition an exemplary skill taxonomy (50 concepts) that is used to assess the skills' similarity was developed. Afterwards, *TMV* requirements for different team sizes up to 6 team members were randomly generated. The hard constraints of the set CON' were disregarded in order to make the results better comparable and to provide an upper bound in terms of computational time. Thus, for all test runs 500 candidates were analyzed. Three different types of tests on a 2.2 GHz CPU with 2 GB RAM were performed: The first one with soft constraints tested competencies only. The second one added a heterogeneity requirement regarding nationality. The third test run comprised the two previous requirements plus the differences in tenure within a team. Hence, the amount of analytic mappings and consequently soft constraints was varied from $|\alpha^*| = 1$ to $|\alpha^*| = 3$. Figure 6.4 depicts the results of the three test runs. The horizontal axis of the diagram shows the number of required team members, the vertical one shows the number of seconds elapsed for computation. The first test run ($|\alpha^*| = 1$) is displayed by the dotted curve, the second one by the dashed curve and the last curve represents the third test run. Each test run was repeated 30 times.

The results show that the consideration of analytic mappings can be very expensive in terms of time in this approach. Nevertheless composing teams of maximum six members with only one analytical mapping ($|\alpha^*|=1$ e.g. skill matching) can be done in approximately a minute even on an average notebook. Overnight, the computation can be done for up to six team members even with two additional constraints (skill matching is always included) regarding heterogeneity or homogeneity, which should be useful in practice.

The variables influencing the run time are the amount of analytic mappings to be evaluated as well as the number of team members. When larger teams are required,

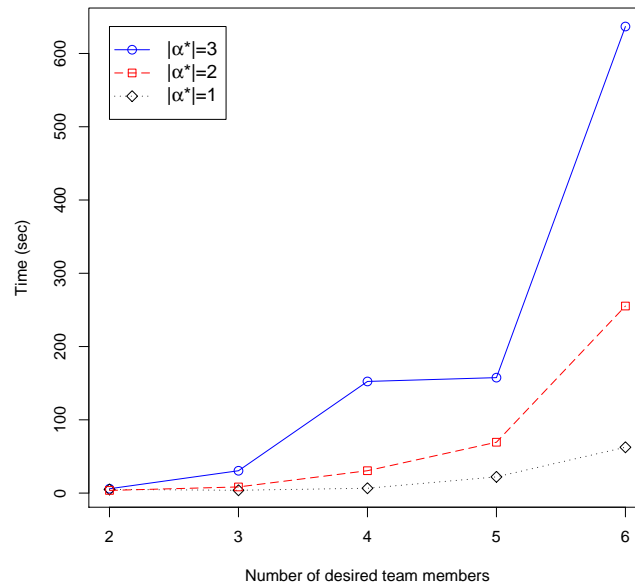


Figure 6.4.: Medians of testruns for constraint-based team recommender

the constraint-based approach run time grows exponentially with the amount of analytic mappings.

However, it has also to be considered that due to restrictions related to the constraint reasoner, the analytic mappings were encoded as a concatenation of constraints which made the computation noticeably slower. In this regard, there is certainly room for significant optimizations also with respect to the search strategy applied for constraint solving. Furthermore, hard constraints were not considered for these test runs. The constraint reasoner's strength resides exactly in evaluating these hard constraints. Thus, the performance shown in this evaluation has to be considered as upper bound only. When all these aspects are taken into account the run time can be considered as reasonable for a team consisting of six to eight team members depending on the team composition strategy, the hard constraints and the pool of team member candidates.

6.3.2. Case-based

Principally, two interesting research questions can be distinguished for a team recommender system. First, it can be analyzed whether an algorithmic approach (in general) facilitates team composition in a large set of open innovation network members making it faster and less challenging for project managers (cf. Section 4.1).

Second, the question is, whether the quality of recommendations is comparable to the quality of recommendations of a domain expert. While the first question can be answered through the application of an arbitrary recommendation algorithm (an arbitrarily defined algorithm can recommend a team among the set of *all* possible teams faster than a project manager; the quality of the recommendation may however turn out to be low), the second research question is much more challenging to tackle. Thus, the goal of the evaluation of this case-based recommender is to evaluate the quality of this recommendation approach by means of user preference rather than predicting actual team outcomes. This means that the instantiated team composition strategy contains only one variable in the set *OUT'* which is predicted, namely “user preference”.

The evaluation of the proposed approach requires to cope with some difficulties. First, a recommended team can only be verified to be suitable for a particular problem within a context if it performs the task in a real-world-scenario. The performance can then be evaluated and the outcomes be retained to the *case base* afterwards. Moreover, both the number of *cases* and new problems needed for the evaluation have to be large, in order to produce good recommendations. For this reason, this recommender system is evaluated in a different manner by comparing the ranking of teams constructed by an expert with the results of the recommender system. This evaluation approach is convenient since it focuses on the meta model rather than the case based recommendation paradigm.

6.3.2.1. Evaluation Approach

The proposed approach was evaluated in a qualitative way aided by a domain expert. Data collected during an experiment was used for this purpose. In the experiment published in [Forster et al., 2010], a number of students (2-4 persons) worked on a shared workspace with their team and tried to come up with as many qualified ideas as possible to a given topic. Idea generation fits very well into the open innovation domain, since this is an important phase in all innovation processes (i. e. also open innovation processes). The idea behind this evaluation is to create a task similar to the above-mentioned experiment and try to recommend the teams that took part in the aforementioned experiment for a new, very similar task. An expert in the field of computer supported creativity then creates another ranking list, in order to determine whether the algorithm works as the expert expects it to. In this instance, the expert estimated the best teams through experience and prior knowledge. Furthermore, he was provided with the result of the past experiment. He compared both task descriptions and considered the different output/outcomes of the teams in the previous task performed during the experiment. The expert works in the same way as a case-based recommender. Thus, this experiment can show whether this approach produces rational results.

The first step is comprised of gathering substantial variables of the teams and of the context of the experiment. This data serves as an example for a creative situation, in order to configure the team composition model. The variables which have to be considered were acquired through an interview with the domain expert. Table 6.1 presents the configuration of these critical variables. Not all variables to which the expert allocated importance had been accumulated during the data collection for the previous experiment. Nevertheless, these variables are also presented to indicate their importance in a team recommendation for this particular task (creativity task). The variables are categorized into those aspect entities which represent the components of the meta model for team recommendation. For some variables with hierarchical feature types relevant parts of taxonomies were constructed together with the domain expert, except for those which in the evaluation had only one value: for occupation for instance only the occupation “Student” was present among the participants. Also in terms of “task type” no taxonomy was required since the same task had to be performed: Finding ideas about how to effectively use student fees.

The structure for communication media constructed by the expert is shown in Figure 6.5.

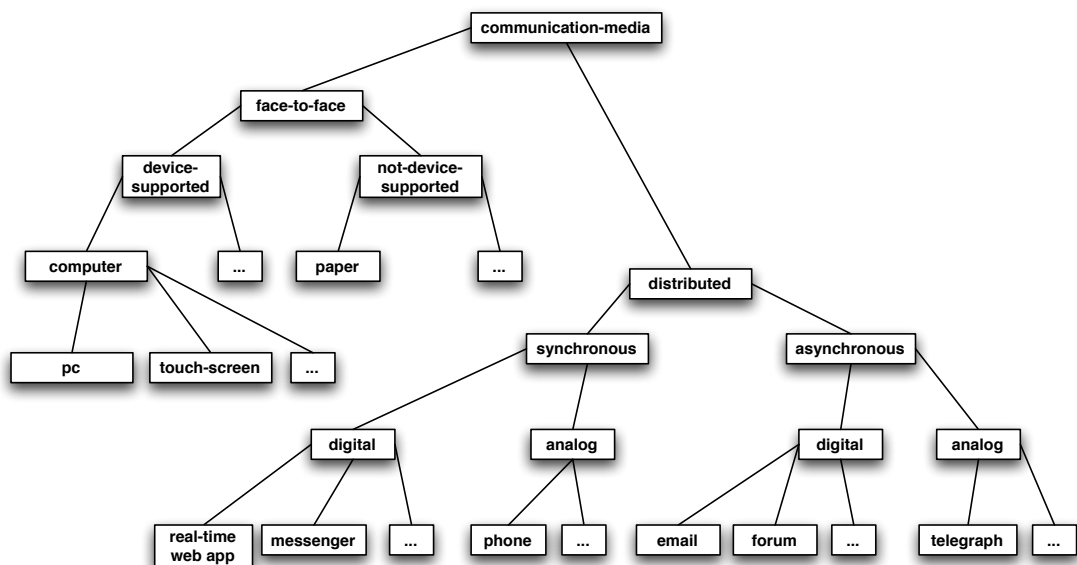


Figure 6.5.: Communication media taxonomy constructed by the expert and used in the recommendation approach

The expert also created a taxonomy characterizing different creativity techniques used. This conceptualization is represented in Figure 6.6.

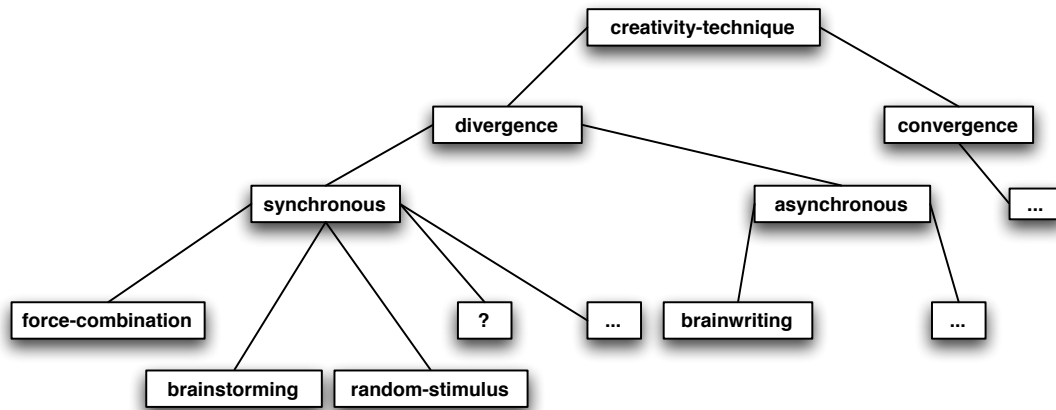


Figure 6.6.: Taxonomy of creativity techniques provided by the expert

Among the features identified by the expert, there are also many that result from various analytic mappings (cf. Table 6.1). Similar to the example in Section 5.2.3, the similarity measurement is performed to produce the recommendation.

Some variable names in Table 6.1 may not be intuitively clear. In this configuration, *creativity support system* denotes a system that supports the creative problem-solving process (cf. Section 2.1.4). The values for *fun at work* are gained from the questionnaire filled in by the participants of the experiment. This variable defines whether they enjoyed participating in the experiment. The next considerable aspect is *domination* of a team member in an experiment session. A member could, for instance, contribute to more than half of the ideas generated by the team. This variable was not included because there is no established reliable metric known to measure this aspect beforehand. *Inference* defines the level or existence of influence of team members on others during the experiment. The domain expert is convinced that this variable can show how a team member affects the performance of other team members. An appropriate formalization for this concept could also not be found, therefore, it was similarly disregarded in the calculation.

6.3.2.2. Evaluation Result

After the definition of appropriate similarity metrics the expert was asked to create a ranking of relevant teams, i. e. teams that he considers as valuable for the forthcoming, fictive task. At the same time, given the identified influence aspects, the case-based recommender produced a ranking of relevant teams by means of its newly integrated knowledge.

Table 6.1.: Features configuration for the experiment.

Entity	Variable	Feature Type
TMV	Age	Numeric
	Gender	Nominal
	Hobbies	Set
	Competencies	Not available
	Occupation	Hierarchical (Taxonomy)
	Experience with creativity technique	Nominal (boolean)
	Experience with CSS*	Nominal (boolean)
	Hours in front of PC (weekly)	Numeric
	Fun at work	Nominal
	Domination	Not available
TEV	Communication media	Hierarchical (Taxonomy)
	Task urgency	Not available
	Topic of Task	Hierarchical (Taxonomy)
	Level of Creativity in Task	Nominal
	Creativity technique	Hierarchical (Taxonomy)
Analytic Mappings	Team size	Numeric
	Gender Variety	Numeric
	Age Variety	Numeric
	Heterogeneity of Hobbies	Numeric
	Interference	Not available
	Heterogeneity of Occupation	Not available
OUT	Idea Quantity	Numeric
	Creativity of Ideas (Median)	Numeric
	Creativity of Ideas (Average)	Numeric
	Feasibility of Ideas (Median)	Numeric
	Feasibility of Ideas (Average)	Numeric

*Creativity Support System

Rank	Team	No-Output	Output
1	Group 18	3	20
2	Group 21	14	1
3	Group 3	16	5
4	Group 5	11	3
5	Group 8	21	15
6	Group 6	7	8
7	Group 1	23	11
8	Group 11	1	9
9	Group 27	5	2
10	Group 25	20	6
11	Group 4	22	7
12	Group 12	15	17
13	Group 13	6	19
14	Group 24	18	23
15	Group 7	9	12
16	Group 14	10	16
17	Group 10	12	10
18	Group 16	19	24
19	Group 9	13	4
20	Group 20	8	22
21	Group 17	4	14
22	Group 2	2	21
23	Group 19	24	13
24	Group 15	17	18

Table 6.2.: Evaluation result: comparison of ranks generated by domain-expert and by the case-based recommender

Table 6.2 presents the rankings of expert and recommender system for the fictive task.

In order to evaluate the suggested ranking mechanism of Section 5.2.2 the produced *output* was involved in the ranking procedure. The column **Rank** defines the original rank proposed by the domain expert. The ranks resulting from the case-based recommendation are presented in two columns: **No-Output** shows the rank of the group which results from the similarity assessment without the *output* information. **Output** includes the *output* features in the similarity measures to derive the rank of the group. For instance, Group 15 was ranked as 17th best team by the case-based recommender that did not use the output features. The same group was ranked as 18th by the recommender when it employed output features. The actual rank given by the expert for Group 15 was 24th.

The configured variable OUT' consists of “creativity” and “feasibility” of ideas pro-

6. Evaluation

duced by the teams which are common measures in creativity research. A special attention has to be paid at this stage: these two output/outcome variables were used in the *previous* experiment as *OUT'* and were assigned by a group of evaluators at the end of the experiment. As mentioned in **this** recommender system the *OUT'* variable consists of the output/outcome variable “user preference” which is reflected in the rankings.

From Table 6.2, it is noticeable that the past *output* plays an important role in the ranking process. By considering the *output* the recommender could reproduce the expert’s ranking in a better way. This shows that the expert might have involved them in his approach and therefore, shows the necessity of the proposed additional ranking heuristic described in Section 5.2.2. In the ranking with output only Group 18 and Group 15 have a very high difference with the actual ranking which is a good result considering that no optimizations were applied.

The evaluation shows encouraging results since the resulting ranks are comparable to the ranks of the domain expert in this specific kind of projects. The ranks shown in the column **Output** include, for example, 7 teams that are suggested by the domain expert in the first best 10 teams. It indicates that the used meta model components sufficiently cover the important variables required for a team recommendation process.

The used *case* features consist of both, properties of the projects and the individuals. Among the features used to map the expert’s team composition strategy, the aspects resulting from the analytic mappings play a major role. Already by evaluating the analytic mappings identified by the expert, good ranking predictions could be generated. This means that these analytic mappings correlate with the user’s preference. Thus, this fact can be further employed in team recommenders using the meta model.

A few significant differences in the ranks can be caused by the following problems. Firstly, all feature types had the same weight. The variables might however differ significantly in their importance for a team composition. Another problem lies in the availability of the data. The entire data came from one experiment (a single type of project) only. Better results can be obtained if the number of available projects in which the teams already participated is sufficiently large. In this case, the similarity assessment gains more information from a number of *team external variables*, namely the variables representing the experiences of the teams in various past projects. The similarity metrics used for the similarity assessment were quite simple. However, the ranking differences between expert and recommender are in the compromised range so that pursuing this approach seems to be reasonable despite the limitations of the presented evaluation (e.g. single type of problem).

6.3.3. Model-based

Among the team recommender systems found in literature (Section 3.3), only very few were evaluated in a quantitative manner. Furthermore, none of the findings resulting from the studies found in literature (Section 3.1.1) were applied a posteriori and evaluated for recommending teams. This section intends to fill this research gap by presenting a conducted experimental evaluation with the goal to determine the quality of the recommender system in a quantitative manner using the meta model in different scenarios.

6.3.3.1. Evaluation Approach

The experiment was conducted within the same platform used for the evaluation of the case-based recommender in the previous section, namely “IdeaStream”. IdeaStream (e. g. , in [Forster and Wartig, 2009]) is a creativity support system that helps to find innovative solutions for a given problem with computer supported creativity techniques. It enables a team to work collaboratively to generate ideas for a defined problem and to evaluate resulting ideas. The system is briefly introduced in Section 2.1.4. A screenshot of the front end is shown in Figure 6.7.

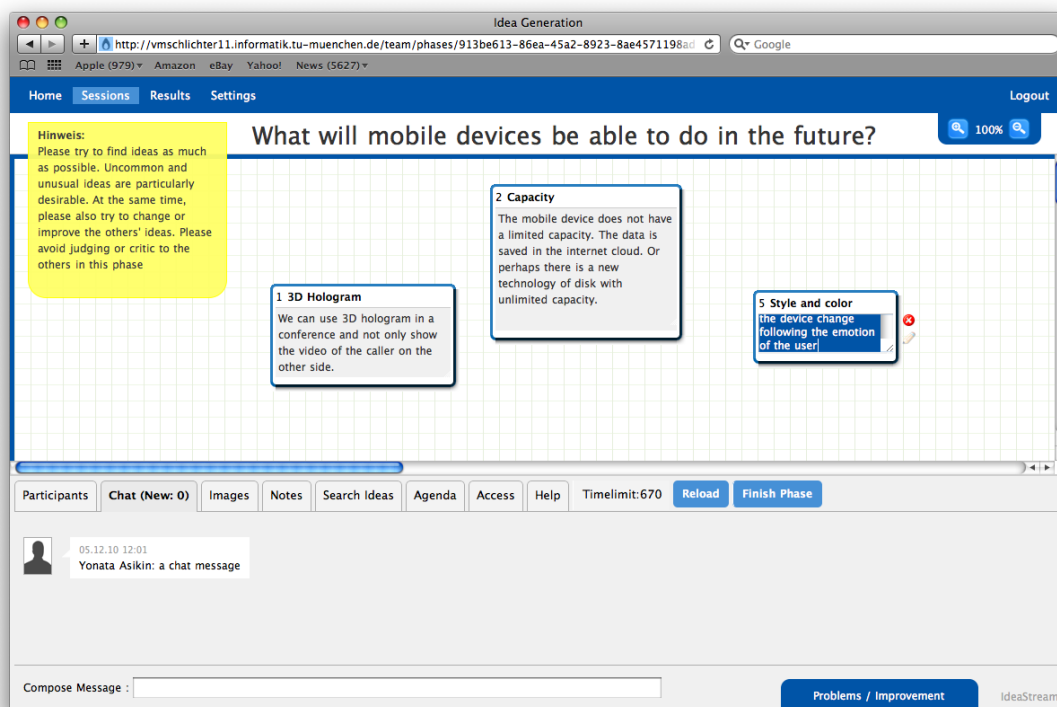


Figure 6.7.: Front end of the IdeaStream platform

6. Evaluation

This platform was used, because creativity is a very important part of innovation processes and hence also of open innovation processes. Moreover, since creative sessions usually are very short, it is possible to collect enough data for a quantitative evaluation of the recommender system.

Participants worked on this online platform in teams and brainstormed on the topic “What will mobile devices be able to do in the future?”. After signing in and completing their profile data, they were asked to collaboratively generate ideas on the given topic for 20 minutes. Then, they were to rate their ideas according to the criteria “creativity” and “feasibility” (which are measures common to creativity research) on a Likert scale. Finally, the participants had to fill in a questionnaire consisting of questions on the performed session and on other personal data needed for the application of team composition strategies.

About 40 team member variables were collected directly from member profiles and the questionnaires. These variables and the operationalizations were filtered out from about 30 social science studies on team composition. The experiment itself produced a number of innovativeness-related output variables, such as *idea quantity*, *creativity of ideas* and *feasibility of ideas*. Apart from this other output variables were measured such as the number of interactions among team members, the number of member activities/actions, or member satisfaction with the team work.

Based on the data collected during this experiment the recommendation approach was evaluated. The main focus in this evaluation was on the actual performance of the team. For this purpose the above-mentioned output variables were monitored and recorded. By reproducing the actual output/outcomes of teams as well as possible, the accuracy of the recommender system is to be determined. If the reproduction rate is acceptable it can be said that the modeling of team composition strategies through the meta model does not negatively affect the recommendation quality.

To address this goal several evaluations were performed. Evaluation 1 was to recommend the best teams that took part in this experiment based on a team composition strategy found in literature only. In this way a baseline value for the recommendation quality was provided. Then, the collected output/outcomes data from the experiment was used for the learning mechanisms in the model-based recommender in Evaluation 2. Four learning algorithms were tested for predicting the value of the selected output/outcomes variables: (1) multiple linear regression (MLR); (2) linear basis function models (LBFM); (3) neural networks (NN); and (4) MARS. For each team recommendation model, a 10-fold cross validation of each algorithm was performed and the result collected for comparison. This simulates a real scenario where the team recommender system would already have

data of past recommendations along with the corresponding team performance.

Finally, in Evaluation 3 the situation was simulated, in which only few data is available. With this evaluation method, a comparison between the usage of learning approaches and predefined team composition models in the model-based recommender is achieved. This serves to derive guidelines for deciding which approach is more promising in which scenario with respect to recommendation accuracy.

For reasons of clarity team composition models that include only one heuristic mapping χ each were used for these evaluations.

6.3.3.2. Evaluation Results

In the study, 195 participants worked in 51 distinct teams ranging from 3 to 5 members, 9 nationalities and different educational backgrounds. The participants were students or employees located in different continents, universities or companies. Figure 6.8 depicts the geographical distribution of the participants.



Figure 6.8.: Geographical distribution of experiment participants

The participants were between 17 and 40 years old and were chosen from the author's extended social network. Each team consisted of members with various social relations

6. Evaluation

(from long-time friends to completely unknown). There were teams whose members have know each other for a long time, while there were also those whose members were perfect strangers. Participants were assigned to teams according to their availability and to the maximal size for a team (which was 5).

Based on a set of selected studies, 22 analytic mappings were calculated. From this set of analytic mappings it was possible to determine new suitable team composition models for different response variables as described in Section 5.3.2.

Evaluation 1: Model from Literature

In the first evaluation an existing team composition model was instantiated to recommend teams. Since the domain of the experiment is creativity and innovative products, the strategy of Kratzer et al. formalized in Section 5.3.3 with the computed regression coefficients was used. In this case team performance is defined as *idea quantity* which hence represents the set OUT' . Nevertheless, w.l.o.g. instead of idea quantity other output measures could also be used depending on the team composition model chosen. 90% of the available data of the experiment was randomly selected from the whole data set 5 times. In each run, 10 teams out of the 90% with the most ideas were predicted. The average of resulting F-measures is 0.4 (because in the evaluation setting precision and recall are correspondingly 0.4 as well). This result indicates that a studied model can be used to recommend teams. However, its performance does not improve since no learning is applied. The role of such a model is further evaluated and compared with other models in the subsection “Cold Start”.

Evaluation 2: Computed Model

For the next evaluation, the predictor parameters of the heuristic mappings were computed by using the MARS stepwise learning. Thus, at this stage the recommender itself derives a team composition strategy (cf. Section 5.3.2) in contrast to the previous evaluation performed. This team composition strategy can then be applied for recommending teams for this case and eventually be inserted in the team composition strategy repository for reuse.

The analytic mappings that were selected by the MARS approach include α_1 : *percentage of members who work in the field the task was derived from* ([Trower and Moore, 1996] e.g. IT, computer science), α_2 : *cohesiveness of friendship* ([Kratzer et al., 2005]), α_3 : *intra-team task disagreement* ([Schönrok, 2009]), and $\alpha_4 =$ *separation of favorite color*,

which was part of the user profile and used as an experimental representant of the participants' current emotion(-diversity) [Kaya and Epps, 2004]. The predicted output variable chosen is $OUT' = \{minimum\ creativity\ value\}$ representing the minimal rating for the self-assessed creativity of an idea within a team on an ordinal scale. Table 6.3 depicts the results of a 10-fold cross-validation for computing several error measures. As threshold for the F_1 score the median of the minimum average creativity ratings was used. This means that all teams are considered as relevant that have rated their worst idea above the median value. In this table, **SSE** denotes the *sum of squared error*, **MSE** the *mean squared error*, **MAE** the *mean absolute error*, **F₁** the F-measure. Time denotes the computation time. In this experiment, algorithms such as MARS, Neural Networks, and LBFM which

Table 6.3.: Result of 10-fold cross-validation for predicting the *minimum average of creativity*.

Algorithm	SSE	MSE	MAE	Precision	Recall	F ₁	Time
MLR	1.866	0.367	0.476	0.568	0.750	0.599	3.285 s
LBFM	2.356	0.464	0.481	0.593	0.713	0.609	3.318 s
NN	2.146	0.415	0.526	0.482	0.967	0.639	4.314 s
MARS	1.535	0.299	0.416	0.613	0.800	0.663	3.188 s

are able to model nonlinear relationships performed better in terms of F-measure. MARS even outperformed MLR in each error measurement, which means that it did not over-fit the training set. This indicates that the dependency on the output is nonlinear.

This second scenario shows how learning algorithms can be used in a model to predict both team rank and actual values of team performance and thereby recommend the best teams. For a particular team output variable, the learning algorithm itself can already recommend a suitable model, i. e. which analytic mappings are suitable for predicting this output variable (e. g. in the above case *minimum average of creativity*). That means, recommendation based on any output variable including user preference can be enabled. Moreover, the performance of the model can improve by receiving more data on past, performed tasks.

Evaluation 3: Cold Start

The two previous scenarios of evaluation - also representing real recommendation scenarios - differ in the amount of test set and the computation of F-measure and are therefore incomparable. This section shows under which circumstances it is better to apply learning

algorithms and under which it is better to use existing models from the repository. As mentioned before, models exploiting learning algorithms can perform better with feedback. The significance of the relations and the accuracy of prediction will constantly improve as more team data is integrated. However, the learning algorithms are unlikely to work accurately when the amount of training data is not sufficient. To overcome this problem, the approach offers two possibilities. First, users can construct a preferred strategy by themselves which they believe to be suitable for the problem domain. Alternatively, existing studies in the same field (here, creative and innovative works) can be instantiated as a recommendation model or strategy.

A scenario representing the latter situation was established by increasing the amount of training data step by step from 2-3 observations to 12-13 and evaluating the F-Measure for each step using k -fold (where $k=51/(\#training\ data)$) cross-validations with inverted proportion of training and test set (i. e. using the smaller partition as training set). The relevant teams to compute the F-Measure were the best 10 teams (with respect to idea quantity) among a set of 30 randomly selected teams in the test set of each fold. The performances of three approaches were compared. First, stepwise regression was used over all available analytic mappings and teams with best predicted outcomes were picked. Other learning algorithms were not evaluated since the number of data points were not sufficient [Briand et al., 2004]. Next, 10 out of the teams contained in the test set were randomly picked and then recommended (random approach). The last method was by instantiating Kratzer et al.'s strategy in the recommender as used in the first evaluation "Model from Literature".

The result of this evaluation is depicted in Figure 6.9. As indicated, the results of stepwise regression improve with more data on past team work until they reach the F-measure values presented in Table 6.3 (in a classic 10-fold cross-validation). This finding is useful to enable an automatic model switching in the model-based recommender. After repeating this experiment 5 times, two sample t-test were performed. Significance between Kratzer's model and random approach for all sizes of training sets was thereby found. In contrast, the difference between Kratzer's model and stepwise regression is only significant up to 7-8 teams in the training set. For this size of the training set, p-values of the first and latter t-test are 0.009083 and 0.001861, respectively.

Discussion

The learning algorithms applied in the model-based recommender system showed satisfying performance, hence demonstrating that team recommendation can be beneficial in large collaboration networks where many possibility for team compositions are possible,

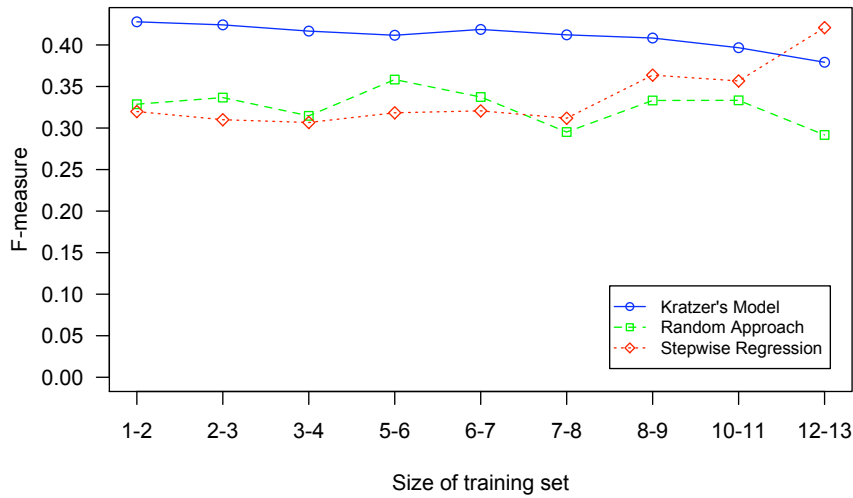


Figure 6.9.: Performance of some approaches with a small amount of training data.

especially in recurring tasks. For recurring tasks with enough learning data, the MARS technique seems to perform well for this kind of multidimensional problems, although improvements in the overall model-based algorithm (e.g. by using blending methods [Jahrer et al., 2010]) are certainly possible.

It must not be overlooked that in many cases, learning data is not available because the context in team activities often varies. In that case, the usage of existing team composition strategies (from social sciences or best practices) in the model-based recommender may be beneficial as shown by Evaluation 3 where the team composition model outperformed the learning technique in the scenario of few data on past team work.

This type of flexible usage is only possible by employing the meta model approach. The operationalization and choice of an adequate team composition model can however be challenging depending on the application domain. For that purpose detailed explanations on the team composition strategies and a large team composition strategy repository in the recommender system can be of use.

As discussed in Section 5.3, this problem does not persist when a team is selected according to someone's "taste" (rather than other output/outcome variables), as is usually the case in recommender systems, by choosing and applying favorite team composition strategies to filter out teams that correspond to the criteria defined in the team composition model. For example, if a project manager knows that he wants a heterogeneous team

6. Evaluation

in terms of tenure and with specific competencies, he may use a corresponding team composition model in the model-based recommender. No learning data is required in this case. The output to be predicted in this example would then be his preference.

This example can be extended and adapted when the project manager in question enters feedback on the produced recommendations. In this case the recommender system will learn what the specific relationship between the matching of his competencies, the heterogeneity of tenure and his preferences are. Hence, this would imply to get the favorite teams rather than the really “best” teams. Whether this type of use is desired, depends on the user of the system. This as well as the usage that considers real outcomes are both possible.

The model-based approach has the advantage of providing learning components that allow to automatically find new team composition strategies or adapt existing ones and thereby simplify the configuration process (if data is provided). For these reasons, a hybrid, switching approach that uses a team composition model up to a certain amount of observations and changes to the learned model afterwards may be appropriate for recommending teams.

6.3.4. Summary

This section presented the evaluation of three recommender systems based on the proposed meta model. The evaluation in terms of applicability has shown that the meta model can be applied to various recommender techniques to recommend teams. Depending on the chosen team composition strategy the results are satisfactory in terms of recommendation quality and run time.

The first recommender prototype used a constraint-based technique for recommending teams. This technique has to be regarded as a pure knowledge-based recommendation approach. The knowledge encoded was derived from best practices of practitioners for composing teams. In a simulation, teams with various team sizes and different numbers of analytic mappings were composed. As a result, it could be shown that in terms of run time, the approach is feasible. It has to be considered that the most popular hard constraints used in companies were not applied. Therefore, the results has to be regarded as an upper bound for the computation time.

The meta model has been shown to cover the most important aspects of team composition. This has been demonstrated in the evaluation of the case-based recommender system where an expert elicited important aspects of team composition that were then sub-

sequently used to instantiate a team composition strategy in the case-based recommender system with the help of the meta model.

The last recommender prototype was evaluated in terms of the accuracy of predicted outcomes. For this purpose an experiment with approximately 200 participants was conducted and the collected data used for an offline evaluation of the approach. The outcome shows that with enough learning data, satisfactory results can be generated by the recommender system with the help of learning techniques. However, when data on past teamwork is missing, learning techniques produce bad results. For this purpose the flexibility of the meta model approach is advantageous since team composition strategies derived from e.g. organizational sciences can be applied in order to recommend teams. This was demonstrated by using one of the team composition models introduced in this thesis which outperformed learning techniques in a ramp-up scenario.

To conclude, in terms of applicability the meta model approach produced considerable results since both, the run time as well as the quality of recommendations are within acceptable ranges for usual team sizes. It has to be considered that the evaluation in terms of quality was performed with the actual outcomes of a team rather than a user's preference. The latter is easier to predict since the data needed for prediction is easier to acquire (e.g. a user could rate according to her preference 50 teams).

6.4. Evaluating regarding Challenges

Section 4.2 elaborated several challenges that are connected with the team recommendation problem. In this thesis, these challenges have been addressed through the meta model and the developed algorithms:

Challenge 1. With the help of the meta model, depending on the algorithm used, it is possible to take into account various aspects of the team composition problem. This comprises the context of the team work, which is represented by the team external variables, as well as the influence (heuristic mappings) of team properties (TMV and analytic mappings) on different outcomes (OUT variables).

Challenge 2. By providing the OUT component in the meta model, diversification of ranking criteria has been made possible. For instance, it becomes possible to rank a team by means of a project manager's taste (by e.g. instantiating a variable "preference of project manager") or customer satisfaction. In case a team composition model provides several output/outcomes, the user can combine them at will (e.g. through desired weights and functions of the output).

Challenge 3. The team recommendation problem can be heavy in terms of computation time. For this purpose constraints can be used that significantly reduce the search space. The evaluation of the first constraint-based algorithm has shown that even without extensively using them it is possible to compose teams within acceptable time ranges. Nevertheless, depending on the use case and desired team size (e.g. 10 or more) these time ranges may not be satisfactory. In that case, analytic mappings should be avoided in favor of hard constraints.

Challenge 4. Since in some application scenarios it is required to recommend teams for very heterogeneous tasks and contexts, the benefits from learning data in terms of prediction accuracy are minimal. On the other hand, using a traditional knowledge-based team recommender with a tailored team composition strategy is similarly ineffective for situations where tasks vary frequently. For this purpose a generic team recommendation system was developed (model-based recommender) that through the meta model allows for the application of several team composition models in a transparent manner. These can moreover be adapted and refined in case data is provided. When no data is available a suitable team composition model can be chosen out of a team composition model/strategy repository.

Challenge 5. A major problem of existing team recommender systems that rely on learning data is the cold start problem. With the knowledge encoded in team composition models the user in charge to compose a team can benefit from past findings by applying an arbitrary team composition model for composing a team. Thus, an adequate team can be found with a higher probability depending on the size of the team composition strategy repository: if strategies for several use cases are provided, these can help to compose a team, especially in this kind of situations.

Challenge 6. The investigation of the team recommendation problem within companies indicated that currently the most popular approach for composing teams in companies is to use logic queries as in the database domain. This has also been used in the developed recommender systems. However, the peculiarity of the meta model approach requires also for an additional interaction with the user regarding the team composition model to be chosen. Hence, a two-step interaction is required that allows for (i) the specification of *TMV* and *CON*, and (ii) for choosing the team composition strategy. The choice of strategy is either guided by the description of the strategy or of the task, the strategies have been derived from (an example implementation was shown in Section 5.3).

Challenge 7. Through the development of the different recommendation algorithms it became evident that the team recommendation problem cannot be exclusively associated to only one problem class. Depending on the perspective and the team composition

strategy the team composition problem can be associated to several classes, even simultaneously. This was demonstrated for instance in the constraint-based recommender, where the team recommendation problem is assigned to the constraint-reasoning class for the hard constraints, as well as to the optimization class for soft constraints. Moreover, the constraint-based recommender can work on top of a database system so that it can also be seen as a problem related to this problem class. For these reasons it can not be restricted to a single problem class.

6.5. Discussion and overall Valuation

The evaluation based on the criteria *expressiveness* and *applicability* has given many insights into the validity of the chosen team recommendation approach and on team recommendation in general. In this section the approach is discussed from an overall perspective considering limitations and defining guidelines for an application of team recommendation in different scenarios.

Output/outcomes

In terms of which *team composition model* to choose, it is important for the evaluation to distinguish between the accuracy criteria related to a user's preference and the actual team output/outcomes: predicting a user's preference is generally less challenging than predicting a team's outcomes. The latter aspect can only be verified in time-consuming offline experiments (a posteriori) since the majority of project managers were interviewed in the interviews in Section 3.2 tended to be skeptical about the quality of results of an algorithmic system. This means that they are negatively biased and cannot evaluate the system from an impartial perspective. This is probably due to the fact that the recommendation process is often not easily comprehensible depending on the algorithm used. In spite of definite limitations, there is certainly also room for improvements in terms of explanations of recommendations and comprehensibility. More simple recommendation techniques (e. g. constraint- or case-based recommenders) can satisfy the user by returning expected results that however may not be the best results from an outcome perspective. For this reason attention should be paid on what output to use as recommendation criteria when creating a team recommender based on the meta model. The available team composition models are also indirectly restricted as each team composition model only considers a limited set of output/outcomes variables.

Construction of team composition models

In the course of this thesis several theories, concepts and best practices for optimal team composition have been introduced. As described in Section 3.4 many of these models are not comprehensively formalized which is inconvenient for the meta model based approach. In this case domain knowledge is required to formalize and operationalize the variables required. Though this step can be challenging it is also very beneficial in environments where tasks and contexts often vary.

Choice of team composition strategy

One drawback of this approach is the difficulty choosing the best team composition strategy/model for the problem to be solved by the team. When sufficient data is provided, this problem can be solved by machine learning algorithms in an accurate manner as explained in Section 5.3.2. When no or few data is available a good domain knowledge is required to choose the appropriate model from the vast amount of team composition models available (Chapter 3, 4 and 6 showed only a small percentage of them).

Nevertheless due to its properties the meta model approach in this matter is certainly as good as any of the current algorithmic approaches discussed in Section 3.3. Hence the shortcoming of having to choose the right mechanism to recommend the right team can be attributed to team recommendation in general: Also in the case of tailored team recommender the project manager has to choose which one is more appropriate for the given use case. Moreover, in case tailored recommender systems are used it has to be considered that several heterogeneous team recommendation systems have to be maintained (since the use of such systems is limited to only one use case) rather than one generic recommendation system with more team composition models as in the approach here proposed.

Recommendation algorithms

Three different algorithms were proposed as a representative selection of recommendation techniques: constraint-based recommendation, case-based recommendation and model-based recommendation (which can be regarded as representative for a knowledge-based and hybrid recommenders). Through the application of these algorithms it could be shown that the idea of a meta model based approach is promising, because (a) the run time has been shown to be within an acceptable time range and (b) with the right technique a good recommendation accuracy can be achieved. It has to be considered that these results have been achieved by increasing the abstraction level as required by the meta model approach.

The results also dissolve fears about the perceived danger of loss of accuracy (cf. Section 4.5.1) when applying this paradigm. Moreover, the generic approach also has the advantage that it mitigates the cold start problem, because team composition models derived from studies can be used to recommend teams when no data is available.

Regarding the choice of algorithm, a hybrid recommender making use of learning techniques and starting with a predefined or arbitrary team composition model is fully suitable for a recurrent problem with similar TEV as has been shown by the evaluation of the model-based recommender. In this case team composition models that choose actual team's output or outcomes can also be used. Whenever the TEV vary a lot, learning techniques are not adequate. In that case it makes sense to adopt knowledge-based approaches or appropriate hybrid approaches with switching technique.

The algorithms implemented to evaluate the meta model based approach are intended as proof-of-concept algorithms. As a consequence, they are not designed to be the best possible algorithms although the evaluation showed that they produced good results. This in turn means that many improvements can still be applied to the algorithms in order to achieve even better results with this generic approach.

Similarity of team external variables

The case-based approach evaluated in Section 6.3.2 relies on team external variables to produce recommendations. The similarity between two projects is computed in order to recommend adequate teams for a problem. It is challenging to define which team external variables can describe a task and environment in a characteristic manner. As a consequence, the similarity assessed may not be appropriate.

The same problem occurs when choosing a team composition model or algorithm as described above. To accomplish this task domain knowledge is needed, too. During the KoPIWA project and the interviews with practitioners presented in this thesis it has become evident that certain companies apply specific categorizations to classify past projects. These categorizations are a good starting point to assess project similarities for team recommendation, although this has not been investigated thoroughly in this thesis.

Collection of data

Apart from the obvious advantages of providing the same data structures for various different algorithms, the meta model approach has also drawbacks in this regard. To be

6. Evaluation

more precise, in order to exploit all the advantages of the meta model approach, a lot of data has to be collected since it is not always known in advance, which variables are crucial. In the experiment conducted this data was acquired implicitly and explicitly with the help of a questionnaire. It is not realistic to always expect users to fill in large questionnaires to acquire data. However, the meta model can be limited to one team composition model which results in the same amount of data required by tailored team composition systems (e. g. those listed in Section 3.3).

Another topic related to the collection of data is the privacy issue. Collecting data of team members cannot be done without the permission of the individuals. In practice (depending on the specific variables to be collected) the most important data such as skills may not be too difficult to acquire. As shown by the interviews many companies already have this kind of information. In contrast, output and performance data seem to be more sensitive. For this purpose, as proposed in the model-based recommender only numerical values for analytic mappings have to be stored for the entire team which guarantees a certain degree of anonymity making it difficult to deduce who the actual team members were, which is a positive aspect in terms of privacy.

Team recommender as support system

In spite of the quality and other advantages of the team recommendation approach shown in this thesis, it should always be kept in mind that a team recommender system is a **support system** and should be treated as one; In other words, the recommendations given by the team recommender system are only suggestions for a prefiltering and are merely to facilitate project managers' work, not make decisions for them. Hence, even if the recommender produces excellent recommendations these recommendations have to be verified since many variables are difficult to operationalize and can therefore only be assessed by a human (e. g. sympathy, charisma, etc.).

Regarding expressiveness the meta model seems to be satisfactory since it could be shown that the most important aspects can be modeled with its help. For instance, many team composition models were modeled in the course of this thesis. From this perspective it is not problematic to create a variety of team composition models that can be used to recommend teams.

Overall Valuation

The meta model approach has been demonstrated to be a feasible solution for the problem of team recommendation, although some problems could not be solved such as the issues associated with rapidly changing TEV (e. g. accurately assessing context similarity).

Overall it makes sense to use learning techniques in combination with models based on practical knowledge or behavioral sciences. The degree of formalization and operationalization of team composition models (cf. Section 3.4) varies depending on the source the model is derived from (e. g. statistical analysis vs. best practices). Thus, it may require no effort at all or be very time consuming and complex to bring team composition strategies to a processable structure needed for the meta model.

The transparent use of team composition models in fact can only be implemented if a generic approach such as the meta model approach is available which moreover allows for an application in arbitrary domains. The meta model can moreover be used as conceptual framework to identify critical aspects of team composition when designing a team recommender system and hence is advantageous also from this descriptive perspective.

In terms of run time, the approach seems to be usable in a real world scenario. This has been shown by the evaluation of the constraint-based recommender in Section 6.3.1 which returned recommendation in reasonable time without extensively applying organizational constraints. The interviews in Section 3.2 have demonstrated that these constraints drastically restrict the search space and can thus also significantly reduce computation time. For this reason the run time is certain to decrease in a real world scenario that considers these organizational aspects (e. g. availability, team size). Moreover, delegating the consideration of these aspects to the recommender system instead of manually inspecting project databases saves a lot of time and human resources.

Last, but not least it has to be noted, that team recommender systems in general rely on fresh data, i. e. data on projects, team members and outcomes has to be kept up-to-date to guarantee up-to-date recommendations. In scenarios where this cannot be guaranteed, team recommendation cannot be effective, irrespective of the type of approach used.

6. *Evaluation*

7. Conclusion

The previous chapters have described the peculiarities of team composition from several perspectives and proposed a generic approach for team composition which is based on a meta model for team composition strategies. To conclude this dissertation final remarks are presented in this chapter.

The first part of this chapter summarizes the work and methodology used in this thesis. The second part summarizes the main results and contributions derived from the previous evaluation chapter. Based on these results the last part of this chapter proposes several extensions that can be used to inspire future work in this field of research.

7.1. Summary

In this study the problem of team recommendation in large innovator networks has been addressed. The main idea towards the solution of this problem consisted in exploiting the knowledge of studies of behavioral sciences as well as best practices from experts of the domain to improve the recommendation quality.

For this purpose the domain of team composition has been investigated from a practical, theoretical and algorithmic perspective by reviewing literature in each field of research. The main finding of this investigation was that each approach, independent of its origin, is based on a team composition strategy. The difference resides in the level of formalization of the strategy and the operationalization of the therein contained aspects. For example, while algorithmic approaches are completely formalized and operationalized, certain approaches from practice rather lack formalization and are fuzzy. As a consequence, to use the latter approaches further operationalization steps are required.

After this investigation all the approaches were analyzed with respect to aspect entities of the team composition problem. As a result, several categories of aspect entities were specified including team member variables, team intermediate variables, output/outcomes, team external variables, dependencies and constraints. These classes served as foundation to develop a meta model that is used to instantiate team composition models.

7. Conclusion

The functionality of the meta model was demonstrated by first instantiating three team composition models that were previously described in the literature review and then by implementing other strategies in three different team recommender systems.

These recommender systems were evaluated against the criteria expressiveness and applicability by investigating the run time behaviour and the recommendation quality. Thereby the flexibility of the meta model with respect to its use in several algorithms was demonstrated.

Within the evaluation chapter also drawbacks and limitations of the approach were addressed and guidelines for the application of this approach specified. For instance, it was emphasized that the selection of appropriate team composition models requires domain knowledge and hence represents an obstacle for inexperienced people. However, with sufficient data, model selection can be performed by learning algorithms that can be integrated into the recommender system. In terms of run time, the approach seems to be valid if recommendations for larger teams are not required within short time frames. The variables with the greatest impact on run time are the desired size of the team and the number of analytic mappings to be evaluated.

In terms of **contributions** this work hence provided

- an interdisciplinary review on team composition methods in the social and natural sciences as well as in practice,
- a framework that allows for the analysis of the team composition problem. On the one hand this framework is a theoretical framework since it emphasizes which aspects have to be investigated to build a team recommender system. At the same time it also provides an implementation framework that can be used in various recommendation systems,
- three team recommender systems based on this framework and using appropriately adapted state-of-the-art recommendation techniques,
- in contrast to the majority of other papers on team recommendation, a quantitative evaluation on team recommendation using performance metrics that are common in recommender system research,
- an analysis that indicates which recommendation technique is appropriate under which circumstances,
- guidelines for effectively applying team recommendation to support a project manager's work.

7.2. Extensions and future work

This work provided a foundation for team recommender systems based on team composition models and strategies. The evaluation has shown that many new research questions and possibilities for improvements remain open that can inspire future work.

As has been said in Section 6.5, in case a team recommender predicts a team's output/outcome, users tend to be skeptical and do not accept the recommendations. To mitigate this reaction in future work an explanation system can be integrated to expound why certain recommendations have been generated. This may also include confidence values to signalize whether the prediction is more or less accurate.

In terms of additional experiments with team recommender systems it may be of interest to compare recommendations derived from user preference with recommendations derived from the team output/outcomes prediction. In this context, interviews with the participants should be conducted in order to understand more clearly which recommendation criterion should be used under which circumstances.

One of the main challenges of the presented approach has been identified in the selection of appropriate models for team recommendation. Concerning this aspect it may be beneficial first to recommend models that are adequate for certain scenarios. A possible implementation of this idea may be a case-based reasoning system that instead of recommending teams for a specific task and environment, recommends team composition models. These can in turn be used to recommend teams e.g. with the model-based recommender system.

In terms of improving the recommendation accuracy, the results of several team recommender systems using the same team composition strategy or team composition strategies using the same output/outcomes as ranking criteria, can be combined using blending techniques. Also other improvements by using feature augmentation (e.g. creating new team member variables by using output variables predicted by another team recommender) can enhance the recommendation quality.

Since it has been shown that a large repository of team recommendation strategies and models is beneficial, it may be of interest to engineer a community of experts sharing their team composition models on a dedicated platform. Moreover, by having such a community, feedback on team composition models can be gathered and employed to guide users when choosing team composition models for recommendation in meta model based approaches.

7. Conclusion

One of the main challenges that had to be faced in this thesis concerned the assessment of context similarity. The context (including task description, reward system type, etc.) in the meta model was specified by team external variables. To improve the recommendation quality, an analysis has to be performed that investigates which context dimensions matter for which task. Moreover, to be able to decide whether two or more tasks or projects are similar, adequate similarity measures have to be developed and tested. With the help of this categorization it can be possible to feed the learning techniques use in recommenders (e.g. the model-based team recommender) with more data and thereby improve the recommendation accuracy.

Temporal changes in context and the team itself have not been considered in this piece of research in order to better isolate core components of the team recommendation problem. This means that the current approach does not pay regard to the change of team members' properties over time. As a consequence, when such properties change, the team recommendation approaches based on this meta model have to be rerun in order to produce up-to-date recommendations. It might be of interest to find appropriate approaches that already take into account temporal influences (e.g. when a team works together for 5 years, their creativity may decrease, etc.) in the recommendation phase.

Finally, future work could analyze the suitability of this model in other application domains. For example, it may be of interest applying the team recommendation approach to the computer games domain [Brocco and Woerndl, 2011]. This domain is very convenient since the acquisition of operationalized data is easier than in the case of real teams. Games using a team recommendation system comprise multi-player games as well as games in which players have to compose teams. For instance, Sommer successfully applied the presented meta model in a model-based recommender system to recommend teams in football manager games [Sommer, 2011]. The acquisition of sufficient user data in the computer games domain is also by far less challenging since the tasks to perform in these games have a smaller dimensionality than tasks in innovator networks.

Bibliography

- [Aamodt and Plaza, 1994] Aamodt, A. and Plaza, E. (1994). Case-based reasoning: foundational issues, methodological variations, and system approaches. *AI Commun.*, 7:39–59.
- [Adomavicius et al., 2005] Adomavicius, G., Sankaranarayanan, R., Sen, S., and Tuzhilin, A. (2005). Incorporating contextual information in recommender systems using a multidimensional approach. *ACM Trans. Inf. Syst.*, 23(1):103–145.
- [Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *Knowledge and Data Engineering, IEEE Transactions on*, 17(6):734 – 749.
- [Adomavicius and Tuzhilin, 2011] Adomavicius, G. and Tuzhilin, A. (2011). Context-aware recommender systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, pages 217–253. Springer US.
- [Aguzzoli et al., 2002] Aguzzoli, S., Avesani, P., and Massa, P. (2002). Collaborative case-based recommender systems. In *ECCBR '02: Proceedings of the 6th European Conference on Advances in Case-Based Reasoning*, pages 460–474, London, UK. Springer-Verlag.
- [Amabile et al., 1996] Amabile, T. M., Conti, R., Coon, H., Lazenby, J., and Herron, M. (1996). Assessing the work environment for creativity. *The Academy of Management Journal*, 39(5):1154–1184.
- [Ancona and Caldwell, 1989] Ancona, D. G. and Caldwell, D. E. (1989). Demography and design : predictors of new product team performance. Working papers 3078-89., Massachusetts Institute of Technology (MIT), Sloan School of Management.
- [Andergassen-Sölva, 2010] Andergassen-Sölva, E. (2010). Constraint-based team recommendation. Bachelor thesis, Technische Universität München.
- [Asikin et al., 2010] Asikin, Y. A., Brocco, M., and Woerndl, W. (2010). Case-based

- team recommendation. In *Proceedings of the Second international conference on Social informatics*, SocInfo'10, pages 1–18, Berlin, Heidelberg. Springer-Verlag.
- [Attwell, 2007] Attwell, G., editor (2007). *Searching, Lurking, and the Zone of Proximal Development. e-learning in small and medium enterprises in Europe*, volume 5 of *Series navreme*. Pontybridd/Bremen.
- [Avramenko and Kraslawski, 2006] Avramenko, Y. and Kraslawski, A. (2006). Similarity concept for case-based design in process engineering. *Computers & Chemical Engineering*, 30(3):548–557.
- [Bacharach, 1989] Bacharach, S. B. (1989). Organizational theories: Some criteria for evaluation. *Academy of Management Review*, 14(4):496–515.
- [Bartak, 2002] Bartak, R. (2002). Modelling soft constraints: A survey. *Neural Network World*, 12(5):421–431.
- [Belbin, 2010] Belbin, R. M. (2010). *Management Teams. Why They Succeed Or Fail*. Butterworth-Heinemann, Oxford, third edition.
- [Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*, chapter 3, pages 137–177. Springer-Verlag New York, Inc.
- [Bortz and Döring, 2006] Bortz, J. and Döring, N. (2006). *Forschungsmethoden und Evaluation für Human- und Sozialwissenschaftler*. Springer-Lehrbuch. Springer.
- [Bouncken and Winkler, 2008] Bouncken, R. and Winkler, V. (2008). National and cultural diversity in global innovation teams. creativity and innovation as a function of cultural team composition. In *Management of Engineering Technology, 2008. PICMET 2008. Portland International Conference on*, pages 1764 –1772.
- [Bradner et al., 2005] Bradner, E., Mark, G., and Hertel, T. (2005). Team size and technology fit: participation, awareness, and rapport in distributed teams. *Professional Communication, IEEE Transactions on*, 48(1):68 – 77.
- [Breese et al., 1998] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 43–52, San Francisco. Morgan Kaufmann.
- [Briand et al., 2004] Briand, L. C., Freimut, B., and Vollei, F. (2004). Using multiple

- adaptive regression splines to support decision making in code inspections. *J. Syst. Softw.*, 73:205–217.
- [Briggs-Myers, 1980] Briggs-Myers, I. (1980). *Introduction to Type*. Consulting Psychologists Press, Palo Alto, CA.
- [Brocco, 2008] Brocco, M. (2008). *Ein kontextsensitives Recommender System für mobile Anwendungen in Fahrzeugen*. Vdm Verlag Dr. Müller.
- [Brocco and Asikin, 2011] Brocco, M. and Asikin, Y. A. (2011). Employing team composition strategies for recommending teams. In *IEEE ICDMW Proceedings, Workshop on Mining Communities and People Recommenders*, COMMPER'11. IEEE.
- [Brocco et al., 2011a] Brocco, M., Forster, F., and Frieß, M. R. (2011a). 360° open creativity support. *Journal of Universal Computer Science*, 17(12):1673–1689. http://www.jucs.org/jucs_17_12/360_degree_open_creativity.
- [Brocco and Groh, 2009] Brocco, M. and Groh, G. (2009). Team recommendation in open innovation networks. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 365–368, New York, NY, USA. ACM.
- [Brocco and Groh, 2010] Brocco, M. and Groh, G. (2010). Team Recommender für Open Innovation Netzwerke. In *Open Innovation erfolgreich umsetzen - Unternehmensstrategien und Kompetenzmanagement*.
- [Brocco et al., 2010a] Brocco, M., Groh, G., and Forster, F. (2010a). A meta model for team recommendations. In Bolc, L., Makowski, M., and Wierzbicki, A., editors, *Social Informatics*, volume 6430 of *Lecture Notes in Computer Science*, pages 35–50. Springer Berlin / Heidelberg.
- [Brocco et al., 2010b] Brocco, M., Groh, G., and Kern, C. (2010b). On the influence of social factors on team recommendations. In *Proc. Second International Workshop on Modeling, Managing and Mining of Evolving Social Networks (M3SN), 26th IEEE International Conference on Data Engineering*.
- [Brocco et al., 2011b] Brocco, M., Hauptmann, C., and Andergassen-Soelva, E. (2011b). Recommender system augmentation of hr databases for team recommendation. *22nd Conference on Database and Expert Systems Applications, International Workshop on Recommender Systems meet Databases*, pages 554–558.
- [Brocco and Woerndl, 2011] Brocco, M. and Woerndl, W. (2011). Location-based team recommendation in computer gaming scenarios. In *Proceedings of the 2nd ACM*

- SIGSPATIAL International Workshop on Querying and Mining Uncertain Spatio-Temporal Data*, QUeST '11, pages 21–28, New York, NY, USA. ACM.
- [Burke, 2000] Burke, R. (2000). Knowledge-based recommender systems. In *Encyclopedia of Library and Information Systems*, volume 69.
- [Burke, 2002] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.
- [Burke, 2007] Burke, R. (2007). *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, chapter Hybrid Web Recommender Systems, pages 377–408. Springer Berlin / Heidelberg.
- [Byrne, 1997] Byrne, D. (1997). An overview (and underview) of research and theory within the attraction paradigm. *Journal of Social and Personal Relationships*, 14(3):417–431.
- [Cai and Kock, 2009] Cai, G. and Kock, N. (2009). An evolutionary game theoretic perspective on e-collaboration: The collaboration effort and media relativeness. *European Journal of Operational Research*, 194(3):821–833.
- [Chen, 2005] Chen, A. (2005). *Location- and Context-Awareness*, volume 3479 of *Lecture Notes in Computer Science*, chapter Context-Aware Collaborative Filtering System: Predicting the User’s Preference in the Ubiquitous Computing Environment, pages 244–253. Springer Berlin / Heidelberg.
- [Chen and Lin, 2004] Chen, S.-J. and Lin, L. (2004). Modeling team member characteristics for the formation of a multifunctional team in concurrent engineering. *Engineering Management, IEEE Transactions on*, 51(2):111–124.
- [Chesbrough, 2003] Chesbrough, H. (2003). *Open Innovation: The new Imperative for creating and profiting from technology*. Harvard Business School Press, Massachusetts.
- [Chesbrough et al., 2006] Chesbrough, H. W., Vanhaverbeke, W., and West, J. (2006). *Open innovation: researching a new paradigm*. Oxford University Press, illustrated, reprint edition.
- [Colucci et al., 2005] Colucci, S., Noia, T. D., Sciascio, E. D., Donini, F. M., Piscitelli, G., and Coppi, S. (2005). Knowledge based approach to semantic composition of teams in an organization. In *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*, pages 1314–1319, New York, NY, USA. ACM.

- [Consoli and Patrucco, 2007] Consoli, D. and Patrucco, P. P. (2007). Technological platforms and the governance of knowledge: Evidence from Italy and the UK. Dipartimento di economia “S. Cogneetti de Martiis”LEI & BRICK - laboratorio di economia dell’innovazione “Franco Momigliano”, Bureau of Research in Innovation, Complexity and Knowledge, Collegio Carlo Alberto. WP series, University of Turin.
- [Convertino et al., 2008] Convertino, G., Billman, D., Pirolli, P., Massar, J., and Shrager, J. (2008). The cache study: Group effects in computer-supported collaborative analysis. *Computer Supported Cooperative Work (CSCW)*, 17:353–393. 10.1007/s10606-008-9080-9.
- [Copeland and Griggs, 1986] Copeland, L. and Griggs, L. (1986). *Going International. How to Make Friends and Deal Effectively in the Global Marketplace*. Plume, New York.
- [Cox and Blake, 1991] Cox, T. H. and Blake, S. (1991). Managing cultural diversity: Implications for organizational competitiveness. *The Executive*, 5(3):45–56.
- [Crowston et al., 2004] Crowston, K., Annabi, H., Howison, J., and Masango, C. (2004). Effective work practices for software engineering: free/libre open source software development. In *WISER '04: Proceedings of the 2004 ACM workshop on Interdisciplinary software engineering research*, pages 18–26, New York, NY, USA. ACM.
- [Cunningham, 2009] Cunningham, P. (2009). A taxonomy of similarity mechanisms for case-based reasoning. *IEEE Transactions on Knowledge and Data Engineering*, 21:1532–1543.
- [De Dreu and Weingart, 2003] De Dreu, C. K. W. and Weingart, L. R. (2003). Task versus relationship conflict, team performance, and team member satisfaction: A meta-analysis. *Journal of Applied Psychology*, 88(4):741–749.
- [Dempster et al., 1977] Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.
- [Eisenhardt, 1989] Eisenhardt, K. M. (1989). Building Theories from Case Study Research. *The Academy of Management Review*, 14(4):532–550.
- [Fagerberg et al., 2006] Fagerberg, J., Mowery, D. C., and Nelson, R. R. (2006). *The Oxford handbook of innovation*. Oxford University Press, illustrated, reprint edition.
- [Felfernig and Burke, 2008] Felfernig, A. and Burke, R. (2008). Constraint-based recom-

- mender systems: technologies and research issues. In *ICEC '08: Proceedings of the 10th international conference on Electronic commerce*, pages 1–10, New York, NY, USA. ACM.
- [Fjermestad and Hiltz, 2000] Fjermestad, J. and Hiltz, S. R. (2000). Group support systems: A descriptive evaluation of case and field studies. *J. Manage. Inf. Syst.*, 17:115–159.
- [Fokusgruppe Management offener Innovationsprozesse, 2010] Fokusgruppe Management offener Innovationsprozesse (Last Access 27/04/2010). Fokusgruppe Management offener Innovationsprozesse. <http://www.management-offener-innovationsprozesse.de>.
- [Forster, 2008] Forster, F. (2008). Distributed creative problem solving over the web. In *ICIW '08: Proceedings of the 2008 Third International Conference on Internet and Web Applications and Services*, pages 283–288, Washington, DC, USA. IEEE Computer Society.
- [Forster and Brocco, 2008] Forster, F. and Brocco, M. (2008). *Knowledge-Based Intelligent Information and Engineering Systems*, volume 5178 of *Lecture Notes in Computer Science*, chapter Understanding Creativity-Technique Based Problem Solving Processes, pages 806–813. Springer Berlin Heidelberg.
- [Forster et al., 2010] Forster, F., Frieß, M. R., Brocco, M., and Groh, G. (2010). On the impact of chat communication on computer-supported idea generation processes. In *Proc. of the First International Conference on Computational Creativity (ICCC X), Lisbon, Portugal, Jan. 2010*.
- [Forster and Wartig, 2009] Forster, F. and Wartig, H. (2009). Creativity techniques for collocated teams using a web-based virtual whiteboard. In *ICIW '09: Proceedings of the 2009 Fourth International Conference on Internet and Web Applications and Services*, pages 7–11, Washington, DC, USA. IEEE Computer Society.
- [Frieß et al., 2012] Frieß, M. R., Groh, G., Reinhardt, M., Forster, F., and Schlichter, J. (2012). Context-aware creativity support for corporate open innovation. *International Journal of Knowledge-Based Organizations*, 2(1):38–55.
- [Gonçalves et al., 2006] Gonçalves, B., Lopes, A., Esteves, S., and Botelho, L. (2006). Cognitive agents based simulation for decisions regarding human team composition. In *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 65–72, New York, NY, USA. ACM.

- [Granovetter, 1973] Granovetter, M. (1973). The Strength of Weak Ties. *The American Journal of Sociology*, 78(6):1360–1380.
- [Greenberg and Gutwin, 1996] Greenberg, S. and Gutwin, C. (1996). Awareness through fisheye views in relaxed-wysiwis groupware. In *Proceedings of Graphics Interface*, pages 28–38. Morgan-Kaufmann.
- [Groh et al., 2010] Groh, G., Brocco, M., and Asikin, Y. A. (2010). *Competence Management for Open Innovation*, volume 30 of *Schriften zu Kooperations- und Mediensystemen*, chapter Contribution Awareness and Fame in Open Innovation Networks, pages 115–138. EUL Verlag.
- [Groh and Ehmig, 2007] Groh, G. and Ehmig, C. (2007). Recommendations in taste related domains: collaborative filtering vs. social filtering. In *GROUP '07: Proceedings of the 2007 international ACM conference on Supporting group work*, pages 127–136, New York, NY, USA. ACM.
- [Grznar et al., 2007] Grznar, J., Prasad, S., and Tata, J. (2007). Neural networks and organizational systems: Modeling non-linear relationships. *European Journal of Operational Research*, 181(2):939 – 955.
- [Hackman, 1987] Hackman, R. J. (1987). *Handbook of organizational behaviour*, chapter The design of work teams. NJ: Prentice Hall, Englewood Cliffs.
- [Hall and Hall, 1990] Hall, E. T. and Hall, M. R. (1990). *Understanding cultural differences: Germans, French and Americans*. Intercultural Press.
- [Hall and Graham, 2004] Hall, H. and Graham, D. (2004). Creation and recreation: motivating collaboration to generate knowledge capital in online communities. *International Journal of Information Management*, 24(3):235–246.
- [Harhoff et al., 2003] Harhoff, D., Henkel, J., and von Hippel, E. (2003). Profiting from voluntary information spillovers: how users benefit by freely revealing their innovations. *Research Policy*, 32(10):1753–1769.
- [Harrison and Klein, 2007] Harrison, D. A. and Klein, K. J. (2007). What’s the difference? Diversity constructs as separation, variety, or disparity in organizations. *The Academy of Management Review*, 32(4):1199–1228.
- [Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*, chapter 9.4, pages 321–329. Springer Series in Statistics. Springer, 2 edition.

- [Hevner et al., 2004] Hevner, A. R., March, S. T., Park, J., and Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1):75–106.
- [Hlaoittinun et al., 2007] Hlaoittinun, O., Bonjour, E., and Dulmet, M. (2007). A team building approach for competency development. pages 1004 –1008.
- [Hlaoittinun et al., 2008] Hlaoittinun, O., Bonjour, E., and Dulmet, M. (2008). A multidisciplinary team building method based on competency modelling in design project management. *International Journal of Management Science and Engineering Management.*, 3:163–175.
- [Hoffman and Maier, 1961] Hoffman, L. and Maier, N. (1961). Quality and acceptance of problem solutions by members of homogeneous and heterogeneous groups. *Journal of Abnormal and Social Psychology*, 62(2):401 – 407.
- [Hofmann, 1999] Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence, UAI*, Stockholm.
- [Hofmann and Puzicha, 1999] Hofmann, T. and Puzicha, J. (1999). Latent class models for collaborative filtering. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 688–693.
- [Hofstede, 1983] Hofstede, G. (1983). National cultures in four dimensions. a research-based theory of cultural differences among nations. *International Studies of Management and Organization*, 13(1-2):46–74.
- [Hofstede, 1984] Hofstede, G. (1984). *Culture’s consequences : international differences in work-related values / Geert Hofstede*. Sage Publications, Beverly Hills :, abridged ed. edition.
- [Holtzblatt et al., 2010] Holtzblatt, L. J., Damianos, L. E., and Weiss, D. (2010). Factors impeding wiki use in the enterprise: a case study. In *CHI EA ’10: Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems*, pages 4661–4676, New York, NY, USA. ACM.
- [Hoppe, 2006] Hoppe, U. (2006). Kompetenzeentwicklung und Prozessunterstützung in Open Innovation-Netzwerken der IT-Branche durch Wissensmodellierung und Analyse. *Projektskizze zum Projekt*.
- [Horwitz, 2005] Horwitz, S. K. (2005). The compositional impact of team diversity on performance: Theoretical considerations. *Human Resource Development Review*, 4(2):219–245.

- [Hurley, 1998] Hurley, S. (1998). Application of team-based 360° feedback systems. *Team Performance Management*, 4(5):202–210.
- [Ilgen et al., 2005] Ilgen, D. R., Hollenbeck, J. R., Johnson, M., and Jundt, D. (2005). Teams in organizations: From input-process-output models to imoi models. *Annual Review of Psychology*, 56:517–544.
- [Jackson and Joshi, 2004] Jackson, S. E. and Joshi, A. (2004). Diversity in social context: a multi-attribute, multilevel analysis of team diversity and sales performance. *Journal of Organizational Behavior*, 25(6):675–702.
- [Jahrer et al., 2010] Jahrer, M., Töschler, A., and Legenstein, R. (2010). Combining predictions for accurate recommender systems. In *KDD '10: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 693–702, New York, NY, USA. ACM.
- [Johnson and Torcivia, 1967] Johnson, H. H. and Torcivia, J. M. (1967). Group and individual performance on a single-stage task as a function of distribution of individual performance. *Journal of Experimental Social Psychology*, 3(3):266 – 273.
- [Jung, 1971] Jung, C. G. (1971). *Psychological types*, volume 6. Princeton University Press, New Jersey.
- [Kahai and Cooper, 1999] Kahai, S. S. and Cooper, R. B. (1999). The effect of computer-mediated communication on agreement and acceptance. *Journal of Management Information Systems*, 16(1):165–188.
- [Kammergruber et al., 2010] Kammergruber, W., Brocco, M., Groh, G., and Langen, M. (2010). *Competence Management for Open Innovation*, volume 30 of *Schriften zu Kooperations- und Mediensystemen*, chapter Collaborative Lightweight Ontologies in Open Innovation Networks, pages 93–114. EUL Verlag.
- [Karp, 1972] Karp, R. (1972). Reducibility among combinatorial problems. In Miller, R. and Thatcher, J., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press.
- [Kaya and Epps, 2004] Kaya, N. and Epps, H. (2004). Relationship between color and emotion: a study of college students. *College Student Journal*, 38(3):396–405.
- [Kent and McGrath, 1969] Kent, R. N. and McGrath, J. E. (1969). Task and group characteristics as factors influencing group performance. *Journal of Experimental Social Psychology*, 5(4):429–440.

Bibliography

- [Koren et al., 2009] Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42:30–37.
- [Kozlowski and Ilgen, 2006] Kozlowski, S. W. and Ilgen, D. R. (2006). Enhancing the effectiveness of work groups and teams. *Psychological Science in the Public Interest*, 7:77–124.
- [Krackhardt, 1992] Krackhardt, D. (1992). The strength of strong ties: The importance of philos in organizations. In Nohria, N. and Eccles, R., editors, *Networks and Organizations: Structure, Form, and Action*, pages 216–239. Harvard Business School Press, Boston, MA.
- [Kratzer et al., 2005] Kratzer, J., Leenders, R. T., and Van Engelen, J. M. (2005). Informal contacts and performance in innovation teams. *International Journal of Manpower*, 26(6):513–528.
- [Lam et al., 2008] Lam, X. N., Vu, T., Le, T. D., and Duong, A. D. (2008). Addressing cold-start problem in recommendation systems. In *ICUIMC '08: Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 208–211, New York, NY, USA. ACM.
- [Lappas et al., 2009] Lappas, T., Liu, K., and Terzi, E. (2009). Finding a team of experts in social networks. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 467–476, New York, NY, USA. ACM.
- [Latane et al., 1979] Latane, B., Williams, K., and Harkins, S. (1979). Many hands make light the work: The causes and consequences of social loafing. *Journal of Personality and Social Psychology*, 37(6):822–832.
- [Leake, 1996] Leake, D. B. (1996). *CBR in Context: The Present and Future*, chapter 1. AAAI Press/MIT Press.
- [Leake et al., 2002] Leake, D. B., Maguitman, A. G., and Cañas, A. J. (2002). Assessing conceptual similarity to support concept mapping. In *Proceedings of the Fifteenth International Florida Artificial Intelligence Research Society Conference*, pages 168–172. AAAI Press.
- [Li and Shan, 2010] Li, C.-T. and Shan, M.-K. (2010). Team formation for generalized tasks in expertise social networks. pages 9–16.
- [Licorish et al., 2009] Licorish, S., Philpott, A., and MacDonell, S. G. (2009). Supporting

- agile team composition: A prototype tool for identifying personality (in)compatibilities. In *CHASE '09: Proceedings of the 2009 ICSE Workshop on Cooperative and Human Aspects on Software Engineering*, pages 66–73, Washington, DC, USA. IEEE Computer Society.
- [Lyman and Richter, 1995] Lyman, D. and Richter, K. (1995). Qfd and personality type: the key to team energy and effectiveness. *Industrial Engineering*, 27(2):57–61.
- [Malinowski et al., 2005] Malinowski, J., Keim, T., Weitzel, T., and Wendt, O. (2005). Decision support for team building: Incorporating trust into a recommender-based approach. In *PACIS 2005 Proceedings*.
- [Manning et al., 2008] Manning, C. D., Raghavan, P., and Schuetze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- [Mayring, 2007] Mayring, P. (2007). *Qualitative Inhaltsanalyse: Grundlagen und Techniken*. 9th edition.
- [McDonald, 2001] McDonald, D. W. (2001). Evaluating expertise recommendations. In *GROUP '01: Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work*, pages 214–223, New York, NY, USA. ACM.
- [McDonald and Ackerman, 2000] McDonald, D. W. and Ackerman, M. S. (2000). Expertise recommender: a flexible recommendation system and architecture. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pages 231–240, New York, NY, USA. ACM.
- [McGrath, 1984] McGrath, J. E. (1984). *Groups: Interaction and Performance*. NJ: Prentice Hall, Englewood Cliffs.
- [Mears and Voehl, 1994] Mears, P. and Voehl, F. (1994). *Team building : a structured learning approach*. St. Lucie Press, Delray Beach, Fla.
- [Mello and Ruckes, 2006] Mello, A. S. and Ruckes, M. E. (2006). Team composition. *Journal of Business*, 79(3):1019–1040.
- [Meseguer et al., 2003] Meseguer, P., Bouhmala, N., Bouzoubaa, T., Irgens, M., and Sánchez, M. (2003). Current approaches for solving over-constrained problems. *Constraints*, 8:9–39.
- [Montaner et al., 2002] Montaner, M., López, B., and Rosa, J. L. d. l. (2002). Opinion-based filtering through trust. In *CIA '02: Proceedings of the 6th International Work-*

- shop on Cooperative Information Agents VI*, pages 164–178, London, UK. Springer-Verlag.
- [Moore and Serva, 2007] Moore, T. D. and Serva, M. A. (2007). Understanding member motivation for contributing to different types of virtual communities: a proposed framework. In *SIGMIS CPR '07: Proceedings of the 2007 ACM SIGMIS CPR conference on Computer personnel research*, pages 153–158, New York, NY, USA. ACM.
- [Pelled, 1996] Pelled, L. H. (1996). Demographic diversity, conflict and work group outcomes: An intervening process theory. *Organization Science*, 7:615–631.
- [Peter Brusilovsky, 2007] Peter Brusilovsky, A. K. u. W. N., editor (2007). *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg.
- [Powell et al., 2004] Powell, A., Piccoli, G., and Ives, B. (2004). Virtual teams: a review of current literature and directions for future research. *SIGMIS Database*, 35(1):6–36.
- [Powell and Grodal, 2005] Powell, W. W. and Grodal, S. (2005). *The Oxford Handbook of Innovation*, chapter Networks of Innovators, pages 56–85. Oxford University Press.
- [Ramsin and Paige, 2008] Ramsin, R. and Paige, R. F. (2008). Process-centered review of object oriented software development methodologies. *ACM Comput. Surv.*, 40(1):1–89.
- [Raymond, 2001] Raymond, E. S. (2001). *The Cathedral and the Bazaar: Musings on Linux and Open Source by an Accidental Revolutionary*. O'Reilly & Associates, Inc., Sebastopol, CA, USA. Foreword By-Young, Bob.
- [Reichling and Wulf, 2009] Reichling, T. and Wulf, V. (2009). Expert recommender systems in practice: evaluating semi-automatic profile generation. In *Proceedings of the 27th international conference on Human factors in computing systems, CHI '09*, pages 59–68, New York, NY, USA. ACM.
- [Reinhardt et al., 2010] Reinhardt, M., Wiener, M., and Amberg, M. (2010). It support for intra-organizational innovation networks – an exploratory study. In *AMCIS 2010 Proceedings*.
- [Rhodes, 1961] Rhodes, M. (1961). An analysis of creativity. *Phi Delta Kappa*, 42:305–310.
- [Ricci et al., 2011] Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors (2011). *Recommender Systems Handbook*. Springer.
- [Richter and Tolksdorf, 2009] Richter, S. and Tolksdorf, R. (2009). Planning teams with semantic web technologies. In Fischer, S., Maehle, E., and Reischuk, R., editors, *Pro-*

- ceedings of the 4th International Workshop on Applications of Semantic Technologies (AST 2009), part of INFORMATIK 2009*, Lecture Notes in Informatics LNI, pages 396, 3106–3119. Gesellschaft für Informatik.
- [Riege et al., 2009] Riege, C., Saat, J., and Bucher, T. (2009). Systematisierung von evaluationsmethoden in der gestaltungsorientierten wirtschaftsinformatik. In Becker, J., Krcmar, H., and Niehaves, B., editors, *Wissenschaftstheorie und gestaltungsorientierte Wirtschaftsinformatik*, pages 69–86. Physica-Verlag HD.
- [Rigopoulos et al., 2008] Rigopoulos, G., Karadimas, N., and Orsonni, A. (2008). Modeling group decision-making for collaborative teams in enterprises. pages 738–743.
- [Rossi et al., 2006] Rossi, F., Beek, P. v., and Walsh, T. (2006). *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA.
- [Rothstein, 1999] Rothstein, B. (1999). Majority algorithm solution. <http://www.ugcs.caltech.edu/brothste/majority.pdf>.
- [Ru et al., 2008] Ru, Z., Guo, J., and Xu, W. (2008). Improving expertise recommender systems by odds ratio. In Li, H., Liu, T., Ma, W.-Y., Sakai, T., Wong, K.-F., and Zhou, G., editors, *Information Retrieval Technology*, volume 4993 of *Lecture Notes in Computer Science*, pages 1–9. Springer Berlin / Heidelberg.
- [Russell and Norvig, 2003] Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition.
- [Saaty, 2008] Saaty, T. L. (2008). Decision making with the analytic hierarchy process. *International Journal of Services Sciences*, 1(1):83–98.
- [Sarwar et al., 2001] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 285–295, New York, NY, USA. ACM.
- [Schein et al., 2002] Schein, A. I., Popescul, A., Ungar, L. H., and Pennock, D. M. (2002). Methods and metrics for cold-start recommendations. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 253–260, New York, NY, USA. ACM.
- [Schiex et al., 1995] Schiex, T., Fargier, H., and Verfaillie, G. (1995). Valued constraint satisfaction problems: hard and easy problems. In *Proceedings of the 14th international*

- joint conference on Artificial intelligence - Volume 1*, pages 631–637, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Schippers et al., 2003] Schippers, M. C., Hartog, D. N. D., Koopman, P. L., and Wienk, J. A. (2003). Diversity and team outcomes: the moderating effects of outcome interdependence and group longevity and the mediating effect of reflexivity. *Journal of Organizational Behavior*, 24(6):779–802.
- [Schlichter et al., 1998] Schlichter, J., Koch, M., and Chengmao, X. (1998). Awareness - the common link between groupware and community support systems. In *Community Computing and Support Systems*, pages 77–93. Springer Berlin/Heidelberg.
- [Schönrok, 2009] Schönrok, J. E. M. (2009). *Innovation at large: team and project performance in complex new product development*. PhD thesis, University of Groningen.
- [Scott, 2007] Scott, S. (2007). Team performance and the problem-solving approach. *Industrial Technology*, 23(4).
- [Shani and Gunawardana, 2011] Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender Systems Handbook*, pages 257–297.
- [Shim and Srivastava, 2010] Shim, K. J. and Srivastava, J. (2010). Team performance prediction in massively multiplayer online role-playing games (mmorpgs). pages 128–136.
- [Smyth, 2007] Smyth, B. (2007). *The Adaptive Web*, chapter Case-Based Recommendation, pages 342–376. Springer Berlin / Heidelberg.
- [Sommer, 2011] Sommer, V. (2011). Evaluation von Team-Recommender-Algorithmen. Bachelor thesis, Technische Universität München.
- [Steiner, 1972] Steiner, I. D. (1972). *Group Process and Productivity*. Social psychology. Accademic Press, New York, NY.
- [Straus and McGrath, 1994] Straus, S. G. and McGrath, J. E. (1994). Does the medium matter? the interaction of task type and technology on group performance and member reactions. *Journal of Applied Psychology*, 79(1):87–97.
- [Streeter and Lochbaum, 1988] Streeter, L. and Lochbaum, K. (1988). An expert/expert-locating system based on automatic representation of semantic structure. In *Artificial Intelligence Applications, 1988., Proceedings of the Fourth Conference on*, pages 345–350.

- [Takács et al., 2008] Takács, G., Pilászy, I., Németh, B., and Tikk, D. (2008). Investigation of various matrix factorization methods for large recommender systems. In *NETFLIX '08: Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition*, pages 1–8, New York, NY, USA. ACM.
- [Team, 2010] Team, C. (2010). choco: an open source java constraint programming library. Research report 10-02-INFO, Ecole des Mines de Nantes.
- [Thomas and Fink, 1961] Thomas, E. J. and Fink, C. F. (1961). Models of group problem solving. *The Journal of Abnormal and Social Psychology*, 63(1):53–63.
- [Trower and Moore, 1996] Trower, J. K. and Moore, K. K. (1996). A study of the impact of individual goals and team composition variables on team performance. In *SIGCPR '96: Proceedings of the 1996 ACM SIGCPR/SIGMIS conference on Computer personnel research*, pages 206–213, New York, NY, USA. ACM.
- [Turek et al., 2010] Turek, P., Wierzbicki, A., Nielek, R., Hupa, A., and Datta, A. (2010). Learning about the quality of teamwork from wikiteams. pages 17 –24.
- [von Hippel, 2007] von Hippel, E. (2007). Horizontal innovation networks – by and for users. *Industrial and Corporate Change*, 16(2):293–315.
- [Živný and Jeavons, 2009] Živný, S. and Jeavons, P. G. (2009). The complexity of valued constraint models. In *Proceedings of the 15th international conference on Principles and practice of constraint programming, CP'09*, pages 833–841, Berlin, Heidelberg. Springer-Verlag.
- [White, 1984] White, K. B. (1984). An investigation of task team structure and its impact on productivity. In *AFIPS '84: Proceedings of the July 9-12, 1984, national computer conference and exposition*, pages 497–503, New York, NY, USA. ACM.
- [Wong and Burton, 2000] Wong, S.-S. and Burton, R. M. (2000). Virtual teams: What are their characteristics, and impact on team performance? *Comput. Math. Organ. Theory*, 6(4):339–360.
- [Wooldridge, 2002] Wooldridge, M. (2002). *Introduction to Multiagent Systems*. John Wiley & Sons, Inc., New York, NY, USA.
- [Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verb semantics and lexical selection. In *32nd. Annual Meeting of the Association for Computational Linguistics*, pages 133 –138, New Mexico State University, Las Cruces, New Mexico.

- [Zanker et al., 2010] Zanker, M., Jessenitschnig, M., and Schmid, W. (2010). Preference reasoning with soft constraints in constraint-based recommender systems. *Constraints*, 15:574–595.
- [Zeini et al., 2008] Zeini, S., Malzahn, N., Hoppe, H. U., Hafkesbrink, J., Mill, U., Groh, G., Schauf, T., Westermaier, R., Pfeiffer, O., and Scholl, H. (2008). Ansätze zur software-unterstützten Kompetenzentwicklung in innovationsgetriebenen Berufen der Digitalen Wirtschaft. In Meißner, K. and Engelen, M., editors, *Virtuelle Organisationen und Neue Medien 2008 (GeNeMe 2008)*, pages 229–240, Dresden.
- [Ziebarth et al., 2009] Ziebarth, S., Malzahn, N., and Hoppe, H. U. (2009). Using data mining techniques to support the creation of competence ontologies. In *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED 2009)*, Brighton, England.
- [Ziegler and Lausen, 2004] Ziegler, C.-N. and Lausen, G. (2004). Analyzing correlation between trust and user similarity in online communities. In *Proceedings of Second International Conference on Trust Management*, pages 251–265. Springer-Verlag.

A. Interviews

A.1. Outline

Team Composition Criteria and Strategies

- How many teams did you manage that you previously composed in your career?
- Could you choose appropriate team members also among people from outside your company/organization?
- Do usually the teams you compose have a high variety of competencies and functional backgrounds?
- How well do team members usually know each other?
- What kind of hierarchy levels are typically present in these teams?
- Is the cooperation and collaboration rather formal or informal?
- Could you explain with the help of some examples, how do you usually compose teams?

Challenges in Team Composition

- Among how many people can you possibly choose team members?
- Do you consider the task of composing teams challenging considering the amount of possible team members?
- Are there other kind of difficulties when assembling teams?

Support tools and mechanisms

- Do you use any kind of support (IT or traditional) to assemble teams?
- Could you think of any kind of IT support that would make the assembly of teams easier?
- Would you appreciate the implementation of your personal strategy for team composition in an IT system?
- Would you appreciate to use team composition strategies adopted by other people/-companies?
- How do you judge the following team composition approach using IT ... ?

A.2. Statements

Key	Statement
s1	Ich kann mich an Situationen erinnern, in denen ich wusste dass eine Person ein unangenehmer Diskussionspartner ist. Ich habe ihn trotzdem eingeladen, weil er mir in dem Augenblick als jemand erschien, der Wesentliches beitragen könnte.
s2	Es fällt mir jetzt ein Beispiel ein, wo wir uns externe Hilfe geholt haben. Das war jetzt nicht so ein Problem, dass wir selber hätten lösen können. [...] Da haben wir uns dann einen Externen geholt, der uns entsprechend bei der Lösung des Problems geholfen hat.
s3	Wenn ich einen Auftrag habe für eine zehntägige Arbeit, dann nehme ich die zwei, die am besten aufeinander eingespielt sind. Egal wie gut sie sind.
s4	[...] ein Senior Entwickler, jemanden der struktuiert denken kann und was neues ausprobieren möchte, jemanden der konstruktive Kritik einbringen kann, jemanden der mit dem Gesamtsystem lange Erfahrung hat und jemanden der auch Entscheidungsmacht hat. Außerdem müssen auch alle vom gleichen Sprachraum sein um sich schnell verständigen zu können.
s5	... einen Entscheidungsträger vom Kunden, jeweils einen Techniker und einen Business-Vertreter aus jedem Partner, einen Experten aus der Domäne, einen Moderator und eventuell einen Marktforscher.

s6	... Ich habe verschiedene Strategien, aber da ist nichts formalisiertes. Es ist also kein Werkzeug, [...], es ist Einschätzung, Erfahrung und Bauchgefühl.
s7	Es gibt aber auch sehr schwierige Projekte oder Projekte in denen der Kunde schwierig ist. In dem Fall waren sie [das Team] nicht super erfolgreich, aber von Ihnen wurden viel abverlangt und sie waren sehr kreativ.
s8	500 Lebensläufe lesen ist eben in kurzer Zeit nicht möglich.
s9	Wenn ich jetzt ein gewisses Skill suche, und das passiert schon relativ häufig, weil die Kunden sehr genau sagen, wir brauchen jemanden/ein Team, das Expertise in diesen 5 Produkten hat. Und so jemanden zu finden in einem Unternehmen, das immer größer wird, wird immer schwieriger.
s10	Um ein Team innerhalb eines Innovatorennetzwerks wie dem XXXX zusammenzusetzen würde ich gerne ein System haben, das mir sagt was ich habe und was ich insgesamt brauche. Außerdem soll es Anzeigen welche Dinge mir wichtig sind und dementsprechend eine Empfehlung generieren. Diese muss eine Erklärung enthalten, d.h. beschreiben warum dieses Team empfohlen wurde.
s11	Ich suche meistens skillbasiert. D.h. die Team Mitglieder müssen bestimmte Fähigkeiten haben.
s12	Was bei uns sehr gut gepflegt ist, sind die Mitarbeiterprofile. Also jeder Mitarbeiter hat ein Profil, wo seine Projekte drinnen stehen, wo drinsteht, um welches Thema sich dieses Projekt gedreht hat, was seine Aufgaben waren, was waren die eingesetzten Werkzeuge und Produkte.
s13	Was jetzt neu kommt: wir bekommen eine neue Personalsoftware bzw. unsere Personalabteilung, die schon sowas anbietet. Für jeden Mitarbeiter stehen seine Stammdaten drin, was er alles kann, seine Fähigkeiten usw.
s14	Man ist aber dazu übergegangen, eine Datenbank aufzubauen, wo man die Profile der Leute abgespeichert hat, die dann letztendlich auch abgerufen werden konnten. Die Idee dahinter war, dass irgendein Abteilungsleiter oder Projektleiter Zugang zu der Datenbank hat und sich sozusagen seine Leute da raussuchen kann.
s15	Das eine Kriterium ist natürlich die Fähigkeiten und Erfahrungen der Mitarbeiter die gebraucht werden, das andere sind die Kapazitäten die man hat. Natürlich kommen auch noch soziale Kompetenzen dazu, d.h. das Team muss zusammenpassen.

s16	Ich habe Kontakte sowohl innerhalb als auch außerhalb des Unternehmens: Erstmal Universitätskontakte aber auch andere Firmen, das können wir machen und werden nicht abgebremst. Wir sind auch in einem Innovationsverein und da kennen wir auch externe Firmen und mit denen telefoniere ich auch. Ich nenne das mein persönliches Netzwerk. Da gehört das interne Fachberaternetzwerk hinzu und da gehört auch dazu, dass ich mich mit dem einen besser verstehe als mit dem anderen. Denn mit wem ich persönlich nicht zurecht komme kann ich keine Ideen austauschen.
s17	Also man guckt natuerlich schon, haben die Leute in die Richtung schon mal was gemacht und vielleicht wuerfelt man dann unter den Leuten, die schon Know-How haben, auch eine Person, die auch zuarbeiten kann. Die vielleicht noch nicht das Know-How hat, so dass sie sich vielleicht gegenseitig auch ein bisschen austauschen koennen. Also prinzipiell aus Erfahrung, das ist das eine, dann natuerlich Verfügbarkeit, ganz klar.
s18	Klar ist die Methode formalisiert, ich stelle mir mein Team zusammen, ich stelle mir meine Skills zusammen, schaue in die Skilldatenbank, schaue in die Verfügbarkeiten, rufe beim Setting an usw.
s19	Das Management schafft bestimmte Rahmenbedingungen, z.B. versucht es die Verteilung des Projektes zu vermeiden, also die Fragmentierung. Das heißt wenn also von dem Projekt die 5 Leute die es angefangen haben in Stadt A sitzen, dann werden sie nicht noch jemanden aus Stadt B und jemanden aus Stadt C und jemanden aus Stadt D dazu nehmen.
s20	Ich werde auch schauen müssen, dass nicht nur teure Leute dort [im Projekt] drin ist. Bzw. wenn das Projekt das hergibt oder ich keine andere Wahl habe, dann kann oder muss ich es.
s21	Dann gehe ich praktisch an die Skilldatenbank, suche mir die Verfügbarkeitsliste, leg die daneben, gucke, wann diese [die Teammitglieder] verfügbar wären, passt das in meinem Zeitplan und schau ob ich eventuell Aufgaben verschieben kann.
s22	[...] mich hält keiner davon ab wen anderes einzuladen, sobald ich den kenne und ich ihn einlade ist das in Ordnung
s23	Ich habe ein Adressbuch und gehe es auch durch. Aber es ist auch Zufall wer da drinnen steht.

s24	Wir haben die [einige Entwickler] im Forum kennengelernt, und haben gesehen, dass die gute Sachen machen und haben die erst mit kleineren Aufgaben beauftragt. Dann baut sich dann das Vertrauen so auf. Und dann faengt es an mit kleinere Sachen, dann größere und noch größere Sachen. Irgendwann war aber auch der Punkt, wo ich gesagt habe: "Ok, ich will auf jeden Fall mal erstmal runter, lern Sie mal persönlich kennen"
s25	Ja, wir haben 2007 und 2008 sehr stark externe Leute mit eingebunden, weil wir da so viel Bedarf hatten, den wir nicht selber decken konnten. Teilweise war Spezialwissen gefordert, für Bereiche, die wir nicht abdenken konnten.
s26	Wir sind beim extern Verstärken sehr vorsichtig. Vorsichtig deshalb, weil die Einschätzung der Menschen, die dann kommen, die haben wir eben nicht. Da fehlt uns genau dieser Erfahrungswert einer anderen Führungskraft, die diesen Mitarbeiter vielleicht schon längere Zeit kennt.
s27	Es ist etwas schade, dass die Leute den Job seit 40 Jahren machen und immer mit der gleichen Denkweise herangehen. Es kommen somit keine neuen Köpfe hinzu. Es wird Gründe haben, warum auch die benannt werden. Aber ich finde es schade, dadurch kommt nicht viel Neues hinzu.
s28	Es gibt natürlich in der Softwareentwicklung Verfahren, um das systematischer zu machen, nur in der Vergangenheit hatte ich damit Kontakte, und habe da für mich mitgenommen, dass die Erfahrung, also das Bauchgefühl, mindestens genauso gut ist und viel viel schneller geht.
s29	Ich glaube zur Entscheidung wem mit ins Team zu nehmen verwende ich zum einen formale Kriterien und zum anderen das Bauchgefühl.
s30	Das eine hat Vorteile, da kenn ich die Menschen, wenn ich schon einmal gesehen habe wie sie arbeiten, kann sie einschätzen, kann sagen das/derjenige passt für die Aufgabe, integriert sich gut in das Team, oder passt nicht. Das ist die Chance. Das Risiko dabei ist das "schmoren im eigenen Saft". Ich lernen halt nie jemand neuen kennen. Vielleicht gäbe es jemanden, den ich finden würde, wenn ich jetzt in meine Skilldatenbank gehen würde, und eher offener wäre.
s31	Wenn dann ein Kollege frisch von der Uni kommt, der neue Ideen aus der Uni hat und vielleicht andere Erfahrungen aus einem anderen Projekt hat. Dann kann er nochmal ganz viel Schwung reinbringen, ganz viele Sachen besser machen als das Team bis dato gelöst hat.

A. Interviews

s32	Wenn ich die Leute einfach kenne, das ist vielleicht auch ein ganz wichtiges Kriterium , wenn ich mit jemandem schon mal geredet habe vorher, schicke ich dem eher eine E-Mail als wenn das jemand ist den ich gar nicht kenne oder der von mir noch nie gehört hat.
s33	Das habe ich natürlich und gehe es auch durch. Aber es ist auch Zufall wer da drinnen steht.
s34	Und letztendlich muss ich mir vorher überlegen was für ein Profil ich brauche und kann dann auch an die Dienstleister herangehen und sagen: den brauche ich. Dann bekomme ich dafür relativ schnell Vorschläge. In der Regel läuft es dann so ab, dass wir die Leute dann kennenlernen wollen, mit denen ein Interview führen und dann entscheiden müssen. Da kommen dann eben die Kriterien wie soziale Kompetenzen dazu.
s35	Wenn man von diesem Projektteams spricht, dann sind es in der Regel Entwickler plus ein Tester. Was Sie ansprechen, also das wir noch Produktmanager dabei haben, Sales etc., die holen wir uns dann sukzessive rein. [...] Also z.B. am Ende von so einem Projekt, wenn das Produkt schon ziemlich reif wird, ist dann die Produktion mit dabei, dass die dann auch wissen was für eine Produktion dazukommt; da ist natürlich die Logistik dabei, die uns rechtzeitig die Teile beschaffen muss, da ist der Einkauf dabei, weil man die Teile natürlich erstmal einkaufen muss, da ist natürlich ganz am Anfang das Produktmanagement dabei, das letztendlich bei der Definition des Produkts beteiligt ist