ARTICLE IN PRESS

Technological Forecasting & Social Change xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Technological Forecasting & Social Change

journal homepage: www.elsevier.com/locate/techfore

Web-based software-support for collaborative morphological analysis in real-time

Marin Zec*, Florian Matthes

Technische Universität München, Chair for Software Engineering for Business Information Systems (SEBIS), Boltzmannstr. 3, 85748 Garching, Germany

A R T I C L E I N F O

Keywords:
Virtual teams
Collaboration software
Real-time collaboration
Wicked problems
Problem-structuring method
Collaborative morphological analysis
General morphological analysis

A B S T R A C T

Existing software and procedures for General Morphological Analysis (GMA) are primarily designed for synchronous face-to-face meetings. However, virtual teams and telework are on the rise. Against this background, we analyze current approaches and IT support to identify aspects that need to be reconsidered when GMA is applied in a distributed setting. In cooperation with a German non-profit cultural organization, we have developed browser-based collaborative GMA software that provides multi-user support. This paper presents what we have learned from the development process and the results from two empirical studies on the usability and learnability of the developed software. Based on observations and user feedback from the empirical studies, we conclude that the developed software is a useful IT artefact; more research is needed, however, to investigate the implications of distributed team settings for the application and facilitation of GMA.

1. Introduction

Many real-world decision-making problems are “wicked problems” (Rittel and Webber, 1973). One key characteristic of wicked problems is that no single computational formulation of the problem at hand is sufficient to integrate all of the different points of view of its stakeholders (Introne et al., 2013). As a result, hard operations research (OR) methods which require an adequate, straightforward mathematical formulation of the problem (e.g. linear optimization or simulation) may be of help, but are not sufficient to tackle wicked problems holistically. Wicked problems do not have right or wrong solutions (Conklin, 2006; Rittel and Webber, 1973): rather, solution candidates may be considered better or worse from different points of view (Schoder et al., 2014). Thus, soft OR methods have been proposed as an alternative. Whereas the purpose of hard OR techniques is optimal or near-optimal problem *solving*, the key objective of soft OR approaches is problem *structuring*. Various problem structuring methods (PSMs) have been proposed in the literature, among them General Morphological Analysis (GMA) (Ritchey, 2006, 2011; Zwicky, 1971). GMA has been applied in various domains such as policy planning, strategic foresight and idea generation. The core objective of GMA is to promote shared understanding among stakeholders.

Since hard OR approaches build strongly on mathematical methods, early on a strong argument had been proposed to leverage the computational resources of information technology (i.e. software and hardware) to apply them efficiently. In the case of soft OR, the benefits of using information technology are not as obvious. However, Schoder et al. (2014) contend that there is “a significant lack of appropriate information systems (and functionality) that contribute to addressing wicked problems successfully”. They call for more research on and development of information systems for tackling wicked problems. The authors envision information systems which provide appropriate functionality to harness collective intelligence.

For GMA, various examples of these types of information systems such as MA/Carma (Ritchey, 2016) or Parmenides EIDOS (Parmenides, 2016) have been developed. GMA involves iterative cycles of analysis (or: problem decomposition) and synthesis steps to create a shared morphological model (Ritchey, 2011). First, during analysis, the initial problem is decomposed into key parameters. Then, possible values for each parameter are generated. The parameters and associated values define the formal solution space and are represented in a so-called Zwicky Box (i.e. a morphological field or morphospace). Solutions to the initial problem can then be generated by combining different partial solutions. In practice, many configurations are not viable for various reasons (e.g. logical or physical constraints). To take viability into account and reduce the combinatorial explosion, GMA software features pairwise cross-consistency assessment of parameter values (CCA) for the semi-automated synthesis and interactive visualizations of the solution space (e.g. Ritchey, 2006). Thus, dedicated software provides substantial benefits over manual GMA.

Today, knowledge workers facing wicked problems are increasingly engaged in distributed work practices: telework and (heterogeneous)

* Corresponding author.
E-mail address: marin.zec@mytum.de (M. Zec).

http://dx.doi.org/10.1016/j.techfore.2017.05.018
Received 25 November 2016; Received in revised form 15 May 2017; Accepted 16 May 2017

virtual teams are on the rise (Ferrazzi, 2014). Distributed work involves collaboration across "locational, temporal and relational boundaries to accomplish an interdependent task" (Martins et al., 2004). In virtual teams, communication and shared understanding tend to be difficult to achieve for various reasons such as demographic differences or having different information, assumptions and preferences due to geographical dispersion (Durnell Cramton and Hinds, 2004). However, shared mental models of the task environment among team members have been shown to improve team performance (van den Bossche et al., 2011). Therefore, problem structuring plays a key role for effective collaboration both in collocated as well as virtual teams.

The development of the current generation of GMA software started in the 90s when internet-based communication tools such as the worldwide web or e-mail were still in their infancy and, consequently, telework was not as prevalent as it is today. As a result, GMA software packages typically provide a single-user interface since they have been primarily designed for face-to-face workshops during which a facilitator and/or recorder operates the software to capture the group discussion and results.

In this research we partnered with the Goethe Institute (GI), a German non-profit association operating worldwide. From February until May 2015, GI has offered a free, public Massive Open Online Course (MOOC) open to interested individuals worldwide. Many participants worked in teams as desired by the course organizers. During one of their MOOC assignments, the collocated participants were introduced to GMA for scenario and strategy development. Unfortunately, many participants had difficulties to apply GMA in a distributed, collaborative setting. As a result, in August 2015, GI approached the authors to develop an IT artefact that would support participants of the upcoming 2016 edition of the "Managing the Arts" MOOC and address the issues they observed in the first run. After reflecting on their experience throughout the pilot run and discussing potential adjustments to the course, the course organizers concluded that GMA remains to be the method of choice for the group problem solving exercises. However, GI was looking for new ways to facilitate the application of GMA in future instances of the course.

1.1. Research objective

The general objective of this research is to investigate how GMA can be applied efficiently and effectively in distributed settings. More specifically, the aim of our work is two-fold. First, GI approached the researchers to address its specific real world problem concerning the use of GMA as a problem-structuring method in a large-scale, distributed online setting. Second, we want do derive generalizable design knowledge on how to make GMA more feasible in other distributed work contexts.

Since available GMA software lacks functionality required for distributed group work such as multi-user support, we have developed a collaborative, web-based GMA software which supports both synchronous as well as asynchronous group work to account for use cases where face-to-face meetings are not feasible (e.g. because of prohibitive costs or scheduling issues). We propose an artefact-oriented reference process model for collaborative GMA which aims to address common pitfalls in group work and inform our current software design.

1.2. Research approach

We followed an Action Design Research (ADR) approach as described by Sein et al. (2011). Design research seeks to develop prescriptive design knowledge of IT artefacts intended to solve a certain class of problems (Sein et al., 2011). In contrast to traditional design research approaches which propose stage-gate models and separate evaluation from building, the underlying premise of ADR is that IT artefacts are shaped by the organizational context during development and use. Sein et al. (2011) propose four stages to go through when conducting an ADR project Fig. 1:

1. Problem Formulation (see Section 2)
   Identification and formulation of a problem perceived in practice or anticipated by researchers (see Section 1).
2. Building, Intervention, and Evaluation (see Sections 5 and 6)
   In close cooperation with practitioners, the IT artefact is designed, developed and refined during one or more cycles of building, intervention and evaluation (BIE). According to IT-dominant BIE as proposed by Sein et al., lightweight interventions in the form of emerging, early "alpha" versions are developed first, instantiated in a limited organizational context and subjected to the assumptions, expectations, and knowledge of the practitioners. Then, building on the insights from these initial interactions, a more mature "beta" version of the artefact is put into a wider organizational context.
3. Reflection and Learning (see Sections 3 and 4)
   This stage is continuous and parallel to the first two stages. Researchers are encouraged to reflect on the particular solution design and identify learnings to the broader class of problems.
4. Formalization of Learning (see Sections 7 and 8)
   The objective of the final stage is to formalize the learning and builds on the reflection and learning activities throughout the ADR process. Insights and artefacts are then extended to a broader class of problems and solutions.

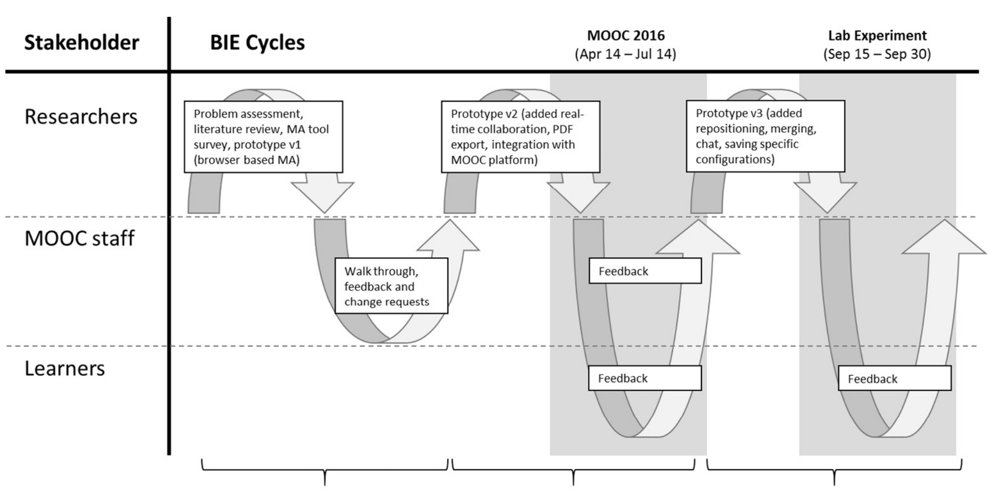
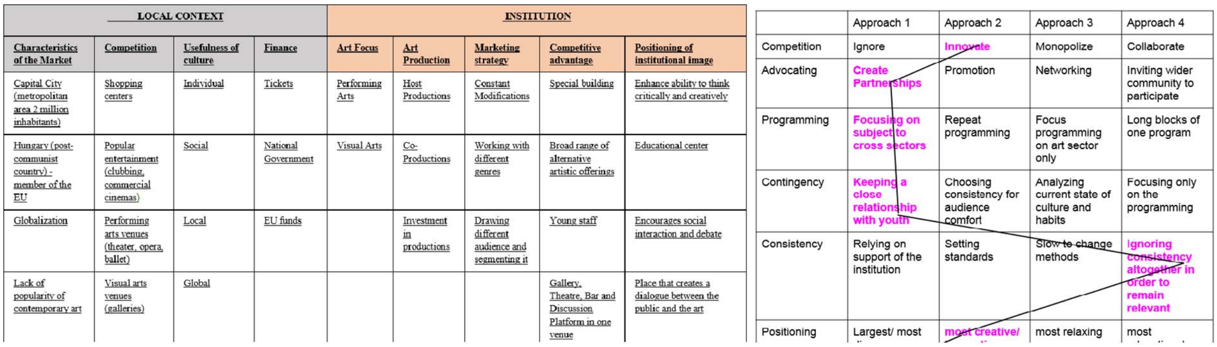


Fig. 1. An overview of the three BIE cycles performed during this research project.



LOCAL CONTEXT				INSTITUTION				
Characteristics of the Market	Competition	Usefulness of culture	Finance	Art Focus	Art Production	Marketing strategy	Competitive advantage	Positioning of institutional image
Capital City (metropolitan area 2 million inhabitants)	Shopping centers	Individual	Tickets	Performing Arts	Host Productions	Constant Modifications	Special building	Enhance ability to think critically and creatively
Hungary (post-communist country) - member of the EU	Popular entertainment (clubbing, commercial cinemas)	Social	National Government	Visual Arts	Co-Productions	Working with different genres	Broad range of alternative artistic offerings	Educational center
Globalization	Performing arts venues (theater, opera, ballet)	Local	EU funds		Investment in productions	Drawing different audience and segmenting it	Young staff	Encourages social interaction and debate
Lack of popularity of contemporary art	Visual arts venues (galleries)	Global					Gallery, Theatre, Bar and Discussion Platform in one venue	Place that creates a dialogue between the public and the art

	Approach 1	Approach 2	Approach 3	Approach 4
Competition	Ignore	Innovate	Monopolize	Collaborate
Advocating	Create Partnerships	Promotion	Networking	Inviting wider community to participate
Programming	Focusing on subject to cross sectors	Repeat programming	Focus programming on art sector only	Long blocks of one program
Contingency	Keeping a close relationship with youth	Choosing consistency for audience comfort	Analyzing current state of culture and habits	Focusing only on the programming
Consistency	Relying on support of the institution	Setting standards	Slow to change methods	Ignoring consistency altogether in order to remain relevant
Positioning	Largest/ most	most creative/	most relaxing	most

Fig. 2. Two examples of submissions from MOOC participants in 2015.

We performed three BIE iterations. Our building activities followed an agile approach, i.e. prototyping included frequent interactions with GI staff to capture evolving requirements based on the incremental changes to the software prototype. In this way, we ensured the validity of the system requirements and aimed for completeness of the requirements (in the context of the MOOC). In the first cycle, a basic alpha prototype was developed, demonstrated and discussed. Subsequently, the prototype was developed further during the second BIE cycle and put in production for the 2016 edition of the MOOC. Based on the observations of the MOOC staff and the authors, as well as feedback from participants, another BIE cycle was started to refine the prototype. In the present paper, we present the result of the third BIE iteration (i.e. revised requirements, version 3 of the prototype and results from a lab experiment that investigated the usability of the prototype).

2. Problem formulation

2.1. Research partner

The mission of the GI is to promote cultural exchange, international relations and the study of the German language abroad. Within its cultural education program, GI offered a free, public 14-week Massive Open Online Course (MOOC) titled “Managing the Arts” from February until May 2015. This MOOC was aimed at cultural managers and individuals interested in cultural work. More than 17,000 participants from 176 countries registered for the pilot course in 2015. During six assignment cycles, participants learned about theoretical foundations of cultural management and developed marketing strategies for one of the four partnering art institutions. Participants were strongly encouraged to collaborate with each other, share their experiences and create a community of knowledge using the provided online platform. A limited number of participants had the opportunity to receive a certificate of completion upon finishing the course. These participants were assigned to small groups. The remaining participants were encouraged to form teams on their own (but were not required to do so). Within one of their case study assignments, participants were asked to develop scenarios and respective strategies for a particular art institution. Both scenario analysis and strategy development can be understood as wicked problems, since they exhibit several of the corresponding characteristics such as multiple stakeholders, potentially conflicting objectives and multidimensionality. Thus, students were introduced to the concept of wicked problems. Among various alternative approaches to structure wicked problems, the course organizers determined GMA as the method of choice within the course. Participants received materials introducing GMA and were asked to apply GMA to structure their scenario development process and outcome. However, many teams had difficulties in applying the technique.

2.2. Formulation of the real-world problem

Due to the great popularity and success of the pilot course, GI decided to run a revised version of the course in 2016. GI acknowledged that GMA is an appropriate and useful technique within the course. However, several issues occurred during the first run of the MOOC which can be divided into three categories: (1) logistical difficulties, (2) methodological difficulties, and (3) social loafing.

2.2.1. Logistical difficulties

Since there was no dedicated GMA software available to the participants, they resorted to different general purpose software. For instance, some teams have used offline spreadsheet software to document their morphological box and exchanged files via e-mail (e.g. see Fig. 2 on the left). Others created tables using web-based collaborative word processors. Since participants differed in the IT infrastructure that was available to them, they had to expend significant effort to coordinate their teamwork. In addition, group mentors had difficulties reviewing submissions because of the different file formats and formatting.

2.2.2. Methodological difficulties

Participants had difficulties in the synthesis stage because of the lack of dedicated GMA software. They relied on a manual, intuitive approach to develop scenario configurations (Fig. 2 on the right). Due to the combinatorial explosion of possible configurations, systematic and comprehensive synthesis was not feasible. Consequently, participants were not as confident with the scenario alternatives they derived as they would have been had they been able to examine the entire scenario space. In addition, general purpose software seemed to promote modelling errors and misconceptions of crucial aspects of GMA because of the high degree of freedom it provides (e.g. in terms of formatting) and the large effort required for subsequent model changes.

2.2.3. Social loafing

A number of participants complained that some team members put significantly less effort into the teamwork causing dissatisfaction among fellow team members. In social psychology, this phenomenon is termed “social loafing” and has been studied extensively. Social loafing tends to negatively impact group performance and should therefore be mitigated.

2.3. Generalizing the problem

The problem situation encountered by GI represents an important class of problems that arise when distributed teams want to use GMA to tackle a multi-dimensional, wicked problem. To date, GMA research and practice have primarily focused on using GMA in face-to-face meetings. However, geographical distribution and increased use of technology present new challenges for group work. While the key tenets of classical GMA can be transferred to distributed collaboration scenarios, two major aspects of GMA need to be reconsidered: *facilitation* and *software support*. For instance, in synchronous face-to-face meetings, a facilitator can typically directly intervene in the group

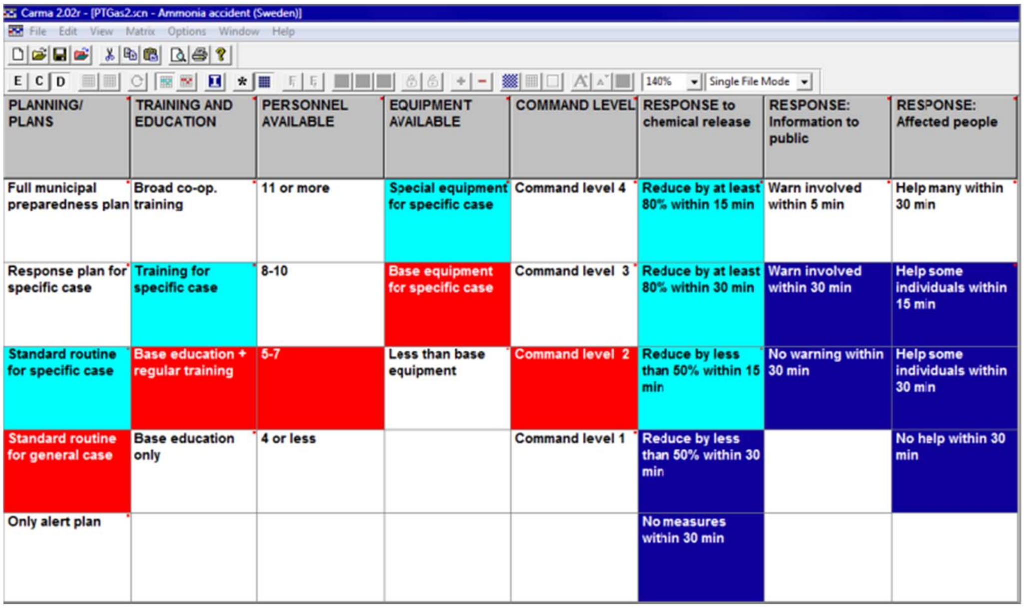


Fig. 3. MA/Carma provides an interactive what-if inference model.
(Source: Ritchey, 2016).

process and manage group dynamics. However, in virtual teams the group interaction is mediated via technology and might perhaps be asynchronous, making it considerably more difficult for a facilitator to stimulate active listening and co-construction of meaning, or to manage constructive conflict. Another obstacle for the efficient use of GMA in distributed settings is the lack of GMA software that provides multi-user support.

In this research project, we primarily address the lack of adequate software support for (distributed) collaborative GMA and present a starting point for a broader discussion of implications on the GMA process and its facilitation in distributed teamwork.

3. Related work

3.1. Software for morphological problem structuring

Various software packages for morphological problem structuring have been developed. Conventional software packages providing morphological problem structuring functionality such as MA/Carma, Parmenides EIDOS' Option Development, ACTIFELD (Coyle, 2004), MEMIC (Arnold et al., 2017) or MORPHOL (Bourse and Godet, 2016) are designed for face-to-face meetings. While all of these packages facilitate morphological problem structuring, each of them constitutes a different derivative of Fritz Zwicky's original approach to GMA (Zwicky, 1969). For instance, MORPHOL is part of a larger toolbox Godet et al. developed to support their relatively sophisticated foresight approach *la prospective* (Godet, 2006). As one of several different modules, Parmenides EIDOS provides a morphological problem structuring component termed “Option Development” which is primarily targeted at supporting scenario development. By contrast, besides scenario development, MA/Carma is targeted at a broader scope of wicked problems and adheres most closely to the principles Zwicky laid down (Voros, 2009). Except for MA/Carma and Parmenides EIDOS, the other software packages seem to have been discontinued.

Building on previous work such as MORPHOL, Bourse and Godet recently developed a web-based “Scenaring Tools” software suite. While Scenaring Tools can be used without charge, their use compels users to adhere to the sophisticated *la prospective* approach pioneered by Godet et al. The feature-rich interface and process involves a relatively steep learning curve (considered too steep for the inexperienced MOOC participants). In addition, Scenaring Tools do not feature real-time, multi-user collaboration. Another web-based GMA software is being developed under the name “Fibonacci MA” (FMA) (Childs and Garvey, 2015). While FMA is largely based on GMA, similar to Scenaring Tools, it does not provide real-time, multi-user collaboration.

All of the above-mentioned software packages differ to some extent in their functionality but still provide substantial support for synchronous, face-to-face GMA processes. Besides historical, technological issues, a major reason why GMA software has traditionally focused on synchronous face-to-face meetings is that its creators acknowledge the importance of professional facilitation and group dynamics. After all, GMA is a soft OR technique that emphasizes stakeholders over technology. Therefore, this research does not aim to replace dedicated face-to-face GMA software but rather complement it for distributed use cases to allow adoption of GMA when face-to-face meetings are not feasible. However, not only technical issues such as multi-user support are important to consider when designing GMA software virtual teams but methodical aspects (e.g. facilitation, process adherence and group dynamics) should be considered as well if GMA is to be conducted effectively in virtual teams (Zec et al., 2015).

3.2. Process steps and key artefacts of computer-aided morphological analysis

Common to all variants of GMA is their artefact-orientation. Typically, morphological approaches are based on three types of artefacts which structure group discussion and capture the shared mental model of the group. The first type of artefact, the Zwicky Box, is the result of analysis activities (i.e. parameter and parameter value specification). The second type of artefact, the consistency matrix, is the result of synthesis activities (i.e. consistency assessments). In the course of the ongoing group interaction, multiple iterations between analysis and synthesis may lead to modifications of the Zwicky Box and consistency matrix. Once the group has settled on a Zwicky Box and consistency matrix, GMA software can create an instance of the third type of artefact: a visual representation of the solution space. For instance, MA/Carma generates an interactive what-if inference model (Fig. 3). On the other hand, Parmenides EIDOS calculates a consistency score for each configuration and provides a cluster view for the solution space (Fig. 4). As a result, the provided main views of GMA software tend to revolve around one of the three artefact types (1.) Zwicky Box, (2.) consistency matrix, and (3.) solution space visualization.

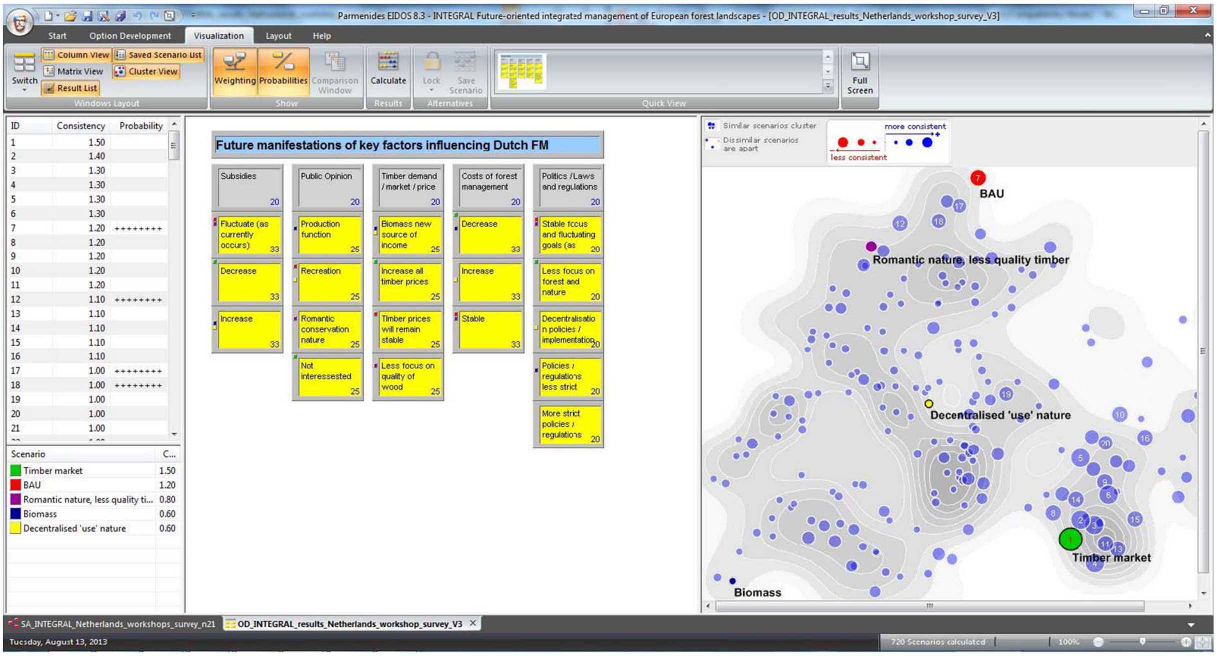


Fig. 4. The cluster view in Parmenides EIDOS performs dimensionality reduction to project the solution space on a two-dimensional plane.
(Source: Future-Oriented Integrated Management of European Forest Landscapes, 2014).

3.3. Facilitation & group procedures

The social processes that are stimulated through soft OR methods in general, and GMA in particular, can be conceptualized as team learning behaviors aimed at constructing a shared mental model. van den Bossche et al. (2011) developed a model for team learning behavior suggesting that team learning behaviors lead to a shared mental model which, in turn, improves team performance. The authors argue that a shared mental model (i.e. mutual understanding *and* mutual agreement) is a result of three types of socio-cognitive processes: (1) construction, (2) co-construction, and (3) constructive conflict. Based on this model, team learning starts with one team member who “inserts meaning by describing the problem situation and how to deal with it” (van den Bossche et al., 2011). Fellow team-members listen actively and try to “grasp the given explanation by using this understanding to give meaning to the situation at hand” (van den Bossche et al., 2011). This process of construction then evolves into co-construction: team members engage in a mutual process of building on and refining the original contribution (e.g. revising parameters or parameter values). Both construction and co-construction of meaning ultimately lead to mutual understanding. However, a shared mental model requires not only mutual understanding but mutual agreement as well. Since team members initially have different points of view and their interpretations diverge at least in some aspects, they need to engage in constructive conflict to achieve mutual acceptance, i.e. “dealing with differences in interpretation between team members by arguments and clarifications” (van den Bossche et al., 2011). In the case of GMA, team members do not only need to *understand* their joint morphological model but also finally *agree*, i.e. commit themselves, to it. In practice, dedicated facilitators can stimulate these processes of (co-)construction and constructive conflict. Ritchey argues that facilitation is the “least appreciated” aspect of GMA workshops (Ritchey, 2011).

A large body of literature on how group processes affect group performance has found various types of process gains and process losses in group processes (for an overview, see Forsyth, 2014). Process gains refer to benefits that arise from using groups. Conversely, process losses refer to phenomena in groups that negatively impact group performance. For instance, numerous studies have studied team performance in verbal brainstorming and found that interactive (“real” groups) tend to produce fewer and less creative ideas than “nominal” groups (i.e. non-interacting groups). One major explanation, among others, is referred to as “production blocking” (Diehl and Stroebe, 1991): since only one team member can talk effectively at one time, fellow team members might forget ideas while waiting to speak or decide not to state their ideas at all (e.g. if a similar idea has been proposed before their turn). Another strategy to reduce process losses and promote process gains besides facilitation is the adoption of certain group procedures. For instance, to address the issue of production blocking, traditional brainstorming can be replaced with brainwriting or electronic brainstorming (Nunamaker et al., 1991) where turn taking is abandoned (i.e. team members can document their ideas in parallel). Examples of well-known group procedures are the Delphi method (Dalkey and Helmer, 1963) and the Nominal Group Technique (Delbecq, 1971).

GMA is a highly iterative approach (Ritchey, 2011). All reviewed software packages take this into account and allow arbitrary transitions between the different phases to update the morphological model based on new insights and data emerging over time during group interaction. In a face-to-face GMA workshop, there is typically a facilitator who guides the transitions between different stages and manages (co-)construction, creative conflict and group dynamics. Facilitation remains an important concern in virtual teams.

4. Requirements

During the initial meeting with a representative of the MOOC project team, the difficulties the MOOC staff and/or participants encountered were documented. A review of related literature and existing approaches followed (see Section 3). Based on the initial problem assessment during the initial meeting with the MOOC project manager, and a subsequent review of related literature and existing approaches, we developed a lightweight “alpha” prototype (version 1). In the first “alpha” prototype, we focused on the general user interface (at this point there was no support for real-time collaboration and user management). The early tangible prototype greatly aided the specification of more concrete requirements and expectations of the practitioners. Table 1 lists the revised requirements collected during the first BIE cycle.

Table 1
Requirements for version 2 of the envisioned software gathered in collaboration with GI's MOOC staff (BIE cycle 1).

ID	High-level requirement and rationale	Priority	Success criteria
1	The GMA software must facilitate the creation and manipulation of a Zwicky Box and consistency matrix	High	The user can define parameters and parameter values to create a Zwicky Box The user can rename parameters and parameter values to adapt the Zwicky Box to his/her current mental model The user can delete parameter and parameter values to adapt the Zwicky Box to his/her current mental model The user can assign pair-wise consistency assessments to create a consistency-matrix based on the Zwicky Box he/she provided The user can update each consistency assessment to adapt the Zwicky Box to his/her current mental model The Zwicky Box and consistency matrix are automatically and continuously persisted to the database
2	The GMA software should run in a web browser	High	The user can run the GMA software using a modern standard web browser (i.e. released in 2013 or later) without having to perform installation or configuration
3	The GMA software must be easy to use	High	The user interface is clear and accessible
4	The GMA software must provide a means for real-time collaboration among arbitrary users within a group (if desired).	High	Users belonging to the same group work on the same Zwicky Box and consistency matrix to allow co-creation Each user's updates of the Zwicky Box and consistency matrix are automatically propagated to fellow team members such that all team members see and work on the most recent version without the need to trigger data updates manually
5	The GMA software must allow users to export the Zwicky Box and consistency matrix in a standard file format	Medium	The user can export both the Zwicky Box and the consistency matrix as a PDF document
6	The GMA software must be integrated into the existing technical MOOC platform	Low	The user can access the GMA software from within the general MOOC platform without having to provide user credentials

Based on this list of requirements, the GMA software was continuously developed further into a more mature beta version (version 2) until the MOOC staff considered it to be ready for use (i.e. requirements 1 to 6 were satisfied). The beta version was then put into production for the course. It was particularly important for GI and its partners responsible for the content and didactic implementation, that participants should not be forced to use the software and/or provide feedback. Participants were encouraged to take part in a voluntary feedback survey following the course. In general, the software satisfied the requirements above. One occasional issue related to the MOOC platform integration, caused by participants who switched their teams spontaneously without notice, was addressed by manual updating the user database. With respect to the prototype as such, we have added three new requirements on the GMA software based on explicit feature requests and our own observations (Table 2). First, users want to be able to change the position of parameters and parameter values to indicate the perceived strength of relationships spatially. Second, to make communication among participants easier and reduce media discontinuity, a requirement for a means of group communication was added. Third, to make the collaborative revision of the Zwicky Box easier, users should be able to group existing parameters and their parameter values together and not be required to delete two parameters and then create a new one just to address redundancy.

We conducted an additional BIE cycle to address these additional requirements (7–9).

5. Solution

We strived for completeness and validity in these requirements and revised their specifications continuously over the course of three BIE cycles. The two key features of our Collaborative Morphological Analysis (CMA) software are (1) real-time multi-user collaboration support and (2) an option for the facilitator to switch between individual and group workspaces as desired during both the analysis and synthesis stage (similar to the Delphi method). In CMA, each GMA project is immediately persisted in the database. Thus, users can collaborate both synchronously and asynchronously. CMA can be used individually as well. The interface responds to the size of the browser and features five key views: the problem description view, the analysis view, the synthesis view, the exploration view and an export view.

5.1. An artefact-oriented process model & facilitation support

Based on our review of literature and existing GMA software, we derived a reference process model for collaborative GMA. The model builds on Zwicky's original GMA approach (Zwicky, 1971), the GMA process model presented by Ritchey (2011) and our review of current variants and software tools. We also took into account the issue of facilitation in distributed settings.

To a large extent, our reference process model (Fig. 5) is very similar to Zwicky's and Ritchey's process. In the first step, problem description, the problem is framed and described as concisely as possible. In the second phase, the problem or system is broken down into parameters and parameter values resulting in the Zwicky Box. Then, in the third phase, possible solutions are generated—typically based on pair-wise consistency (e.g. MA/Carma) or compatibility assessments (e.g. Parmenides Eidos). Finally, the generated solution candidates (i.e. configurations) are examined using one or more types of representations (e.g. table, cluster view, what-if inference model). Since the morphological model evolves over time, the process is highly iterative. For instance, new arguments and additional information emerge or modelling mistakes are often noticed only later in the process. However, there

Table 2
Requirements for version 2 of the envisioned software based on observations and empirical insights from the second MOOC run in 2016 (BIE cycle 2).

ID	High-level requirement and rationale	Priority	Success criteria
7	The GMA software must allow the repositioning of parameters and parameter values	High	User can change the position of both parameters and parameter values
8	The GMA software must provide a means to allow group members to communicate with each other	High	Users can use a group chat functionality to communicate with each other
9	The GMA software must provide a means to group parameters together	Medium	Users can select two parameters and combine them into one single parameter

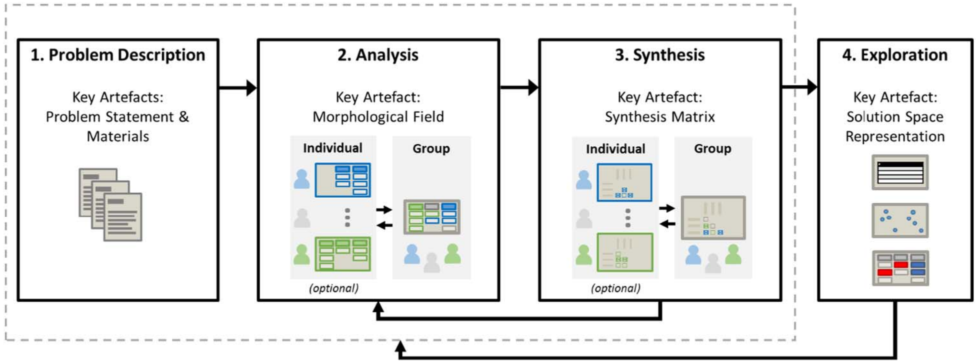


Fig. 5. The CMA process model. The individual sub-steps are optional.

are two major differences to the processes of Zwicky and Ritchey.

First, in contrast to their more activity-oriented process description, we consciously opted for an artefact-oriented process model. Our reference process model describes artefacts on a more abstract level. This leaves room for our concrete implementation to support multiple types of artefacts. For example, depending on the problem situation, a what-if inference model, a cluster view, or both representations of the solution space might be helpful to the user since each type of representation serves a different purpose. Another example concerns the synthesis stage. Some approaches such as MA/Carma (i.e. GMA) focus on discrete notions of consistency and inconsistency whereas other approaches such as Option Development within Parmenides EIDOS rely on degrees of compatibility typically expressed using an ordinal scale (e.g. -3 (strongly incompatible) to $+3$ (strongly compatible)). Another reason why we placed emphasis on the resulting artefacts instead of prescriptive, sequential activities within each stage is that it allows us to experiment with alternative user interface concepts to improve the efficiency and ease of use. While our process model leaves the exact choice of the supported types of artefacts open, the current version of the developed GMA software only supports Ritchey's GMA approach because of its straightforwardness and accessibility. Building the software with the general artefact-oriented process model in mind helped us to take expandability into account and aim for modularity such that additional types of artefacts may be added in future versions.

The second major difference to traditional approaches is that our reference process model aims to explicitly take into account the issues of team-learning, group dynamics and facilitation. Because a facilitator in an (asynchronous) virtual session tends to have less control over the group interaction, we try to pre-structure the group process. Our adjustments to the GMA process are inspired by key ideas of the Delphi method and Nominal Group Technique: Both methods disentangle ideas and judgments from their proponents to eliminate various pitfalls and biases in group discussion and decision-making such as conformity (e.g. Kelman, 1958) and individual dominance. In the Delphi method, there is no direct interaction between group members. Instead, participants provide estimates and arguments anonymously in iterative rounds. In each round, the individual contributions are combined and provided to the participants as a starting point for the next round until consensus is met.

In the case of the Nominal Group Technique, participants first start to generate ideas or judgments individually. In the subsequent step, all contributions are shared and discussed within the group. Finally, group members vote on and rank ideas or judgments. In this way, both techniques also address the problem of production blocking. Another potential benefit of individual sub-steps is that social loafing might be mitigated to some extent since participants are more likely to feel that their individual effort is justified and appreciated. We have adopted the key tenets of the Delphi method and Nominal Group Technique for our process model and propose to conduct parallel individual analysis and synthesis steps prior to the classical joint analysis and synthesis. In this way, group members can create their private Zwicky Box and synthesis matrices free of influence and priming from other team members. However, to take synchronous applications which resemble conventional face-to-face meetings into account, the individual analysis and modelling steps are considered to be optional.

5.2. Problem description view

When users open a CMA project, the first view that is shown is the problem description view. The purpose of this view is to display the description of the task or issue associated with the project. This view is non-interactive since the MOOC staff provides fixed assignments to the users. This can easily be extended to support a more sophisticated and/or interactive problem description view. Within this project, the problem description view was kept simple, since problem description materials were provided on the general MOOC platform.

5.3. Analysis view

The second view available in CMA is the analysis view. In the analysis view, users can define the Zwicky Box (Fig. 6). Parameters and parameter values can quickly be added, modified, merged (see Fig. 7 and Fig. 8) and deleted (Fig. 9). While the visual ordering of parameters is irrelevant for the semantics of a Zwicky Box, users can reposition parameters and parameter values. This feature was requested by multiple MOOC participants (Table 2).

The choice of the interaction mode controls what users can see and do in the analysis view. If the individual mode is activated, users can only see, edit, merge, delete and reposition their own parameters and parameter values (Fig. 6). However, if the group mode is active, the analysis view displays all contributions from each group member (Fig. 7) as well as a chat and team activity feed. The chat component allows group communication. The team activity feed lists all team actions chronologically.

5.4. Synthesis view

The third view available is the synthesis view. The synthesis view provides a cross-consistency matrix template to be filled in. The user can select one or more value pairs and assign a consistency value, provide his/her confidence level and a justification (Fig. 10). If the individual mode is active, the user can only see his/her own consistency assessments. However, if the group mode is active (Fig. 11), all consistency assessments from all group members are visible. If group members provided diverging consistency values, a lightning symbol is displayed in the respective cell of the consistency matrix. If all provided consistency values for a value pair match, the associated cell displays the respective consistency value. It is important to note that CMA does not force each group member to fill in the entire consistency matrix, yet

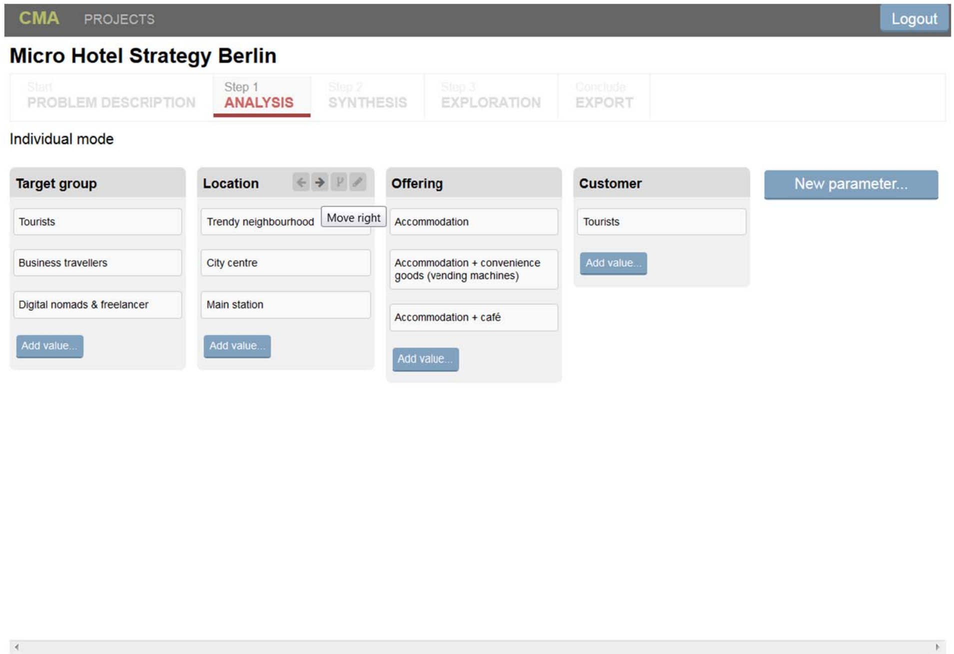


Fig. 6. Individual mode analysis view: users can only see and manipulate their own morphological box.

he or she is free to do so. To resolve disagreements among group members with respect to value pair consistency ratings, CMA displays the distribution of proposed consistency values and—if provided—the associated justification. This allows users to re-evaluate their position and adjust their consistency rating if desired. In addition, the chat window can be used to communicate.

5.5. Exploration view

The exploration view provides a what-if inference model similar to MA/Carma (Fig. 12). Users can select certain parameter values and, based on the consistency ratings provided during the synthesis stage, the remaining values are colored according to their fit to the current selection of values. In this way, internally consistent solution candidates can easily be identified, assigned a name and saved to the solution candidate list.

5.6. Export view

In the current prototype, the export view is very basic. While users do not have to save the project explicitly, since all changes are automatically persisted in the database, MOOC participants were required to submit their final results as a digital file. To facilitate this process, the export view allows users to export the current version of their Zwicky Box and/or consistency matrix as a PDF file.

6. Evaluation

6.1. Experimental design

We assessed the usability of our prototype in a lab experiment. Subjects were recruited from the laboratory for experimental research operated by the School of Management at Technical University of

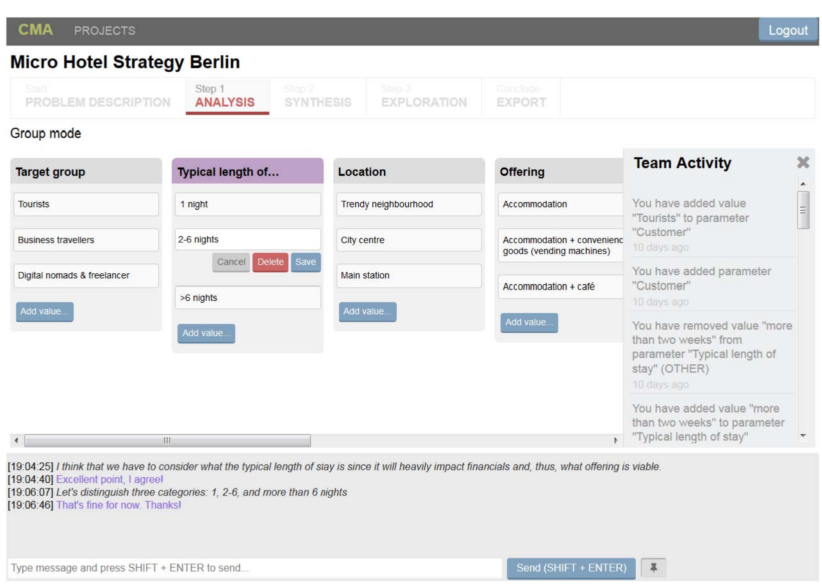


Fig. 7. In the group mode analysis view, users can see the parameters and values provided by other group members (indicated by purple color). A group chat and activity feed support group communication and audit trail. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

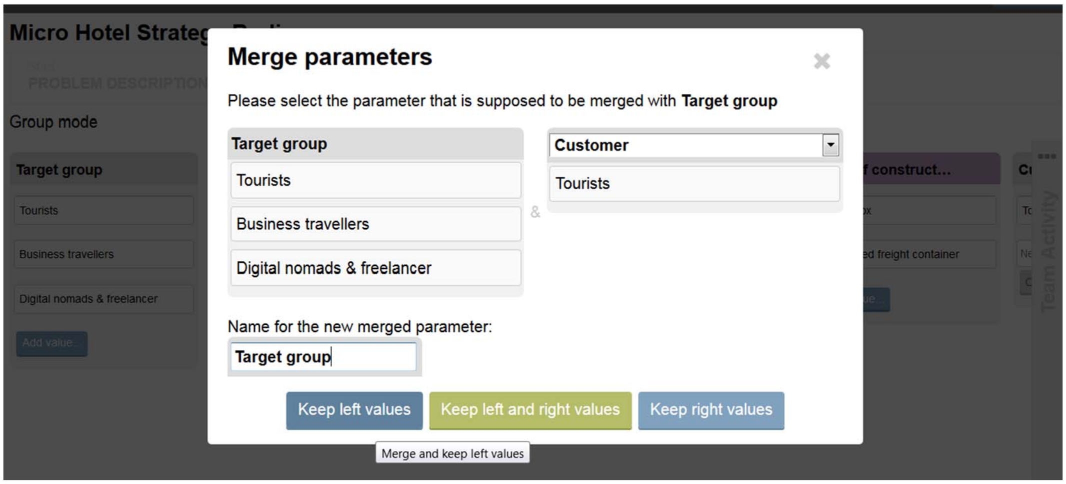


Fig. 8. Parameters can be merged if users decide to group certain parameters together.

Munich. We conducted six experimental sessions during which up to 24 subjects were assigned randomly into anonymous groups of three. All groups were briefly introduced to GMA using handouts and instructed to solve a simplified scenario/strategy development task using GMA (i.e. developing a micro hotel business concept for the city of Berlin). Participants were instructed to base their GMA on the information pieces they would receive over the course of the experiment. To create information asymmetry, each group member received a slightly different variant of the info sheet. While most of the information pieces were provided on all three sheets, some individual sheets contained exclusive or contrary information compared to the other sheets. Thus, effective collaboration and resolution of disagreements was required to identify good solution candidates. We collected data on subjective satisfaction with the group process, group outcome and the usability of CMA software. In this paper, we report on the usability of the aspects.

6.2. Participants

There were 114 participants (42 female, 72 male) between 19 and 33 years of age (mean 23.2, SD 3.14). 109 of the participants were students (68 undergraduate students, 41 graduate students), two were professionals, two were unemployed and one was self-employed. Most participants did not have any experience with GMA. Five subjects reported that they had applied GMA up to three times. Participants received a fixed compensation of 24 EUR for participating in the experiment.

6.3. Apparatus

Each participant was assigned to one of 24 identical personal workstations running CMA in a web browser in “kiosk mode” (i.e. participants were not able to run other software or terminate CMA). Users consented to the anonymous recording of their user actions within CMA. Each user was presented mandatory questionnaires at specific milestones during the experiment (see Section 6.4).

6.4. Procedure

At the outset of the experiment, students received a one-page handout providing a brief introduction to GMA using a table design example. The experiment comprised three sequential phases for analysis, synthesis and exploration (Table 3). At the beginning of each phase, participants received instructions and task-based info sheets for the corresponding phase. In the first phase (analysis), participants were asked to create the Zwicky Box based on a provided list of parameters. The values for each parameter had to be extracted from the info sheets. After 15 min, the analysis phase was concluded. In the second phase (synthesis), a blueprint Zwicky Box was presented to provide the same

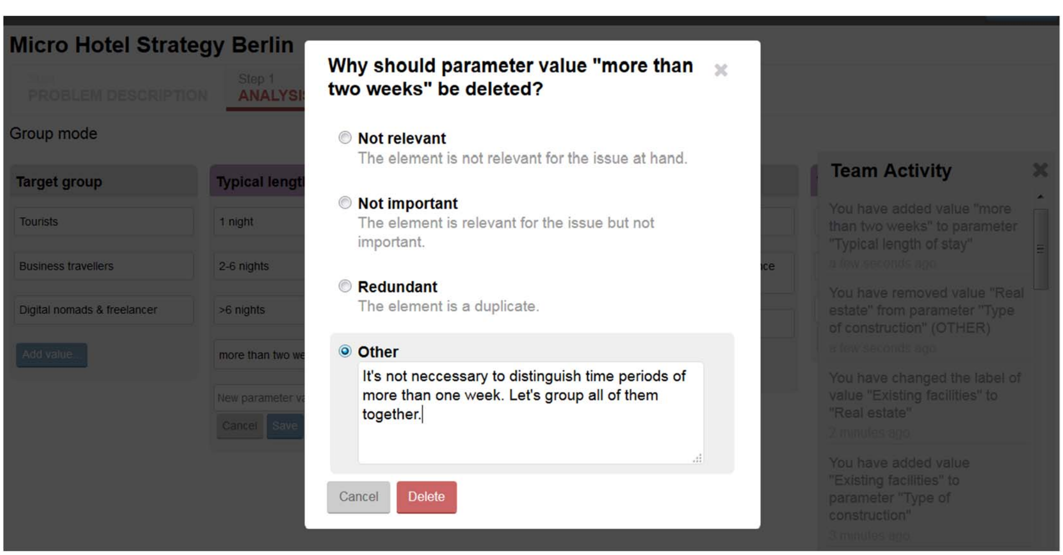


Fig. 9. Users may provide a justification for deleting parameters or parameter values.

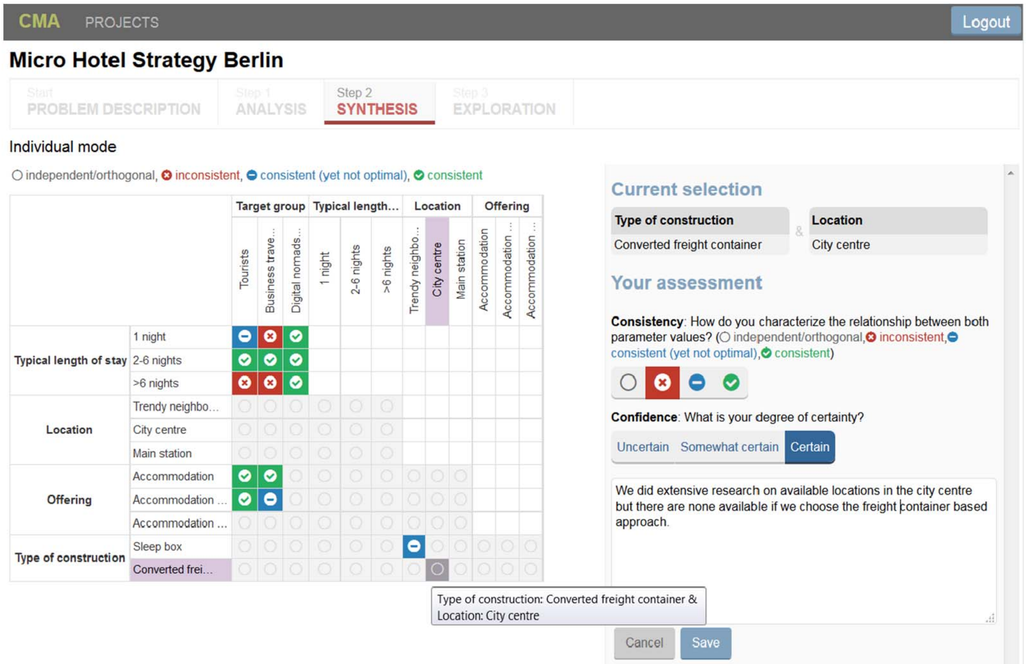


Fig. 10. Individual mode synthesis view: users can only see and manipulate their own cross-consistency matrix.

starting point for all groups. Participants received instructions on how to perform consistency assessments and an info sheet providing information on pairwise consistency. Based on this information, participants conducted cross-consistency assessments for 30 min. Finally, in the exploration phase, students were asked to discuss and identify good solution candidates based on the CMA inference model. It is important to note that participants collaborated anonymously (as in the Delphi method) and were instructed to use the chat feature of the software for general group communication.

All participants were asked to complete a questionnaire on their perception of usability. We employed an extended version of the system usability scale (SUS) (Brooke, 1996). The SUS was designed to measure perceived ease-of-use. Participants were presented with five positively-worded and five negatively-worded items and asked to express their level of agreement to each item using a five-point scale from 1 (strongly disagree) to 5 (strongly agree). To get a composite measure of the

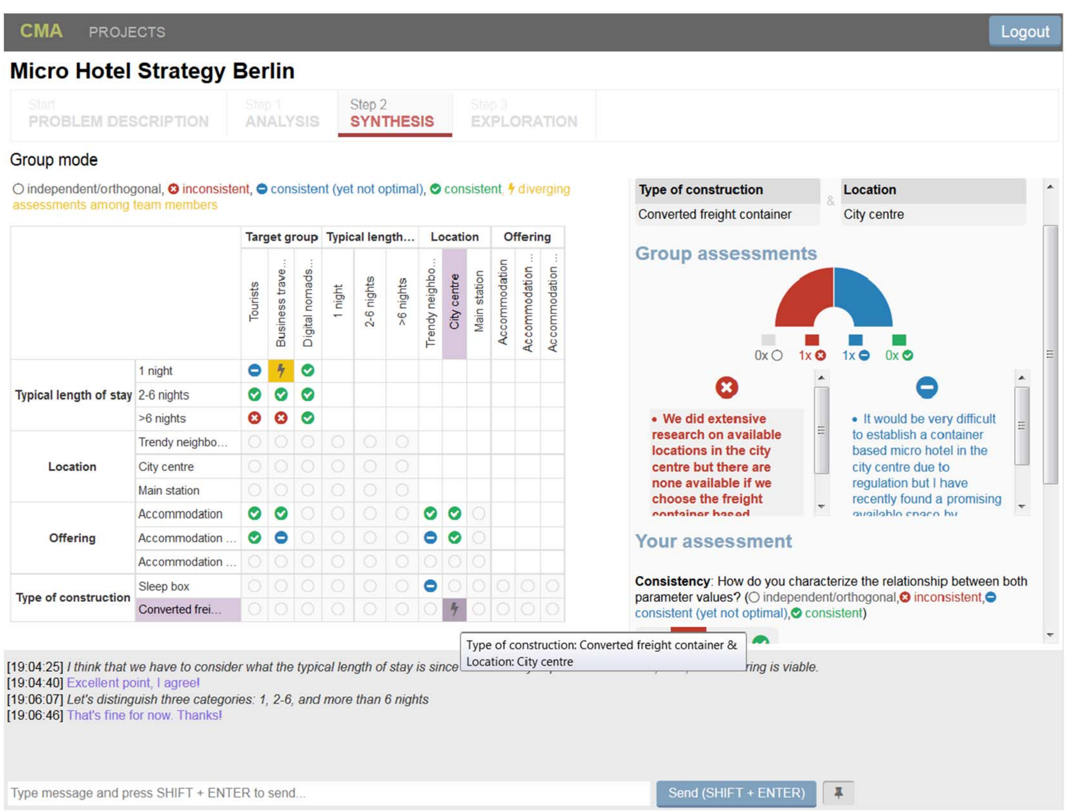


Fig. 11. In the group mode analysis view, users can see the consistency assessments provided by other group members. The group chat facilitates group communication.

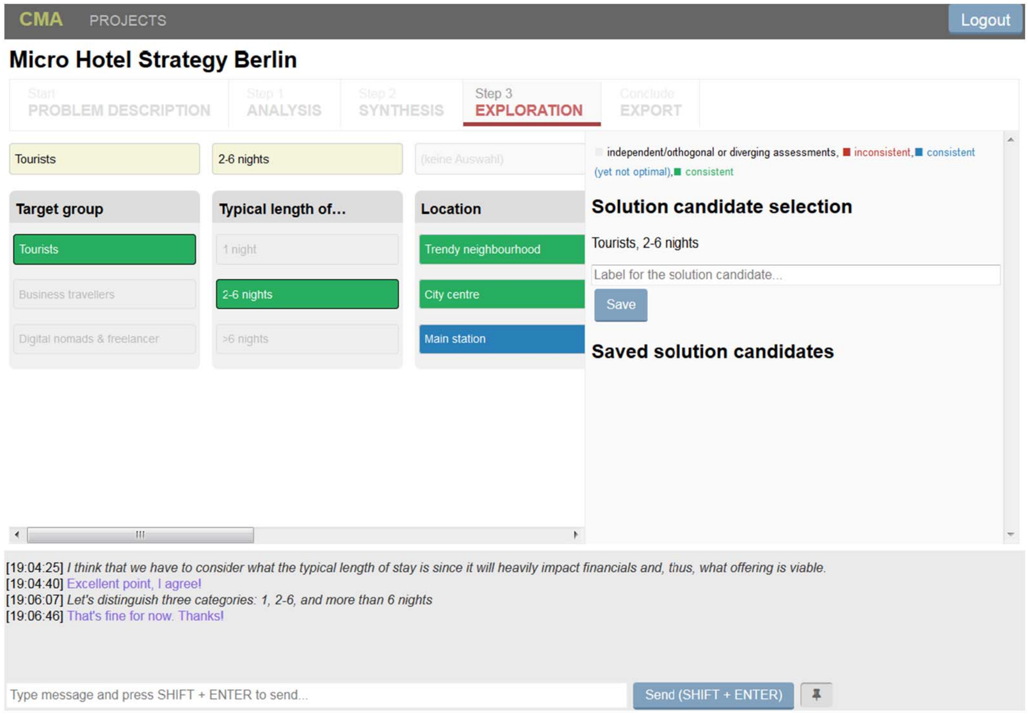


Fig. 12. The exploration view features a what-if inference model similar to MA/Carma.

Table 3
Experimental procedure.

Sequence no.		Duration
1	Reception	5 min
2	Questionnaire on demographics and personality	~5 min
3	Brief introduction to GMA	~ 7 min
4	Analysis	15 min
5	Questionnaire on analysis phase	~5 min
6	Synthesis	30 min
7	Questionnaire on synthesis phase	~5 min
8	Exploration	10 min
9	Questionnaire on exploration phase and usability	~10 min

overall usability, the ten single items need to be combined into a numerical SUS score ranging between 0 and 100 (see Brooke, 1996 for details). To make interpretation of the SUS scores easier, we added a seven-point adjective scale as an eleventh item to the questionnaire as suggested by Bangor et al. (2009). Participants were asked to assign one of the following adjectives to describe the perceived usability of the artefact: worst imaginable, awful, poor, ok, good, excellent, best imaginable.

In addition to the SUS and adjective rating scale, participants were invited to suggest ideas for improvement.

7. Results

Fig. 13 shows the distribution of the responses for each item of the SUS. In general, for all positively-worded items (see first five items in Fig. 13), a majority of users expressed that they agree or strongly agree to the respective statement. Conversely, a majority of users expressed disagreement or strong disagreement with negatively-worded statements (see last five items in Fig. 13).

74% of participants rated the software as easy to use ("I thought the system was easy to use") and 69% indicated that they felt confident using the system ("I felt very confident using the system"). The various functions of the system were generally found to be well integrated (71%). In terms of learnability, the majority of participants (84% of the users) disagreed with the statement that they would need a technical person to use the system and 81% disagreed that they needed to learn a lot of things before they could get going with the system.

The resulting mean SUS score is 70.7 (SD 16.1; see Fig. 14 for a boxplot of the overall SUS score and its subscales usability and learnability). However, interpreting SUS scores is difficult since they

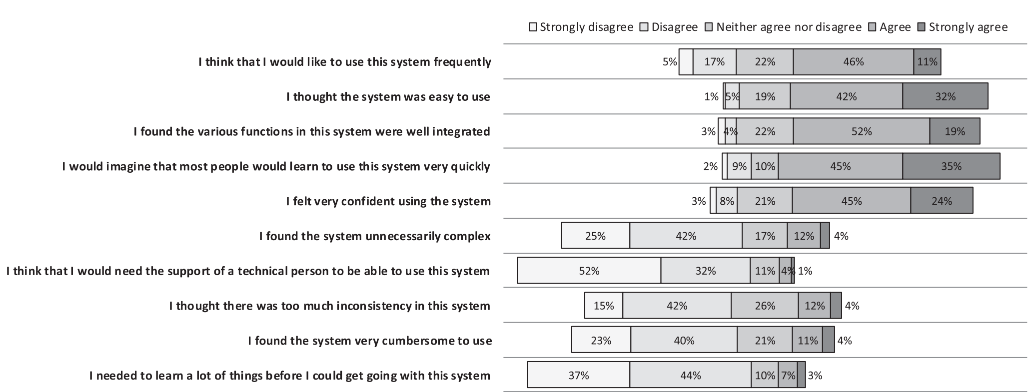


Fig. 13. The distribution of participants' responses to the ten SUS items.

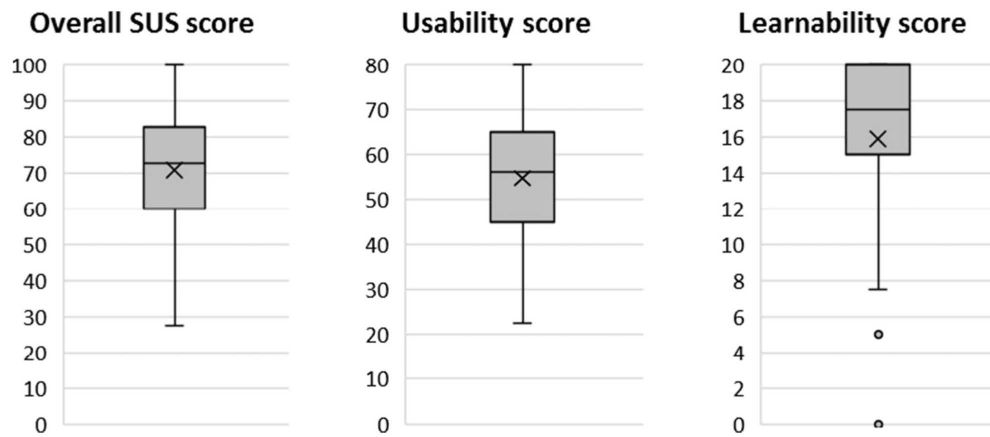


Fig. 14. The distribution of overall SUS (left; mean 70.7, SD 16.1, Q1 60.0, Q3 82.5), usability subscale (center; mean 54.8, SD 13.7, Q1 45.0, Q3 65.0), and learnability subscale (right; mean 15.9, SD 4.0, Q1 15.0, Q3 20.0) scores.

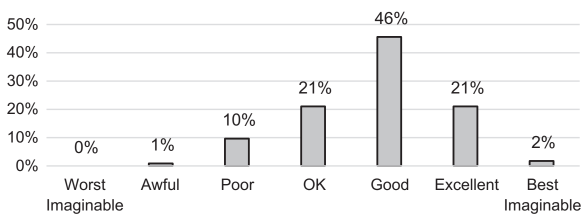


Fig. 15. The distribution of adjective ratings.

8. Discussion

The SUS was originally designed as a one-dimensional measure for usability. However, factor analyses have suggested that the SUS can be interpreted as containing two orthogonal components[1]: usability and learnability (Borsci et al., 2009; Lewis and Sauro, 2009). Overall, study participants reported positive feedback concerning the usability (54.8/80) and learnability (15.9/20) of the system. The majority of the study participants agreed that the software was easy to use. Almost two of three users indicated that they would like to use the software frequently. Most participants have indicated that they did not need to learn new things or require the help of a technical person to use the software. In addition to insights on usability and learnability, we did receive various suggestions for improvement. While many proposals refer to minor aspects or issues, we identified four major areas for improvement: group communication, process guidance, group coordination and solution space exploration.

Table 4
Recurring areas for improvement of the software and/or group process according to the participants.

Idea no.	Description	Number of mentions
1	Improve group communication Provide means for oral communication (e.g. video chat) and/or improve the chat UI (e.g. simplify shortcut for sending messages)	12
2	Provide more process guidance Provide a help function and hints and/or more examples on how to conduct the GMA and use the software	12
3	Improve group coordination Provide visual cues to increase awareness (e.g. highlight items fellow members are currently working on) and features to coordinate group activity (e.g. shared cursor to direct the attention of all group members to a specific element, introduce a group voting mechanism to let the group decide whether changes by individual group members should be accepted or declined)	11
4	Provide automatic recommendations in the exploration stage The system should automatically generate promising solution candidates in the exploration stage based on the consistency matrix	4

do not represent percentages. Rather, SUS scores should be interpreted on the basis of their percentile rankings (see, e.g., Brooke, 2013). However, the additional adjective scale we introduced as suggested by Bangor et al. (2009), provides a straightforward interpretation of the usability: 69% of the participants rated the system's usability as “good” (46%), “excellent” (21%) or “best imaginable” (2%). Still, 21% of the users found the usability to be “OK”. 10% of the users rated the usability as “poor” (Fig. 15).

Participants provided various ideas for improvement. Recurring themes in the submitted pool of ideas are listed in Table 4 (i.e. similar suggestions that were suggested by three or more participants). We have removed proposals relating to aspects not associated with technical limitations of the software but rather based on the constraints imposed by the experimental design. For instance, participants were not able to switch between the views even though the software technically allows users to do so. However, to align the experimental conditions for all participants, this feature was deactivated and the transition through the GMA process was time-triggered (Table 3). Consequently, proposals to introduce software features that are actually already implemented such as “Add possibility to switch views” were removed from the list.

In terms of group communication, several participants would prefer oral over written communication. In real-world projects, complementary tools for group communication such as video chat software might be used. However, facilitators should make sure that group members do not interact during individual sub-phases in order to not contravene the very point of the individual mode, i.e. to work independently without influencing and/or distracting each other. Another suggestion for improvement brought forward is process guidance. Some participants would appreciate a little more help, hints and examples to perform GMA more efficiently and effectively. In real-world projects, novice

[1] The learnability subscale includes items “I think that I would need the support of a technical person to be able to use this system” and “I needed to learn a lot of things before I could get going with this system.”. The remaining items constitute the usability subscale.

users should receive a thorough introduction to GMA. Future versions of the CMA software might feature example projects and additional tooltips to support novice users. Another major theme was group coordination. Currently, the software provides only limited (real-time) awareness information in the synthesis and exploration phases (i.e. information about the activity of fellow team members). As a result, some users considered the group process to be rather uncoordinated. The fourth category of proposals is concerned with software-support in the exploration phase. The suggestions range from minor improvements such as allowing users to modify and rename saved configurations to automatically providing a list of promising solution candidates.

9. Conclusion

This research set out to investigate how an IT artefact could support teams in conducting collaborative GMA modelling in distributed settings. To address the real-world problem of the Goethe Institute (GI), we conducted an ADR project to develop a web-based software tool for collaborative GMA, since existing GMA software is primarily designed for collocated face-to-face meetings. Within the scope of this research project, we focused primarily on software-support for distributed, collaborative GMA. In light of the feedback we received from GI, MOOC participants and experimental subjects, we consider the developed software to be a useful IT artefact for distributed, collaborative GMA. However, we acknowledge that effective facilitation and process guidance are crucially important for applying GMA successfully—maybe even more so in distributed settings compared to face-to-face settings. We consider our research to be a first step of making GMA a more viable problem-structuring method in distributed team settings and propose two major areas for future work.

First, there are various opportunities to extend the current version of the software such as adding support for a wider range of artefacts types. For instance, the problem statement view currently only displays plain text but could be extended to support images, videos, audio or other types of media. In the synthesis stage, the software could provide different choices of consistency/compatibility scales. The exploration step could be supported with additional types of representations of the solution space. We have received some suggestions for improvement from study participants as well (for instance, adding visual cues to indicate which user is working on a particular element or providing a voting mechanism for users to agree on whether a particular parameter or value should be deleted).

Second, we see a great need for research in the facilitation of virtual GMA sessions. The geographical distribution of team members poses great challenges to the facilitator. In this paper, we focused on questions around adequate GMA software for virtual teams. However, we acknowledge the importance of research on implications for the facilitation of online GMA projects both in synchronous as well as asynchronous settings. Collaboration Engineering (e.g. Briggs et al., 2003; Kolfschoten et al., 2006) is a promising methodological foundation for further research on facilitation.

In this paper, we have shown some of the implications for the design of collaborative, multi-user GMA software and presented one possible approach. We hope that the perspective on GMA research in conjunction with practice will be broadened to include increasingly broader, alternative forms of collaboration other than face-to-face meetings. We believe that a major common task of GMA researchers and practitioners is to develop methods and software tools for these new GMA application context, so that as many teams as possible can utilize the full potential of this problem-structuring method to tackle wicked problems.

Acknowledgements

This research was sponsored in part by the German Federal Ministry of Education and Research (BMBF) (grant number 01IS12057, project “Software Campus (TU München)”).

References

Arnold, C.R.B., Stone, R.B., McAdams, D.A., 2017. MEMIC: An interactive morphological matrix tool for automated concept generation. In: Proceedings of the 2008 Industrial Engineering Research Conference.
Bangor, A., Kortum, P.T., Miller, J.T., 2009. Determining what individual SUS scores mean: adding an adjective rating scale. J. Usability Stud. 4, 114–123.
Borsci, S., Federici, S., Lauriola, M., 2009. On the dimensionality of the system usability scale: a test of alternative measurement models. Cogn. Process. 10, 193–197.
van den Bossche, P., Gijselaers, W., Segers, M., Woltjer, G., Kirschner, P., 2011. Team learning: building shared mental models. Instr. Sci. 39, 283–301.
Bourse, F., Godet, M., 2016. Scenaring tools. http://en.laprospective.fr/methods-of-prospective/softwares—cloud-version/5-scenaring-tools.html (Accessed 14.11.2016).
Briggs, R.O., Vreede, G.-J. de, Nunamaker, J.F., 2003. Collaboration engineering with ThinkLets to pursue sustained success with group support systems. J. Manag. Inf. Syst. 19, 31–64.
Brooke, J., 1996. SUS: A ‘quick and dirty’ usability scale. In: Jordan, P.W., Thomas, B., Weerdmeester, B.A., McCleelland, I.L. (Eds.), Usability Evaluation in Industry. Taylor & Francis, London, Bristol, PA, pp. 189–194.
Brooke, J., 2013. SUS: a retrospective. J. Usability Stud. 8, 29–40.
Childs, P., Garvey, B., 2015. Using morphological analysis to tackle uncertainty at the design phase for a safety critical application. Propuls. Power Res. 4, 1–8.
Conklin, E.J., 2006. Dialogue Mapping: Building Shared Understanding of Wicked Problems. Wiley, Chichester, England, Hoboken, NJ.
Coyle, G., 2004. Practical Strategy Structured Tools and Techniques. Pearson Education, Harlow.
Dalkey, N., Helmer, O., 1963. An experimental application of the DELPHI method to the use of experts. Manag. Sci. 9, 458–467.
Delbecq, A.L., 1971. A group process model for problem identification and program planning. J. Appl. Behav. Sci. 7, 466–492.
Diehl, M., Stroebe, W., 1991. Productivity loss in idea-generating groups: tracking down the blocking effect. J. Pers. Soc. Psychol. 61, 392–403.
Durnell Cramton, C., Hinds, P.J., 2004. Subgroup dynamics in internationally distributed teams: ethnocentrism or cross-national learning? Res. Organ. Behav. 26, 231–263.
Ferrazzi, K., 2014. Getting virtual teams right. Harv. Bus. Rev. 92, 120–123.
Forsyth, D.R., 2014. Group Dynamics, sixth ed. Wadsworth Cengage Learning, Belmont, CA.
Future-Oriented Integrated Management of European Forest Landscapes, 2014. https://forestwiki.jrc.ec.europa.eu/integral/images/e/e6/Consistency_scenariospace_southeastveluwe.jpg (Accessed 15.11.2016).
Godet, M., 2006. Strategic foresight: La prospective. In: Problems and Methods. LIPSOR Working Paper. Paris.
Introne, J., Laubacher, R., Olson, G., Malone, T., 2013. Solving wicked social problems with socio-computational systems. Künstl. Intell. 27, 45–52.
Kelman, H.C., 1958. Compliance, identification, and internalization three processes of attitude change. J. Confl. Resolut. 2, 51–60.
Kolfschoten, G.L., Briggs, R.O., Vreede, G.-J. de, Jacobs, P.H., Appelman, J.H., 2006. A conceptual foundation of the thinkLet concept for collaboration engineering. Int. J. Hum. Comput. Stud. 64, 611–621.
Lewis, J.R., Sauro, J., 2009. The factor structure of the system usability scale. In: Kurosu, M. (Ed.), Human Centered Design: First International Conference, HCD 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19–24, 2009; Proceedings. Springer, Berlin, pp. 94–103.
Martins, L.L., Gilson, L.L., Maynard, M.T., 2004. Virtual teams: what do we know and where do we go from here? J. Manag. 30, 805–835.
Nunamaker, J.F., Dennis, A.R., Valacich, J.S., Vogel, D., George, J.F., 1991. Electronic meeting systems. Commun. ACM 34, 40–61.
Parmenides, A.G., 2016. Parmenides EIDOS website. https://www.parmenides-eidos.com/ (Accessed 14.11.2016).
Ritchey, T., 2006. Problem structuring using computer-aided morphological analysis. J. Oper. Res. Soc. 57, 792–801.
Ritchey, T., 2011. Wicked Problems—Social Messes. Springer, Berlin, Heidelberg (Berlin Heidelberg).
Ritchey, T., 2016. MA/Carma. http://www.swemorph.com/macarma.html (Accessed 14.11.2016).
Rittel, H.W.J., Webber, M.M., 1973. Dilemmas in a general theory of planning. Policy. Sci. 4, 155–169.
Schoder, D., Putzke, J., Metaxas, P.T., Gloor, P.A., Fischbach, K., 2014. Information systems for “wicked problems”. Bus. Inf. Syst. Eng. 6, 3–10.
Sein, M.K., Henfridsson, O., Purao, S., Rossi, M., Lindgren, R., 2011. Action design research. MIS Q. 35, 37–56.
Voros, J., 2009. Morphological prospection: profiling the shapes of things to come. Foresight 11, 4–20.
Zec, M., Schneider, A.W., Matthes, F., 2015. Towards a process model for computer-supported collaborative morphological analysis. In: Proceedings of the Twenty-First Americas Conference on Information Systems.
Zwicky, F., 1969. Discovery, Invention, Research through the Morphological Approach, 1st American ed. Macmillan, New York.
Zwicky, F., 1971. Entdecken, Erfinden, Forschen in morphologischen Weltbild. In: Knaur, 264. Droemer Knaur, München.

Marin Zec is a research and teaching associate at the chair for Software Engineering for Business Information Systems at TU München. Previously, he has spent one year as a visiting student and research assistant at MIT in Cambridge, USA. He received a B.Sc.

degree in Computer Science (2009) from TU München and a M.Sc. degree in Software Engineering (2012) from TU München, LMU München and the University of Augsburg. He is interested in knowledge work, collaboration and creative problem solving.

Florian Matthes holds the chair for Software Engineering for Business Information Systems at TU München. His current research focusses on technologies driving the digital transformation of enterprises and societies: Enterprise architecture management, social content and model management, and semantic modelling of legal texts. He is co-founder and chairman of CoreMedia (1996) and infoAsset (1999), co-founder of further small software and service university spin-offs, and scientific advisor of UnternehmerTUM, the center of innovation and business creation at TU München. Earlier stations of his academic career are the Goethe-University Frankfurt (Diploma 1988), the University of Hamburg (PhD 1992), the Digital Systems Research Center (now HP SRC Classic) in Palo Alto, USA (Researcher 1992–1993), and the Technical University Hamburg-Harburg (Associate Professor 1997–2002).