Towards a Process Model for Computer-Supported Collaborative Morphological Analysis

Completed Research Paper

Marin Zec
Technische Universität München
marin.zec@tum.de

Alexander W. Schneider
Technische Universität München
alexander.schneider@tum.de

Florian Matthes
Technische Universität München
matthes@in.tum.de

Abstract

Morphological analysis (MA) is a method to analyze and design complex systems. MA fosters generation of a large number of solution/system design alternatives, yet it requires a considerable amount of manual effort. Therefore, a number of software tools have been developed to automate the construction of consistent design alternatives and support the exploration of the derived design space. However, available tools lack support for collaboration. This inhibits extended use of MA software in practice since system analysis and design tasks are typically conducted by teams. The purpose of this paper is to enhance the classical MA process model by collaboration support. We review seminal psychological research to guide our process design. We propose a collaborative process model that seeks to increase creativity and prevent psychological phenomena which might impair group performance. The revised process model serves as a basis for collaborative MA software implementations.

Keywords

Wicked Problems, Morphological Analysis, Systems Engineering, Systems Analysis, Systems Design, Problem Structuring, Problem Modelling

Introduction

Systems analysis and design (SA&D) aims at better understanding and design of complex natural, technical or social systems. Morphological analysis (MA) – pioneered by Fritz Zwicky (Zwicky 1969) – stands out as a simple and generic, yet structured and powerful approach for SA&D since it focuses on two fundamental concepts in systems engineering: analysis (i.e. decomposition) and synthesis (Jimenez and Mavris 2010). MA has been successfully applied in various domains such as policy planning (Ritchey 2006), strategic foresight (Voros 2009) and technology selection (Jimenez and Mavris 2010). Since MA might quickly yield a large number of solution/system candidates, software has been developed to make MA more manageable and flexible because the morphological model of the problem/system of interest can be easily stored, refined and shared.

Since the cognitive capacity of individuals is bounded and today’s knowledge workers tend to be specialized in a particular discipline, SA&D and, particularly, MA is usually conducted in teamwork. Teamwork is expected to yield better system models and design decisions by integrating the perspectives of diverse team members and leveraging the collective intelligence. However, a large body of literature in social as well as cognitive psychology has evolved over the last decades, indicating that teams are prone to psychological phenomena which impair team performance and might lead to serious misjudgments. Trained facilitators support teams in avoiding group process losses and increase process gains. However, particularly in large organizations, facilitators might easily become a bottleneck for effective collaboration. Then, some SA&D problems occur on short notice or need to be handled by a distributed,
virtual team. While available MA software does support the general steps involved in MA, it has a major drawback: it only provides a single-user interface. Thus, groups tend to use a projector as a workaround in face to face workshops. However, still only one person (e.g. facilitator) can actually manipulate the model, likely causing production blocking every time the group has to wait for the completion of his/her manual model updates. In addition, distributed teams cannot use available MA software in a collaborative manner due to the separated local models. To address this lack, next generation MA software should be designed in a way that supports collaboration by design.

In this paper, we propose an extension of the classical process model for computer-supported morphological analysis based on results from social and cognitive psychology research. The proposed process model accounts for the collaborative nature that is particularly prevalent in SA&D. It seeks to increase creativity and avoid certain psychological effects which negatively affect group performance. We suggest the development of collaborative morphological analysis (CMA) software implementing this process model so that teams can conduct MA collaboratively. CMA software should facilitate ideation and help preventing known potential process losses in teamwork.

Morphological Analysis

MA is a versatile approach for SA&D. They key idea of MA is to focus on the study of the form (morphology) of the system. As a result, MA fosters “contingency thinking” (Voros 2009): all components of the system are considered to be contingent rather than fixed. Thus, MA encourages openness for alternatives in SA&D. MA is best suited for problems/systems of interest which cannot be adequately expressed using quantitative models and thus methods and software for mathematical optimization or simulation are nonapplicable (e.g. objective function and mathematical programming) – a characteristic that is often shared by “wicked problems” (Rittel and Webber 1973). Originally, Zwicky proposed a five-step process (Zwicky 1966). However, at its core, MA can be described as an iterative approach involving three major stages: (1) Analysis, (2) Synthesis, and (3) Exploration (see Figure 1). MA is based on two artifacts: the (1) morphological matrix captures the (intermediate) state of the analysis while the (2) cross-consistency matrix captures judgments and decisions made during the synthesis stage. MA software typically provides additional artifacts for the exploration stage such as a list of solution candidates ranked by consistency, an interactive inference model or graphical representations of the solution/design space.

**Stage 1: Analysis**

In this stage, the system of interest is decomposed. First, all relevant components (parameters) are identified. Then, for each component a discrete range of mutually exclusive (solution) alternatives (values) is defined. At this point, even seemingly implausible or potentially unrealistic values should be included since they might provide an inspiration for creative solutions or designs. Finally, a morphological matrix is constructed which captures all parameters and their values (see Figure 2). The morphological matrix is a compact representation of the formal design space. A formal solution or system alternative (configuration) is given by selecting one particular value for each parameter (e.g. configuration \{A2, B3, C1, D4\} in Figure 2).
Stage 2: Synthesis

In the synthesis stage, components are reintegrated into consistent system configurations. The formal configuration space (i.e., morphological matrix) generally contains configurations which are internally inconsistent. Configurations containing a subset of incompatible parameter values are infeasible and can be excluded from further consideration. A cross-consistency matrix is generated to facilitate the generation of consistent configurations (see Figure 3). The matrix contains all pairwise consistency assessments of all values. Typically, some sort of ordinal scale is applied for the assessments. The consistency matrix is symmetric due to the symmetrical nature of the consistency relation. It allows to generate or check viable system configurations systematically. For instance, according to the matrix in Figure 3, configuration $c_1 = \{A2, B3, C2, D2\}$ is viable while $c_2 = \{A1, B4, C3, D1\}$ is not (because $A1$ is incompatible with $D1$). Sometimes a transition back to the analysis phase might be necessary to modify the morphological matrix according to new insights that came up in the synthesis stage.

Stage 3: Exploration

In the final stage of MA, the remaining set of internally consistent configurations (i.e., viable solution or system alternatives) is explored. The configurations are evaluated based on external criteria which depend on the specific context of the SA&D project (e.g., costs, time to market or ethical values). However, going back to an earlier stage of the MA process might be necessary to adjust the model to new learnings. Also, the results could be required as input for a subsequent MA. For instance, consistent configurations derived in the former MA process might represent values for a parameter of the later MA process in a (hierarchical) system of systems analysis/design.
Computer-Supported Collaborative Morphological Analysis

The morphological matrix derived in the first stage of MA represents the entire space of formal configurations. Since a significant amount of these configurations does not characterize plausible or realistic solutions/systems, the goal of the synthesis stage is to reduce the formal solution space in such a way that it only contains consistent configurations. However, the configuration space may contain a very large number of solution/system alternatives. For instance, a morphological matrix consisting of only 6 parameters (or components) and 5 possible alternatives for each parameter yields $5^6 = 15,625$ distinct configurations. A thorough manual evaluation of all formal configurations is generally impracticable. In practice, teams select a limited set of configurations on the basis of heuristics.

To address this issue, various software tools have been developed to support MA such as MA/Carma (Swedish Morphological Society 2015) or Parmenides Eidos (Parmenides Foundation 2015). Instead of inspecting each formal configuration as a whole, both tools make use of cross-consistency assessment (CCA). The goal of the CCA approach is to identify inconsistencies in any pair of parameter values. Then, the MA software excludes all configurations that contain at least one pair of inconsistent values from the configuration space leaving only plausible configurations for consideration. Using CCA, only 375 pairs of parameter values instead of all 15,625 distinct configurations have to be examined in the example above.

While there is still some effort required to conduct CCA, it offers three important advantages over manual, ad-hoc configuration selection. First, modifying the set of parameters while iterating over the MA stages does not require a full reexamination of each configuration’s consistency (as in the case of manual MA) but at most providing consistency assessments of possibly new parameter value pairs. Second, while manual MA typically relies on some heuristic and time constraints for consistent configuration selection which might result in overlooking promising consistent configurations, MA software can almost instantly filter out all inconsistent configurations. Third, MA software can generate useful visualizations and artifacts that support human reasoning and decision-making over configurations. For instance, MA/Carma provides an interactive inference model for “what-if” simulations. The user can select arbitrary parameter values as fixed input. Then, the software displays which remaining parameter values might be selected so that the selected (partial) configuration remains consistent. By this means, the user is able to dynamically explore the impact of particular parameter value choices (see Figure 4). Another useful visualizations is provided by Parmenides Eidos (see Figure 5): the multidimensional configuration space on a two-dimensional map.

**Figure 4.** An interactive inference model provided by MA/Carma (adopted from Swedish Morphological Society 2015). Users can select values as fixed inputs (red). MA/Carma displays the remaining parameter value which are consistent with the fixed inputs.

**Figure 5.** Parmenides Eidos projects the multidimensional configuration space on a two-dimensional map. Each circle represents one configuration. Spatial proximity correlates with similarity of the respective configurations.

space is projected on a two-dimensional plane while similarity between configurations is mapped to spatial proximity or distance, respectively. This approach enables users to identify similar configurations in a visual manner and, consequently, further reduce the number of solution/system alternatives to be considered by focusing on configuration clusters instead of single configurations. Users can choose what level of detail is appropriate for their solution/system design in their particular context. Sometimes discussing solution/system alternatives on a more general level (i.e. configuration clusters) might be sufficient while at other times distinguishing between concrete configurations might be necessary. In both cases, the visual clustering of configurations enables users to coarsely filter out clusters of irrelevant configurations.

MA software provides significant support concerning synthesis (by automating the generation of viable configurations) and exploration (by offering graphical or interactive representations of the consistent configuration space). However, available tools are not designed to be used in a collaborative manner. Since many SA&D projects are conducted in teamwork, we argue that next-generation MA software needs to support multiple concurrent users and provide an adequate extended process model for MA which incorporates established patterns and strategies to improve group performance and creativity.

The next section introduces findings from social and cognitive psychology which are relevant to group processes. They will motivate design decisions for our proposed process model for CMA.

**Process Losses in Groups**

Analysis and design of complex systems requires a diverse set of cognitive resources that frequently exceed the capabilities of individuals. In those situations, SA&D is conducted in teamwork. Teams are expected to perform better than individuals since they can create a transactive memory system (Wegener et al. 1985): i.e. distribute the encoding, storage and retrieval of information among team members. However, a large body of psychological research has shown that groups are prone to psychological phenomena which impair group performance and decision-making.

A number of studies on group performance in brainstorming have shown that nominal (i.e. non-interacting) groups tend to outperform real, interactive groups (e.g. Mullen et al. 1991). Explanations include social loafing, production blocking and evaluation apprehension.

**Production Blocking**

Production blocking refers to the process loss that occurs when group members have to take turns in expressing their thoughts and ideas (Diehl and Stroebe 1987). As a result, silent group members self-censor or forget their ideas. Diehl et al. have shown that the productivity loss is not a mere effect of available time (Diehl and Stroebe 1991).

**Social Loafing**

If individual effort and performance cannot be identified, individuals are prone to reduce their effort when working in a group compared to working alone (Karau and Williams 1993). The effect is commonly explained with the individual perception that his/her contributions do not have a significant impact on group performance. For instance, less trained or skilled group members might be intimidated by high performing individuals and, consequently, reduce their effort. Moreover, high performing individuals might feel inclined to reduce their effort if they consider other team members to be free riders.

**Social Inhibition**

A number of studies have found that the mere presence of others tends to impair an individual’s complex task performance (Bond and Titus 1983). Social inhibition and facilitation have been explained with physiological, motivational, cognitive and personality processes (Guerin and Innes 1984; Zajonc 1965; Camacho and Paulus 1995, Uziel 2007). The major explanations for social inhibition share the assumption that the presence of others leads to elevated arousal which increases the probability that the individual performs dominant responses (i.e. behavior that has been exercised habitually in reaction to a stimulus).
For simple or well-trained tasks, the dominant response tends to be the appropriate reaction. However, for complex tasks, the dominant response tends not to lead to a correct response.

**Group Polarization**

Group polarization refers to the phenomenon that groups tend to enhance the initial inclinations of their group members (for an overview, see Myers 2010). Proposed explanations for this effect include persuasion, social comparison and differentiation. Prior to group discussion, an individual has a set of arguments favoring his/her attitude stance. However, individuals tend to maintain various arguments for a particular opinion. Thus, it is likely that group members will learn additional arguments for their initial belief. Another aspect is the pressure to conform to norms that are socially desirable. A strategy to mitigate group polarization is to grant group members a certain period of time to work privately. Afterwards, all individual contributions are collected and discussed anonymously.

**Failure to Solve Hidden Profiles**

In group decision-making, a hidden profile describes a correct solution which is not identifiable on the basis of individual information since some partial information is shared and other relevant pieces of information are unshared (Stasser and Titus 1985). Individual group members cannot detect a hidden profile because they only have insufficient information. Only if relevant information is pooled effectively, group members might succeed in finding the hidden profile. If crucial information is not shared in hidden profile decision tasks, the result may be an incorrect decision.

**Common Knowledge Effect**

The common knowledge effect was first observed by (Gigone and Hastie 1993). They found that the influence of a piece of information is directly, positively related to the number of people who have knowledge of that item prior to discussion. Therefore, group members should be encouraged to express unique and unshared information since that information might be crucial for the correct decision/solution.

**Computer-Supported Collaborative Morphological Analysis**

The previous sections demonstrated the (1) major benefits of computer-supported over manual MA, (2) the lack of collaboration support in available MA software and (3) potential process losses in teamwork. In this section, we propose a modified process model for MA to account for the prevalent collaborative nature of SA&D tasks. First, we will analyze general collaboration patterns which are relevant for CMA. Then we will show how software might support these collaboration patterns in such a way that the process losses discussed above are avoided.

**Patterns of Collaboration in CMA**

Six general patterns can be observed in collaboration settings (Briggs et al. 2006): Generate, Reduce, Clarify, Organize, Evaluate and Build Consensus. The Generate pattern describes the adding of new items to the pool of concepts and ideas shared by the group. The Reduce pattern refers to the group activity of removing items which do not seem to be worthy of further attention. The Clarify pattern signifies group activities which aim to increase shared understanding of concepts and ideas. The group of activities which aim to increase the understanding of the relationship among concepts and ideas is captured by the Organize pattern. The Evaluate pattern refers to group activities which increase the understanding of the (relative) value of concepts and ideas. All group actions which increase the number of group members who are willing to commit to a proposal fall into the category of Build Consensus.

We reconstructed the general pattern sequences in CMA as follows.
Analysis Stage

The general sequence of collaboration patterns within the analysis stage is:

1. **Generate**: parameters and values are created
2. **Clarify**: shared understanding of parameters and values is reached (if necessary)
3. **Organize**: parameters and values are modified/rearranged (if necessary)
4. **Reduce**: irrelevant parameters and values are dropped
5. **Build Consensus**: all team members commit to the joint morphological matrix

Synthesis Stage

This subtask of MA can be performed by software. For the sake of completeness, we provide the collaboration pattern sequence for manual generation of consistent configurations. The general pattern sequence within the synthesis stage is as follows:

1. **Consistency Assessment**
   1.1. **Evaluate**: pairwise consistency assessments are provided
   1.2. **Clarify**: evidence or arguments are presented (in case of conflicting assessments)
   1.3. **Build Consensus**: all team members commit to the joint consistency matrix

2. **Generation of Consistent Configurations**
   2.1. **Generate**: the desired number of internally consistent configurations is generated
   2.2. **Clarify**: possible questions are discussed and shared understanding is reached

Exploration Stage

The general pattern sequence within the exploration stage is as follows:

1. **Organize**: similar configurations are grouped together (if desired)
2. **Reduce**: configurations which are not deemed worthy of further examination are dropped
3. **Evaluate**: configurations are evaluated based on agreed or given criteria
4. **Build Consensus**: a shared commitment to the best solution/system design candidate is reached

In practice, new insights and arguments that come up during group discussion might require iterating over the stages multiple times until the matrices converge. For this purpose, it makes sense to assign a facilitator role which monitors and guides the process flow or use software which handles process control. Software for CMA should provide adequate support for each pattern within each stage. For instance, the **Generate** pattern should start with an individual phase during which each group member works on his/her own. CMA software should provide an intuitive way for group members to mark items which are supposed to be deleted (**Reduce**), merged or grouped together (**Organize**). An anonymous voting mechanism and an informal communication channel might help groups in deciding which configurations should actually be removed, merged or clustered (**Build Consensus**). CMA software can support the **Clarify** pattern by allowing group members to both mark elements which raise questions and provide evidence and arguments to address these questions. The **Evaluate** pattern can be supported by offering some sort of voting mechanism.
Introducing Individual Substeps

The major enhancement of the classical MA process is the subdivision of both the analysis and synthesis stage into two (possibly iterative) substeps to improve information sharing among groups. Figure 6 shows the extended process model.

In the first substep, each individual performs the respective task on his/her own to avoid social inhibition, production block and the common knowledge effect. A similar approach has been suggested for brainstorming (e.g. Gallupe et al. 1991). Only afterwards, in the second step, the individual morphological or consistency matrices, respectively, are shared, discussed and merged into a joint model by voting. In this way, unique and controversial ideas are more probable to be shared (Dugosh and Paulus 2005). Ensuring anonymity is important so that evaluation apprehension can be avoided. During analysis and synthesis multiple iterations between their substeps might be required until consensus is built. Only joint models (i.e. matrices) serve as input for subsequent stages.

The suggested process model for CMA is inspired by the more general Delphi technique (e.g. Dalkey and Helmer 1963). The Delphi technique is a series of sequential questionnaires and controlled feedback. The purpose of the Delphi technique is to structure group communication, tap individual expertise and judgment and, as a result, reach consensus among a group of experts. The proposed process model differs from the Delphi technique in at least two aspects. First, the CMA technique prescribes the form of central artefacts (i.e. morphological matrix and CCA matrix). Second, the CMA technique does not require each participant to provide an answer. The reasoning behind this design decision is that participants may not be knowledgeable or confident of their expertise in some domains. They should be able to defer or restrain from judgment if they choose to. This way, contributions occur on voluntary basis and the likelihood of biases that are caused by mere guessing or lack of relevant knowledge decreases.

Implications

The suggested enhancements of the classical MA process model are generally applicable both to manual as well computer-supported MA. While CMA software is not available to date, it could offer various benefits over manual MA such as:

Ensuring process adherence

Software can control the process flow and adjust the user interface in dependence of the current state of the process. For instance, only private contributions might be displayed in the individual substeps. While experienced facilitators can improve the process, CMA software can take over process management (e.g. scheduling of deadlines, sending out reminders, and collecting individual contributions).
Providing a higher degree of anonymity where necessary

Software can make sure that author information is hidden in the collaborative substeps. In this way, contributions are expected to be evaluated more objectively. At the same time, techniques are available to reduce social loafing (e.g. by measuring and displaying the activity level of each group member on an abstract level without disclosing authorship of particular contributions).

Support for distributed teams

If implemented accordingly (e.g. as a web application), software can offer real-time support for teams that are distributed in terms of location and/or time. The suggested process model can be applied both in real-time (e.g. for ad-hoc problems) as well as asynchronous sessions (e.g. for SA&D projects with a more long-term roadmap). Asynchronous sessions allow to leverage the incubation effect on creative problem solving (e.g. Sio and Ormerod 2009): Sometimes good ideas emerge only after a certain period of time (even past the face-to-face workshop session). Using a web-based CMA software, participants are able to contribute ideas and input at any time during the project period.

Support for conflict resolution

It is expected that some team members will disagree about the choice of parameters, parameter values and/or consistency values. CMA software can compare individual models and/or consistency assessments and, as a result, identify those disagreements. Then, the user interface could highlight the controversial assessments. The disagreeing parties could save time by concentrating on resolving disagreements instead of reassuring established agreements. The CMA software might ask the disagreeing parties to provide evidence or justifications for their choice and let the team vote for the final decision.

Support for consensus building

Software can support consensus building by providing simple voting facilities and means to merge matrices. In contrast to the Delphi method, participants do not have to contribute if they are not confident about particular issues. For instance, an engineer might only provide consistency assessments for technical value pairs. He is not required to provide a full consistency matrix. This way, unnecessary discussion or conflicts can be avoided. The only constraint for the CMA process is that there is at least one consistency assessment per value pair from at least one participant.

Conclusion

MA is a prevalent method for SA&D that has been used in various domains, particularly when the system of interest cannot be adequately expressed in a quantitative model (e.g. mathematical optimization) as in the case of wicked problems. Available MA software provides significant support for conducting MA such as automated generation of consistent configurations or interactive/visual representations of the resulting design space. SA&D is typically conducted in teamwork. We identified the lack of collaboration support in available MA software. To address this issue, we proposed adjustments to the classical MA process model which are based on evidence from psychological research and inspired by the Delphi technique. The revised process model can serve as a basis for CMA software implementations. The next step is to develop a prototypical software based on the proposed process model. Using the prototype, the suggested process model can be evaluated in empirical studies. Given the general validity of our approach, a plethora of arising research questions regarding CMA software have to be addressed such as:

- **General**
  - Which process flow control mechanisms should be supported?
  - How should the collaboration patterns within each stage be supported?
  - How does CMA software affect creativity/richness of models or satisfaction of the participants with the process and outcome?
• **Analysis/Synthesis stage**
  - Which matrix merging mechanisms should be supported?
  - Which conflict resolution mechanisms should be supported?
  - Which mechanisms for consensus building should be supported?

• **Exploration stage**
  - Which types of representations/visualizations provide significant support for the exploration of the consistent configuration space?

**Acknowledgments**

This research was sponsored in part by the German Federal Ministry of Education and Research (BMBF) (project “Software Campus (TU München)”, grant no. 01IS12057).

**REFERENCES**


