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

ARTICLE INFO

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

ABSTRACT

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.

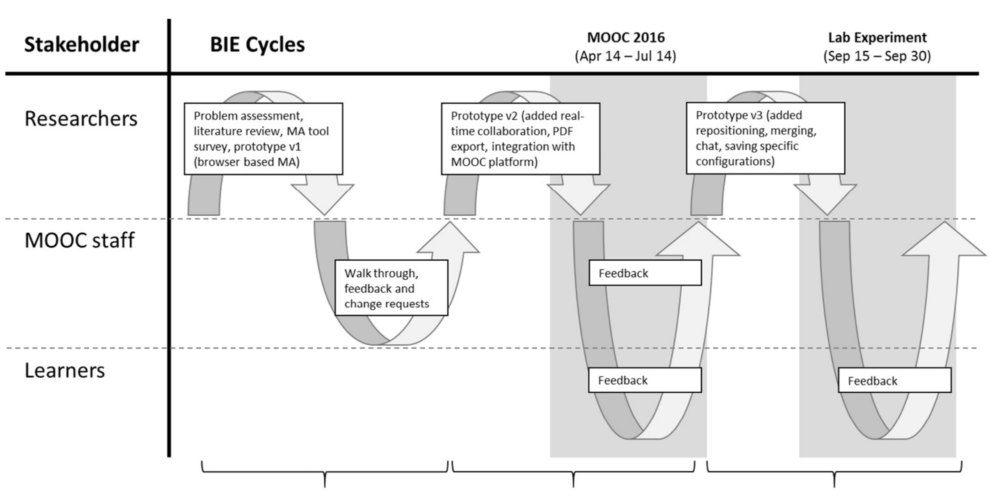

Fig. 1. An overview of the three BIE cycles performed during this research project.