DEVELOPING ARGUMENT ASSISTANT SYSTEMS FROM A USABILITY VIEWPOINT

María Paula González, Carlos Iván Chesñevar
CONICET / Department of Computer Scicence & Engineering, Universidad Nacional del Sur, Bahía Blanca, Argentina

Niels Pinkwart
Department of Informatics, Clausthal University of Technology, Clausthal-Zellerfeld, Germany

Mauro J. Gómez Lucero
CONICET / Department of Computer Scicence & Engineering, Universidad Nacional del Sur, Bahía Blanca, Argentina

Keywords: Intelligent information systems, Tools for knowledge management, Argument assistant systems, Usability.

Abstract: This paper discusses the role of usability as a quality key attribute for the deployment of Argument Assistant Systems, which are software tools intended to provide effective knowledge management facilities when solving problems in different contexts, helping to identify, create, represent and analyze the arguments involved as well as their interrelationships. Based on a reverse engineering process, a set of usability-oriented Design Guidelines were identified and instantiated for the Argument Assistant System domain. Besides, some usability principles are proposed and linked to every suggested guideline to evaluate its quality in use.

1 INTRODUCTION

Argumentation is an important aspect of human decision making. *Argumentation systems* (Rahwan and Simari, 2009) have been increasingly considered for applications in developing software tools, constituting an important component of multi-agent systems for negotiation, problem solving, and for the fusion of data and knowledge. Theoretical advances in the area have consolidated different computational models of argument, paving the way for the development of soft implementations, specially *Argument Assistance Systems* (AAS) (Verheij, 2003). AAS are intended to provide effective knowledge management facilities when solving problems in different contexts (legal reasoning (Verheij, 2007), organizational knowledge (Brena et al., 2007), CSCW (Scheuer et al., 2010), etc.).

Several AAS have been developed in the last years (e.g. Araucaria (Reed and Rowe, 2004), ArguMed (Verheij, 2003), Compendium (Kirschner et al., 2003), etc.). As pointed out in (Verheij, 2003), AAS are to be distinguished from fully automated reasoning systems; the latter can do complex reasoning tasks for the user, whereas AAS's goal is not to *replace* the user's reasoning, but rather to *assist* him as a knowledge management tool for reasoning.

A common element present in all AAS is the *Argument Assistant user interface* (AAI), which plays a key role with respect to the user experience when interacting with the system. The AAI user's acceptance is directly proportional to its quality, since if his experience when using the AAI is rewarding, it will lead to higher productivity and applicability of the tool. A qualified AAI is not only the result of an appropriate surface design of the interaction. It relies on the achievements of some critical characteristics. In particular, one of these characteristics is *usability*, formed by a set of attributes that bear on the effort needed for use, and on the individual assessment of such use, by a stated or implied set of users. Indeed, some software implementations have explicitly considered usability features in the AAI design (e.g. the Hermes system (Karacapilidis and Papadias, 2001), ArguMED (Verheij, 2003), AVERs (Van den Braak et al., 2007), among others). However, there are no standard adopted criteria for assessing the usability of AAI within the argumentation community.

This paper proposes a set of minimum design

Paula González M., Iván Chesñevar C., Pinkwart N. and J. Gómez Lucero M. (2010).
DEVELOPING ARGUMENT ASSISTANT SYSTEMS FROM A USABILITY VIEWPOINT.
In *Proceedings of the International Conference on Knowledge Management and Information Sharing*, pages 157-163
DOI: 10.5220/0003066801570163

guidelines towards usability (DG) instantiated to the particular domain of AAI. First, key features included in the most well known AAIs are identified. Taking as starting point the most solid factors proposed by the Usability Engineering approach (Mayhew, 1999), general mechanisms are added to every feature, thus providing a set of minimal DG for developing usable AAS. Some questions are added to every DG in order to clarify its intended meaning in this particular domain. Furthermore, on the basis of the comparison between the proposed recommendations and the most classical usability related questions (e.g. those questions which help to concretize every general UPs), traditional usability principles are linked to measure quality in use of every DG. Our final goal is to help the design, development and evaluation of usable AAS.

2 ARGUMENT ASSISTANT SYSTEMS

Argument assistant systems (AAS) (Verheij, 2003; Van den Braak et al., 2006) have evolved as software tools which provide an aid for drafting and generating arguments, assisting the user in his reasoning process. This assistance involves several aspects of the argumentation process (e.g. keeping track of the issues that have been raised, assumptions that have been made, evaluating the justification status of the statements involved in the argumentation process, etc.). More specifically (Verheij, 2003), AAS are aids to drafting and generating arguments by a) administering and supervising the argument process, b) keeping track of the issues that are raised and the assumptions that are made, c) keeping track of the reasons that have been adduced for and against a conclusion, d) evaluating the justification status of the statements made, and d) checking whether the users of the system obey the pertaining rules of argument.

Most AAS provide different kinds of facilities to support *argument diagramming*, resulting in “box and arrow” diagrams and allowing to formulate premises and conclusions as statements. These are represented by nodes which can be connected by lines to display inferences; arrows in such lines indicate the inference direction. Some AAS provide facilities for displaying a text file in natural language, from which arguments are to be extracted and analyzed. Several AAS currently exist (Kirschner et al., 2003), among which we can mention Araucaria (Reed and Rowe, 2004), ArguMed (Verheij, 2003), Compendium (Okada et al., 2008), and AVERs (Van den Braak et al., 2006).

In spite of their differences (e.g. the intended application domain), some common features can be observed in most AAS interfaces (AAI). First, they convey the representation of some *user mental model* (i.e., all the cultural and personal-biased users’ perceptions and assumptions, as well as their pre-conceptions about how the system is expected to react), together with the *interaction style* (i.e., all the ways the user can communicate or interact with the software, including both physical and mental actions). Additionally, AAS offer *feedback and support* for the user (related to explicitate the current system status, helping the user to prevent, recognize, diagnose, and recover from errors and misuse, e.g. by means of help and documentation, undo options, etc.) as well as diverse *interoperability* facilities (such as links to multimedia elements). In many cases, there are also *collaborative features* associated with AAS (such as different kinds of awareness, the synchronization, the visualization of shared workspaces, the communication mechanisms, the representation of self and other’s performance and profiles, the shared knowledge , etc.). On the other hand, there are some common features in AAS interfaces typically associated with the argumentation process itself. Two central features are the *visual argument representation* (including the recognition of different types of arguments, their statuses, etc.) and the modeling of *conflict among arguments* which allows the user to recognize the argumentation situation under consideration, Another feature is the *preference criteria* associated with the possibility of visualizing or deducing how the conflict among arguments are resolved.

3 USABILITY PRINCIPLES

Usability is formally defined by ISO 9241-11 as “the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use”.where the *context of use* is a description of the actual conditions under which the interactive system is being assessed, or will be used in a normal working situation. Usability Engineering is a systematic approach to improving the usability of user interfaces by applying a set of proven methods throughout the system development lifecycle.

Usability is such a complex concept that has been divided in a series of measurable principles (also denoted usability attributes) in order to be understood in a better way. Alternative sets of UPs have been proposed, each of them emphasizing different features present in usability definition. Also diverse classifications are proposed when linking them with the formal definition. In spite of this situation, nowadays some common UPs have been agreed within the major part of the Usability Engineering community (Constantine and Lockwood, 1999; Dumas and Redish, 2000; Mayhew, 1999). Consequently, we also have considered these UPs and this classification as the most relevant for the AAI scope. The most accepted of those UPs principles are listed below, including some general questions that have been abstractly defined to clarify and determine the scope of each one of them:

Effectiveness. How do users define success? Is success the same for all stakeholders? What are the goals; what are the tasks? Are there hidden goals?. Can be decomposed in Completeness and Accuracy.

Efficiency. In how many ways the user and system exchange information?. How long do users expect a task to take? Is the task completed in a single session? What styles of interaction do users prefer? What would make the interface feel efficient?. Can be decomposed in Flexibility, Speed and Effort.

Engaging. What kind of work (or play) is this? What are the expectations for style and tone? How often? How long? When, where, how and why?. Can be decomposed in Pleasant and Satisfaction.

Error Tolerance. How familiar is the domain? The terminology? What will users find difficult? What kinds of errors are likely? How serious are their consequence? Will the users understand the problem, or they will need an explanation?. Can be decomposed in Error Prevention and Recovery.

Easy to Learn. Will users expect to have to learn to use it? Are they learning something new? How complex is the task? How often will it be used? How important is it to get it right?. Can be decomposed in Predictability, Consistency and Affordance.

4 THE PROPOSAL

As stated before, this paper proposes a set of minimum design guidelines (DG) that should be taken into account when deploying high qualified AAI from a Usability Engineering viewpoint. The main contributions are summarized in the tables included in Fig. 1 and Fig. 2. First, the key features included at AAIs were identified (see Section 2). Following the lessons learned form Usability Engineering, a minimal set of DG that should be taken into account when designing usable AAI were linked to every of the above specific key features (see third column in Figs. 1 and 2). Besides, to minimize the bias and interpretation of a particular AAS development team, DG related suggestions and recommendations were included, helping to instantiate the proposal to the specific domain of AAI (4th column in Figs. 1 and 2). For clarity reasons, some of those recommendations were expressed as questions, aiming to facilitate their applicability in future real scenarios. Finally, the information of the 2nd column in Figs. 1 and 2 was contrasted against the UPs corresponding specific questions. As a result, each DG was associated with a set of UPs that should be considered when evaluating a AAI (either a final version or a prototype). To improve this selection, in the case of the less novel features (e.g. user mental model, collaborative features, feedback and support) UPs presented in practical examples in (Carroll, 2000; Dumas and Redish, 2000; González et al., 2008a; Sutcliffe, 2002) were considered, since these were used before to test interface features that materialized similar DGs as those shown in Figs. 1 and 2. Note that, even when an alternative development process model is chosen to cope with the AAI creation, the above DGs and their corresponding UPs are still to be considered, being the unique restriction related to its applicability the inclusion of usability as a key factor during the AAI life cycle.

5 DISCUSSION

The AAI Design Guidelines presented in the previous section are a first attempt to systematically describe the factors that constitute usability in the field of argument assistant systems. While the 8 categories and 31 guidelines are, thus, rooted in the relevant research both in the field of usability engineering and in the AAS field, the DGs are still a first attempt and will need further refinement and research. In particular, the following directions are important.

5.1 Operationalization

While all the DGs we propose are concrete and can be used to inform system design, some of the aspects will need a further operationalization in order to simplify the decision if (or to what degree) a guideline was met. In that respect, note that our proposal combines both qualitative and quantitative approaches to asses usability. Consequently, some DGs can be operatized by developing usability metrics where a numerical value (or range) is obtained as a final result. Part of this work has been already done, and some of the DGs in our proposal do contain specific, measurable criteria (e.g., *VR2*, *CA2*, *IO1*, and more). However,

Feature	DG ID	Design Guideline (DG)	Recommendations	Usability Principle
User Mental Model	*UM1*	Provide a tool that match both underlying theory and real world	Are the fundamental elements of the underlying theory represented? How easy and natural is for the user to recognize each type of element?	Affordance Easy to Learn Predictability
	UM2	Apply a cross-cultural viewpoint	Consider cultural-biased issues when choosing design patters (cultural-biased metaphors, genre considerations, iconography, etc.)	Easy to Learn Engaging Accuracy Affordance
	UM3	Explicit the domain	How domain-oriented is the interface? Does the grade of generality helping the user to argue? Do the main elements follow domain-oriented metaphors?	Engaging Satisfaction Effectiveness
Interaction Style	*IS1*	Interaction design should be chosen according to the underlying theory	Is there coherence between the underlying theory and the interaction design? In what ways can the user argue? Are those ways reflecting the underlying theory?	Flexibility Accuracy
	IS2	Enhance user control	How active should the user be to complete the task? Are the main User-Centered Design principles followed?	Engaging Satisfaction
	IS3	Look for aesthetic and minimalistic design	Is there irrelevant information included in the dialogues? Are there unnecessary elements or options cluttering the interface? In what measure is the interface concise and user-focused?	Predictability Satisfaction Affordance
	IS4	Minimize technical considerations. Provide high-level actions and user-oriented language	Are technical restrictions and related issues subordinated to the user experiences? Is the vocabulary domain-oriented or plain in all possible situations (labels, menus, alerts, error screens, help, etc)?	Easy to Learn Engaging Efficiency
	IS5	Do not include unnecessary interaction styles	Does the tool impose adequate limits on the ways in which users can argue? How many styles are supported? Are those styles necessary?	Engaging Satisfaction Error Prevention Easy to Learn
	IS6	Information architecture and interaction design should emphasize general tool purposes (specially if the tool is web-oriented)	How is the layout of the tool's elements in the interface? Are they grouped following some criterion? Are the conceptual structures of the tool coherent and efficient? Is the findability[1] of the tool achieved?	Engaging Effectiveness Error Prevention
	IS7	Balance configurable and fixed issues	How configurable is the interface? Do alternative configurations help to understand what was happening? How many different elements can be included in a single view?	Flexibility Error Prevention Consistency Accuracy
Visual Argument Representation	*VR1*	Arguments must be clearly represented	How are arguments visualized? Are there common design patterns for argument visualization?	Effort Consistency Affordance
	VR2	Arguments should be easy distinguished by the user	Is it easy to recognize different types of arguments (e.g. warranted vs. non-warranted, the user's arguments from other arguments, etc.)?	Predictability Consistency Easy to Learn
	VR3	Argument relevance should be suggested	Is the relevance of different arguments visualized? (screen position, visual design, emphasis by size, by colors, etc.)	Predictability Consistency Easy to Learn
Conflict among Arguments	*CA1*	Provide simple mechanisms to express conflict between arguments	What strategies are used to model conflict analysis among arguments? If grade of generality allows more than one domain, how transferable are those strategies across the different domains?	Consistency Easy to Learn Predictability
	CA2	Provide alternative modeling of the arguing situation currently under consideration	Are different views provided? (e.g. graph view, matrix view, etc.)	Consistency Affordance
	CA3	Provide different views to allow bird's-eye perspective and focalization	Are there mechanisms to zoom in and out? Is it possible to focalize in a particular part of the view? When zooming in or out, is the represented situation still understandable for the user?	Flexibility Consistency Easy to Learn Affordance

Figure 1: Features, design guidelines and related usability principles for developing AAIs.

others do not contain such criteria yet, and in order to assess the value of the catalogue beyond theoretical considerations, they will be required (cf. subsections below). Others DGs will require a more qualitative viewpoint to be operatized. In these cases, our proposal can rely on datamining-based methodologies like those described in (González et al., 2008b). For instance, DG *UM3* states that an interface should explicit the domain - but the precise way of measuring this needs to be defined in order to assess if a specific tool adheres with the guideline. As before, further experimentation will be required (cf. subsections below).

5.2 Empirical Validation of Categories

Another aspect worth discussing are the guidelines themselves. Backed up by literature in usability research and argumentation, and supported by a reverse engineering process carried out, it is reasonable to assume that all the DGs in our list are, in fact, supportive to AAI usability, and that a tool that adheres to

Feature	DG ID	Design Guideline (DG)	Recommendations	Usability Principle
Preference Criteria	*PC1*	Preference criteria should be clear enough	Can non-expert users state the preference criteria after using the tool?	Accuracy
	PC2	Allow user changes in preference criteria	If it is possible to redefine the preference criteria dynamically, how is this visualized? The current view changes radically when doing this? When changing the preference criteria dynamically, how many single actions are necessaries to re-calculate the conflict between arguments?	Predictability Efficiency
Interoperability	*IO1*	Interoperability with other AAS should be supported	Are there import mechanisms? Can arguments from others frameworks be captured? Does the tool support customized plug-ins for different argumentation domains, techniques, etc.?	Efficiency Predictability
	IO2	Multimedia elements should be supported	Is it possible to link multimedia elements to arguments? (as evidence / explanation / follow-ups, etc)	Error Prevention Effectiveness Satisfaction
Feedback and support	*FS1*	Provide feedback regarding conflict among arguments	In which form is the user aware about conflict between arguments?	Consistency Easy to Learn Predictability
	FS2	Allow easy reversal of actions	Are backward and forward mechanisms provided? How difficult is to do/undo actions? How many single actions (mouse clicks, opening task bars, etc) are necessaries?	Error prevention Correctness Accuracy
	FS3	Include help and documentation	Are help and documentation easy to search and focused on the user's task? Do they list concrete steps to be carried out?	Error Prevention Easy to Learn
	FS4	Support user minor errors	Is the system supporting minor user's errors and reporting status when major errors occur?	Error tolerance Error Prevention Predictability Accuracy
	FS5	Inform the current status	Is the system keeping users informed about what is going on, through appropriate feedback within reasonable time?	Error Prevention Efficacy Correctness
Collaborative Issues (if the AAS is collaborative)	*CI1*	Enhance communication mechanisms	In how many ways can users exchange information? Is it possible to keep the view of the interface (the current discussion/ the arguments under consideration) when communicating with others? How easy is to distinguish sent and received information?	Accuracy Engaging Efficacy
	CI2	Include alerts regarding collaboration	Is there any mechanism helping me to detect that others are waiting for my contribution? Can the user distinguish logged in / logged out users? Can the user send an alert to others?	Completeness Prevention Consistency
	CI3	Provide an explicit sensor of the user's and others' collaboration performance	Can the user identify the ownership of other arguments on the current screen?	Effort Consistency Predictability Affordance
	CI4	Distinguish user profiles (specially if the profile defines part of the preference criteria, e.g. arguments weighed by user's expertise)	Are other users' profiles available? Are there any elements in the current view helping the user to be aware of other's expertise? How can the user establish his profile? Can he change it dynamically?	Accuracy Consistency Predictability
	CI5	Implement mechanisms to restart actual state of the task (asynchronous tools)	How difficult is to restart the system's last state after turning off it? How many steps are necessaries to do it?	Effort Consistency Predictability
	CI6	Provide human-oriented awareness	Are there explicit mechanisms to cope with group awareness, social awareness, task-specific awareness, situation awareness, objective self awareness and shared knowledge awareness? How visible and explicit are those mechanisms?	Effort Effectiveness Engaging Satisfaction

Figure 2: Features, design guidelines and related usability principles for developing AAIs (cont.).

all these criteria will achieve high scores in usability tests. Yet, empirical backup for this argument is required. It might well be that some of the guidelines are theoretically valid, but just not important enough to make a real difference (and thus should be removed from the list). Or, other factors not contained in the list may come up in actual system usage in the field (and thus the list needs to be extended). Finally, due to the complexity and diversity of the AAS domain, without empirical data we cannot be sure if all DGs hold for all AAS (e.g., is the guideline *IS2* really helpful for all possible usage scenarios and domains, or do some application scenarios benefit from a more passive user role?). As such, studies which measure the usability of AAIs and relate this information to each single item in our list will be required.

5.3 Grounding in Existing Systems

A systematic comparison of existing AAS with respect to the 31 DGs will be highly interesting: do the tools that are available today fulfil the guidelines (and to which extent)? A prerequisite for this systematic comparison is the availability of measurable success and/or fulfilment criteria for each single DG. Our hypothesis is that many of the existing tools meet some of our DGs (e.g., *VR1* through *VR3* are met by most systems), but that no currently available tool meets them all. For some of the DGs, there are actually quite few tools that fulfil them (e.g., *CA2*). In combination with the empirical backup for the validity of the guidelines, a systematic comparison of the degree of DG fulfilment for each category will enable us to ground the DG list in practice and assess the usability of existing single AAS.

6 RELATED WORK

In the last decade some AAS like Araucaria (Reed and Rowe, 2004), Compendium (Okada et al., 2008), AVERS (Van den Braak et al., 2007), Hermes (Karacapilidis and Papadias, 2001) or the LASAD Prototype (Scheuer et al., 2010) were designed having usability into account. However, they constitute particular approaches, rather than a systematic reverse engineering process as the one which supports the conclusions reported here. Indeed, to the best of our knowledge, this paper is the first to propose usability guidelines for designing AAS in a general setting.

Several separate efforts have been made towards enhancing and assessing usability in AAS. In (Sidlar and Rinner, 2007), the authors offer an usability analysis of Argumaps (Argumentation Maps) to support participants in geographically referenced debates as they occurs. Argumaps are based on the combination of an online discussion forum and an online geographic information system (GIS) component. In contrast with our approach, the usability analysis is focused on this particular tool, not being extensible to AAI in general. Another example where usability features f an AAS have been explicitly studied is the the AVERs system which combines visualizing argumentation and anchored narrative theories. As part of AVERs development, some features related with usability were analyzed (Van den Braak et al., 2006; Van den Braak et al., 2007). Different independent experiments were reported and critically examined giving as well methodological recommendations for future experiments. The authors stress the importance of testing “the usability and user-friendliness of the visualization tool”. However, (Van den Braak et al., 2006; Van den Braak et al., 2007) are not focused on generalizing usability but rather on describing the effects of AAS tools use on an user’s argumentation skills. In (Muñoz et al., 2009), the authors propose the materialization of a complete argumentation system ready to be built in conventional agent software platforms. In contrast with our work, the focus here is on making theoretical argumentation usable for software engineers, not being involved with the development and assessment of AAI.

7 CONCLUSIONS AND FUTURE WORK

Argument Assistant Systems (AAS) are one of the most promising software products emerging from the maturity of argumentation technologies. Their success relies partly upon the possibility of offering a way of solving sense-making problems in terms which are natural and easy to understand for human beings as end users (arguments, conflicts among arguments, etc.). In this paper we have presented a first approach towards standardization of usability for the development of AAS. After identifying relevant features common to the AAS interfaces, we have proposed a minimum set of design guidelines that should be taken into account when designing and testing such interfaces from an Usability Engineering viewpoint. By means of a series of questions and recommendations, the guidelines were instanciated to focus on the particular domain of AAS. Different usability principles were identified and associated with every suggested guideline to evaluate its quality in actual AAS interfaces, ranging form early prototypes to fully implemented final versions.

The consolidation of argumentation technologies has recently led to the definition of standards for argument exchange (notably the Argumentation Interchange Format AIF (Chesñevar et al., 2006)). We think that in the near future similar standards will be required concerning usability of AAS, establishing uniform criteria for its assessment and evaluation. Future work includes the performance of alternative full usability evaluation that will help to validate and improve the set of minimal DG currently identified, as well as the corresponding UPs. In particular, recently a novel usability evaluation methodology called $QUTC_{KDD}$ has been characterized (González et al., 2008b), providing a datamining-based technique for detecting common usability problems of particular contexts of use. The application of such kind of approaches to characterize most relevant usability problems of the specific domain of AAS is currently under consideration.

ACKNOWLEDGEMENTS

This paper was partially funded by CONICET (Argentina) and Projects PIP-CONICET 112-200801-02798 (Argentina), CICYT TIN2008-06596-C02-01 (Spain), and LASAD Project (DFG and TU Clausthal, Germany).

REFERENCES

Brena, R., Aguirre, J., Chesñevar, C., Ramírez, E., and Garrido, L. (2007). Knowledge and information distribution leveraged by intelligent agents. *Knowl. Inf. Syst.*, 12(2):203–227.

Carroll, J. (2000). *Making Use:Scenario-Based Design of Human-Computer Interactions*. MIT Press.

Chesñevar, C., McGinnis, J., Modgil, S., Rahwan, I., Reed, C., Simari, G. R., South, M., Vreeswijk, G., and Willmott, S. (2006). Towards an argument interchange format. *Knowledge Eng. Review*, 21(4):293–316.

Constantine, L. and Lockwood, L. (1999). *Software for Use. A practical Guide to the Models and Methods of Usage-Centered Design*. Addison-Wesley.

Dumas, J. and Redish, J. (2000). *A Practical Guide to Usability Testing*. Intl. Specialized Book Service Inc.

González, M., Granollers, T., Pascual, A., and Lores, J. (2008a). Testing website usability in spanish-speaking academia through heuristic evaluation and cognitive walkthroughs. *Int. Journal of Universal Computer Studies*, 14(9):1513–1529.

González, M., Lorés, J., and Granollers, T. (2008b). Enhancing usability testing through datamining techniques: A novel approach to detecting usability problem patterns for a context of use. *Information & Software Technology*, 50(6):547–568.

Karacapilidis, N. and Papadias, D. (2001). Computer supported argumentation and collaborative decision making: the hermes system. *Inf. Syst.*, 26(4):259–277.

Kirschner, P., Shum, S. B., and (Eds), C. C. (2003). *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making.* Springer-Verlag.

Mayhew, D. J. (1999). *The Usability Engineering Lifecicle. A practioner's handbook for user interface desing*. M. Kaufmann.

Muñoz, A., Sanchez, A., and Botia, J. A. (2009). A software architecture for an argumentation-oriented multi-agent system. *Advances in Intelligent Soft Computing*, 55:197– 206.

Okada, A., Shum, S. B., and (Eds.), T. S. (2008). *Knowledge Cartography: Software Tools and Mapping Techniques*. Springer: Advanced Information and Knowledge Processing Series.

Rahwan, I. and Simari, G. (2009). *Argumentation in Artificial Intelligence*. Springer-Verlag.

Reed, C. and Rowe, G. (2004). Araucaria: Software for argument analysis, diagramming and representation. *Int. Journal on AI Tools*, 13(4):983–.

Scheuer, O., Loll, F., Pinkwart, N., and McLaren, N. (to appear 2010). Computer-supported argumentation: A review of the state-of-the-art. *Int. Journal CSCL*, 5(1).

Sidlar, C. and Rinner, C. (2007). Analyzing the usability of an argumentation map as a participatory spatial decision support tool. *The Urban and Regional Information Systems Association Journal*, pages 47–55.

Sutcliffe, A. (2002). *User-Centred Requirements Engineering. Theory and Practice*. Springer–Verlag.

Van den Braak, S., van Oostendorp, H., Prakken, H., and Vreeswijk, G. (2006). A critical review of argument visualization tools: Do users become better reasoners? In Grasso, F., Kibble, R., and Reed, C., editors, *Workshop on Computational Models of Natural Argument. ECAI 06*, page 6775, Riva del Garda, Italy.

Van den Braak, S., Vreeswijk, G., and Prakken, H. (2007). *AVERs*: an argument visualization tool for representing stories about evidence. In *Proc. of the 11th ICAIL Conf.*, pages 11–15.

Verheij, B. (2003). Artificial argument assistants for defeasible argumentation. *Artif. Intell.*, 150(1-2):291–324.

Verheij, B. (2007). Argumentation support software: Boxes-and-arrows and beyond. *Law, Probability & Risk*, 6:187–208.