Supporting decision-making at the TOUCHE process model by argument assistant systems: a first approach

María Paula González
National Council of Scientific and Technical Research (CONICET), Argentina
Department of Computer Science and Engineering, Universidad Nacional del Sur– Bahía Blanca, Argentina
mpg@cs.uns.edu.ar

Victor M. R. Penichet, Sebastián Romero
Computer Systems Department
University of Castilla-La Mancha
02071 Albacete, Spain
Victor.Penichet@uclm.es
Sebastian.Romero@uclm.es

Abstract

Argument Assistant Systems are software tools used for negotiation and problem solving that have been gaining importance in several areas of Artificial Intelligence, mainly as a vehicle for facilitating rational justifiable decision making when handling incomplete and potentially inconsistent information. This paper proposes the integration of Argument Assistant Systems into the Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments (TOUCHE). The final goal is aimed to provide software tools capable of supporting appropriate rational discussions (and consequently better decision-making processes) within such process model.

1. Introduction and motivations

Over the past decade Argument Systems (AS) have experienced a tremendous growth within different communities in Computer Science [2] [17]. AS implement a dialectical reasoning process by determining when a proposition follows from certain assumptions, analyzing whether some of those assumptions can be disproved by other assumptions in our premises. Argumentation is defeasible when it is possible that conclusions are withdrawn in the light of further argumentation. In this context, Argument Assistance Systems for defeasible argumentation (AAS) -a term coined by [26]- are particularly interesting. AAS and fully AS automated reasoning systems have some differences; the second one can do complex reasoning tasks for the user, whereas AAS's goal is not to replace the user's reasoning, but to assist him in his reasoning process. In the same way, word-processing software assists in the writing and formatting of texts, AAS supports argumentative tasks, as the organization, visualization and evaluation of arguments.

Recently, the integration of the AS within the Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments (TOUCHE) has been proposed for handling commonsense and qualitative reasoning, including incomplete and, probably, contradictory information [8]. A number of TOUCHE limitations were discussed regarding its capabilities for supporting and recording the decision making process underlying final agreements between members of development teams that are carrying out the TOUCHE model. In particular, the incorporation of the AS DeLP [6] within TOUCHE was sketched by means of a proof of concept. However, although DeLP provides an interface for carrying out the dialectical argumentation procedure, experimentation beyond [8] suggests that this kind of AS were not as accepted as expected when the composition of the TOUCHE development team includes a majority of people with few background on formal languages and Computer Science related fields.

On the basis of the obtained results, our hypotheses is that the unexpected lack of engaging was not due to the proposal itself, but related to the fact of being used as an AS engine oriented towards automatisation. Consequently, this paper characterizes the inclusion of AAS systems within TOUCHE to support and document the decision making process carried out by the development team, enhancing more rational and justified procedures. In particular, we will focus on the first and second stages of TOUCHE, where knowledge representation and inference will be handled by means of AAS. Besides, our proposal includes the record at the System Requirement Document of TOUCHE of the arguing situation as Argument Maps (AA), one of the most relevant AAS

elements which consist of a "box and arrow" diagram that represents graphically the arguing situation. Our final goal still reminds: providing software tools capable of supporting appropriate rational discussions (and consequently better decision- making processes) within TOUCHE, aiming to enhance its capabilities, especially those related to maintenance and scalability.

2. Related work

Some early argumentation-based conflict resolution methods and systems have been proposed at the field of engineering design during the decade of 90', as gIBIS [4] and their successors IBIS (which allow to represent issues, positions, and arguments, but fail to support representation of goals (requirements and outcomes) and SISCO [1] (where the objective is providing an ambient for modelling discussion with arguments and contra arguments without arising to a final conclusion and enhancing awareness), REMAP [18] or some pioneer proposals [22]. Even they can be seen as precursors of the current state of the art, at that time the maturity of the AS systems and the available technology were far from being like nowadays.

Recent research has led to some interesting results on modelling dialectical discussions and negotiation by AAS in CSCW scenarios. An exhaustive survey is presented at [20]. Regarding the TOUCHE model, an alternative is proposed by integrating Design Rationale [15]. However, even though some Design Rationale models are based on the use of arguments, they deal with concepts and their relationships whereas AM in AAS are constructed from a premise with a number of reasons and objections. The same distinction can be made when considering the integration of Knowledge Cartography [12] (including interactive Concept Maps) in CSCW systems in opposition at integrating AM. Other interesting distinction has to be done between using AM as proposed in this paper and using AM to cope with GIS-oriented discussions [21]. Whereas the GIS-oriented AM supports participants in geographically referenced debates as they occur (for example, as part of urban planning processes), AMs in AAS are extensible to argumentation situations in general.

An interesting alternative to our proposal is discussed at [11], where a web-based model was presented with the emphasis on a fuzzy-weighed mechanism to compute conflicts among arguments. Finally, note that results presented at [7] and [8] can be seen as previous work that explores alternative uses of general AS in CSCW environments. Nevertheless, in these papers AM was not included. Indeed, the focus was oriented towards computing Shared Knowledge and Shared Knowledge Awareness (see [7]); and carrying on a Proof of Concept to explore the integration of AS systems at the first two stages of the TOUCHE model (see [8]).

3. Sketching the TOUCHE process model

The Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments (TOUCHE) is a process model and a methodology for the development of user interfaces for groupware applications. A deep description of TOUCHE is available in [14]. TOUCHE includes four development stages, namely Requirements Gathering [15], Analysis [16], Design, and Implementation. Key TOUCHE contributions empathise the inclusion of several pre-existing formal models and methodologies, aimed to formalize as much as possible the whole development process to ensure the reliability of the final deployed software. Indeed, Class Diagrams, Co-interaction Diagrams and Organizational Structure Diagram expressed with CTT notation [13] are used at the Analysis stage; Abstract Interaction Object for the Abstract User Interface plus UsiXML conceptual scheme [10] are included at the Design stage; and reference frameworks, such as Cameleon [1], are suggested for the development of user applications in HCI.

In addition, TOUCHE proposes some novel semi-formal model, such as requirement gathering templates, which include metadata for the specification of groupware applications, as well as the System Requirement Document (DRS) [14] based on the Durán's work [5]. Besides, TOUCHE presents new Abstract Interaction Objects providing -in combination with several facets- more expressiveness to represent CSCW interfaces; and new specific CIOs for groupware applications. TOUCHE also includes issues for the traceability between the defined models and

the different stages [15]. A transversal feature of TOUCHE is the inclusion of the user as member of the development team during all the development process, thus providing a user-centred approach where CSCW criteria dominate the whole scenario. Finally, note that the TOUCHE Case Tool was fully implemented to support its application to real CSCW interfaces development (http://www.penichet.net/).

Even TOUCHE strongly relies in formal models to ensure the global quality of both the development process and the resulting CSCW interface, one limitation is that these models only account for final achieved agreement, omitting the record of the arguing process carried out as part of the associated decision making process, where incomplete and possible inconsistent information is normally included within the debate. This limitations turn out to be critical especially in the first and second stages of the TOUCHE model, as decisions made here strongly condition the final product characteristics.

As explained at Section 1, a first proposal shown at [8] was presented to cope with the above problem by means of the integration of the AS DeLP at the TOUCHE, enriching it with a rule-based approach for efficient defeasible reasoning. However, some unexpected resistance was observed in experimentation performed beyond [8], associated with luck of engagement of some TOUCHE development team towards this kind of AS, especially in the cases when the composition includes a majority of people with few background on formal languages and Computer Science related fields. As we will see later on, the inclusion of an alternative AS system named AAS can combine all the improvements arises at [8] with more acceptance, even when performance in automated calculation of arguments will be probably dismissed.

4. Argument assistant systems and argument mapping: an overview

Argument Assistant Systems for defeasible argumentation (AAS) were coined at [26] and have evolved rapidly in the last decade 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.) [9].

As explained in Section 1, AAS are to be distinguished from fully AS automated reasoning systems; the latter can do complex reasoning tasks for the user, whereas AAS goal is not to replace the user's reasoning, but rather to assist him in his reasoning process. Indeed, AAS provide often a realization of a formal argumentation theory, offering a good test bed for analyzing the advantages and disadvantages of the actual application of the theory. An interesting example in this direction is the AS Dungine Java Reasoner which has been successfully integrated with the AAS Araucaria [23].

Most AAS provide different kinds of facilities to support argument diagramming, resulting in "box and arrow" diagrams which represent premises and conclusions as statements. The statements are represented by nodes which can be connected by lines to display inferences, and the arrows in such lines indicate the inference direction. Several AAS currently exist, and in some cases usability has been included, as in ArguMed [25] (http://www.ai.rug.nl/~verheij/aaa/), Araucaria [19], and AVERs [24], among others.

In spite of the differences (e.g. the intended application domain), some common facilities are provided by all AAS, as a visual argument representation (including the recognition of different types of arguments, their statuses, etc.) and the so called Argument Mapping (AM) [9], a visual modelling of conflict among arguments which allows the user to recognize the argumentation situation under consideration, including facilities to visualize or deduce the preference criteria used to solve the conflict among arguments. Additionally, most AAS offer feedback and support for the final user. In some cases, also diverse interoperability facilities are provided (as links to multimedia elements), as well as collaborative issues in the case of collaborative AAS (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.).

5. Supporting TOUCHE decision-making by argument assistant systems

This paper proposes the integration of AAS within the TOUCHE process model, aiming to manage decision making process within the first and the second stages. In our approach, AAS are included to handle commonsense and qualitative reasoning, including the management of incomplete and probably contradictory information. As pointed out in Section 4, AAS enhance both a proactive participation of users and a visual representation of arguing situations, including in many cases the manipulation of natural language sentences.

A feasibility analysis regarding the inclusion of AS within TOUCHE is shown in [8]. As AAS are just a special kind of AS, the positive conclusion arisen at [8] are still valid in the current context. In addition, the graphical nature and user-centred approach of AAS enhances their chance of being better accepted by heterogeneous development teams, in particular those teams including a majority of people with few background on formal languages and Computer Science related fields.

First, new theoretical frames were added at TOUCHE definition, aimed to reflex the selection of AAS. Among other actions, the definition of the SRD was extend and new issues were added at the exiting templates to record AM and some extra evidence of the new arguing methodology under consideration (e,g, links to multimedia objects or web pages). From a more practical side, links to some specific AAS were added at the TCT, thus providing direct access to these resources. In our opinion, having more than one AAS available worth, as different development teams can prefer "box and arrow" over nodes (or the opposite), or web-oriented AAS that allow link web content to nodes (as nodes related issues, evidences, or just reminders). In addition, exploring more than one possible representation of the Argument Maps to be added at the SRD can enhance interdisciplinary understanding of the arguing situation under discussion, as different representation could cope with different user mental models and comprehensions.

Respecting the Design Stage of TOUCHE, as before theoretical frameworks were added to include both AAS and AM within the pool of existing TOUCHE methodologies. Note that in this 2nd Stage, roles and tasks will be instantiated and model by means of some diagrams and descriptions. This time, AAS can be used as a front-end methodology to decide how to link elements in the design diagrams, especially in the cases where more than one solution must be taken into account. The resulting AM situation can be consulted to analyze alternative design responses for requirement descriptions, thus minimizing the subjectiveness and cultural bias present in the decision making process. Besides, instead of document only the final design agreements, alternative designs representing rejected issues and features can be also complied as part of the TOUCHE documentation, adding in this case the AM that provides evidence supporting the rejection. This way, in future similar software development under TOUCHE -or even later on in future iterations of the same process under running- the information captured by the "rejected" AM can be examined again as part of the "lessons learned". Besides, if new information has to be considered (due, for example, to changes in requirements), the rejected AM can be re-embedded in the correspondent AAS, and new arguments can be added modifying final warrant and not warrant conclusion.

6. Case study. Discussion

To illustrate how AAS can be integrated at 1st and 2nd stages of TOUCHE, the scenario presented in [8] will be reused. This way, a comparison between the different solutions (e.g. the one presented at [8] and the one sketched at Section 5) will be possible. Indeed, as more complex or delicate is the arguing situation under discussion, more evident is the value of providing methodologies as the one described in this paper.

Let us suppose we are developing the interface of a groupware application which allows several authors to create the same document through the Internet. When the authors of the document have written a draft, one of them is responsible for sending (through the same application) the document candidate to be published to some reviews. Then the reviewers discuss about the document and give their opinion about the document and give their own opinion on whether it should be published or not. A published document can be read by all the users of the system, even if they are *not* authors or reviewers.

Table 1. Description of part of the functional requirement *Document edition* with the proposed template at the System Requirement Document of TOUCHE (only relevant rows for current case study are included).

RF-8	**Document Edition**
… … …	
Awareness issues	The following actors should be aware of this requirement: • #G-1 (AUTHORS): - *What*: an actor is modifying part of the current document - *How*: current modification is showed graphically - *When*: in real-time - *Where*: in the same workspace, in the same window - *Why*: to know who is modifying what and not to interfere • #G-2 (REVIEW): - *What*: an actor modified a document - *How*: a past modification is showed by e-mail - *When*: after saving the current version, asynchronously - *Where*: in the actor's intranet and by e-mail - *Why*: to know who modified the document and what part
… … …	
CSCW description	Because of the collaborative nature of the current requirement: • Notifications are necessary for user awareness • Insertion, modification, and modification in a document are issues to be careful. Awareness in real time is important. Some actions such as deleting an image could be too fast for the rest of authors to be aware. They should be aware in some way. • Real-time feeling in the document elaboration is important but not vital.
… … …	

```
% Defeasible rules (Commonsense knowledge)
%W stands for an arbitrary writer,
% ┐stands for "not" and T stands for "text"

author (W)  ⇒ show real time (awareness, W)
author (W), Deleting (W, T) ⇒ show_real_time (awareness, W)
author (W), Deleting (W,T), Image (T) ⇒ ┐show_real_time (awareness, W)

reviewwer (W) ⇒ ┐show real time (awareness, W)
reviewer (W), author (W)⇒ show_real_time (awareness, W) %the reviewer is one of the
authors
```

Figure 1. Representation of defeasible arguments derived from statements s_1, s_2... s_7 in the AS DeLP

In the above settings, Table 1 shows part of the specification of a functional requirement called document edition (DC) by means of the metadata of the general template and the extensions introduced to consider the specific features concerning CSCW systems. In this specification, some statements can be observed or deduced, as s_1= "*notification is necessary for user awareness*", *real time is important*", s_4= "s*ome actions such as deleting an image could be too fast for the rest of authors to be aware"*, s_5 = *"real time feeling is important but not vital", s_6="insertion, modification, and modification*

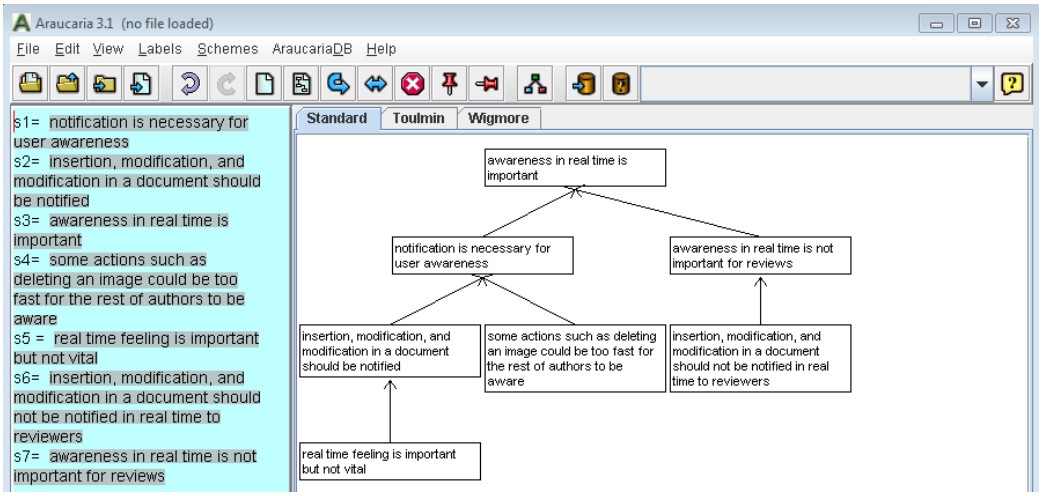

Figure 2. Visualization of the defeasible arguments derived from statements s1, s2… s7 at the Argument Assistant System Auraria (without refutation marks)

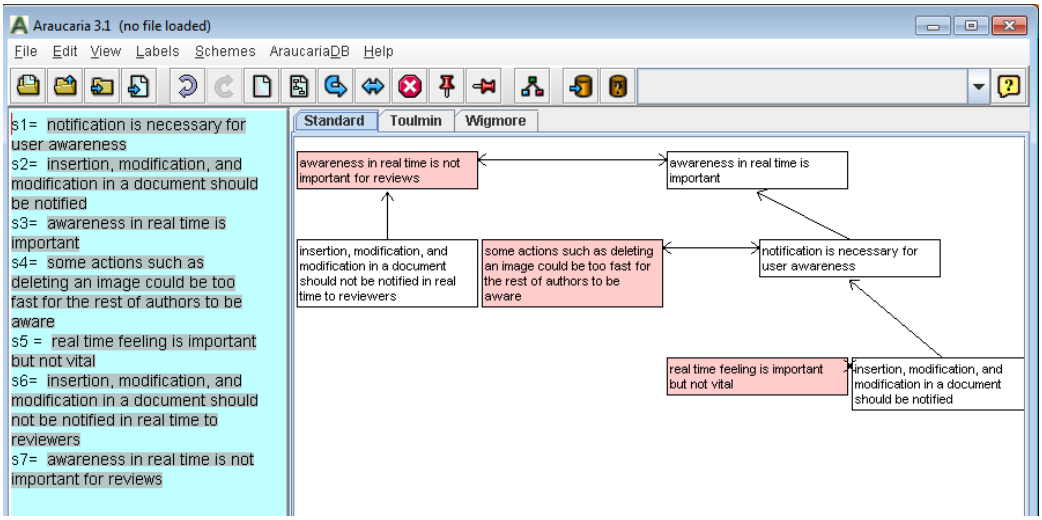

Figure 3. Visualization of the conflict between arguments derived from statements s1, s2… s7 at the Argument Assistant System Auraria (with refutation marks).

in a document should not be notified in real time to reviewers", s_7= "awareness in real times not important for reviewers".

Note also that while some the above statements are in conflict -like s_1 and s_6, or s_3, s_4 and s_7- some other statements like s_3= *"real-time feeling in the document elaboration is important but not vital"* are somehow vague or incomplete, and consequently their interpretation will be probably biased by the development team later on.

At the above point, AAS can be used by members of the development team to deal with the conflictive and incomplete information in DC, thus exploring in a rational and visual way possible correct solutions regarding the involved requirements. Instead of adding the DeLP repre-

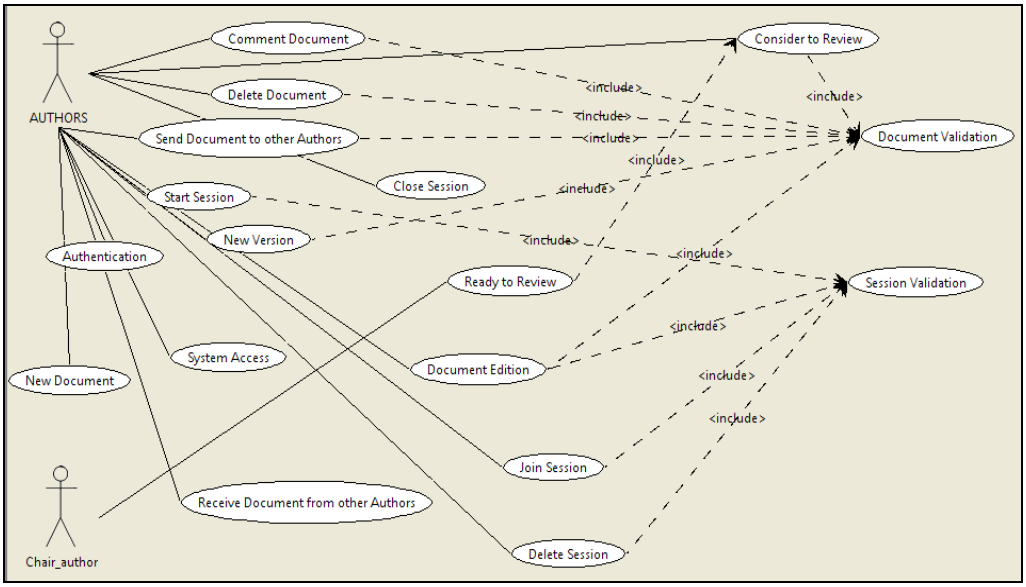

Figure 4. Use Case Diagram for the requirement expressed in RF-8 (chair-author = reviewer)

sentation of the arguing situation under consideration (see Figure 1), the AM shown in Figure 2 and Figure 3 could be included as evidence at the DC template in the SRD under definition, among others AM. While Figure 2 shows a screen shoot of an AM representing the chaining between arguments derived from $s_1,...,s_7$ at the Araucaria interface; Figure 3 depicted the screen shoot of the same Araucaria interface where some conflict between arguments were remarked (arguments in white boxes against "contra-arguments" in coloured boxes) .

Later on, during the development of the Second Sate in TOUCHE, roles and tasks will be instantiated and modeled by means of some diagrams and descriptions. At this point the above evidence (e.g. the obtained AM) can be consulted to arise more appropriate design decisions, following a rational procedure based on the arguing process carried out as part of the associated decision making process, where defeasible (incomplete and possible inconsistent) information were handled. Coming back to our example, the DC related requirements will be designed by means of the analysis of real Use Case Diagrams (see one example at Figure 4) that can be easily collected and instantiated. Following our proposal, the AM describing conflictive situations like those showed above can be used to decide how to link elements in the diagrams of the Second Stage, especially in those cases where more than one solution must be considered.

Indeed, members of the development team can rely on AAS expressiveness to analyze alternative sets of design responses for requirement descriptions, thus minimizing the cultural bias present in the decision making process. Note that the graphical scenario and natural language provided by AAS promote the comprehension and active participation of different stakeholder involved in the development process beyond their expertise on formal languages and/or Computer Science related fields.

7. Conclusions and future work

This paper characterises the integration of Argument Assistant Software (AAS) within the Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments (TOUCHE), aiming to assisting the development team in reasoning processes associated with decision making. In our approach, this reasoning process is captured by means of AAS available through the graphical TOUCHE Case Tool. Indeed, AAS were selected based on its proactive users' participation, the visual modelling of arguing situations under discussion, and the possibility of expressing

argument in natural language. As pointed out in [2,8,16,19], AAS demonstrated to be very helpful during sense making process, and has been used in many situations of every day's life, when people when faced with new information need to ponder its consequences, in particular when attempting to understand problems and come to a decision.

Besides, this paper proposes the inclusion of the Argument Mapping (AM) provided by AAS at the Systems Requirement Document, destined to evidence the negotiation procedures carried out by the development team. This way, the goal of going beyond the record of the final agreements is achieved. The current contributions were oriented towards 1st and 2nd stages of TOUCHE, since at early stages of software development critical selections based on development team agreements should be specially debate, strongly conditioning the rest of the deployment. Our final goal is to provide software tools capable of supporting appropriate rationale discussions (and consequently better decision- making processes) within TOUCHE, aiming to enhance its capabilities, especially those related to maintenance and scalability.

Future work is focused on performing a set of experiments in order to validate the real scope of the current proposal in comparison with the results discussed in [8]. In that respect, the emerging of hybrid AS systems where the AAS are enriched with capabilities towards automatic calculation of conflict among defeasible arguments open new perspectives. In particular, the recent java applet version of Dunguine has to be explored [23], since it provides a plug-in software component. Indeed, the fully embedding of the AAS functionalities as part of the TOUCHE Case Tool will probably minimize users mental effort associated with the necessity of focusing the attention simultaneously on both the TCT and the AAS interfaces. Work in this direction is being pursued.

Acknowledgments

This work was founded by CONICET (Argentina) and the projects PGI 24/N020, PGI 24/N023 (UNS, Argentina), PIP-CONICET 112-200801-02798 (Argentina), CICYT TIN2008-06596-C02-01 (Spain) and the Junta de Comunidades de Castilla-La Mancha PEII09-0054-9581 and PII2C09-0185-1030 projects.

References

[1] Bellarassai, G. et al. SISCO: A tool to Improve Meeting Productivity", International Workshop on Groupware, CRIWG'95, pp 149–161

[2] Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Bouillon, L., Vanderdonckt, J., 2003: A Unifying Reference Framework for Multi-Target User Interfaces. Interacting with Computers. Vol. 15, No. 3, pp. 289-308

[3] Chesñevar,C. I.; Maguitman, A., Loui, R. Logical Models of Argument. ACM Computing Surveys, 32(4) pp. 337-383 (2000).

[4] Conklin, J., Begeman, M. gIBIS: A Hypertext Tool for Exploratory Policy Discussion. ACM Transactions on Inf. Systems, v.6 (4), pp. 303-331 (1998).

[5] Durán, A.: Applying Requirements Engineering. Catedral Publicaciones, ISBN: 84-96086-06-2 Spain (2003)

[6] García, A.; Simari, G. Defeasible Logic Programming: An Argumentative Approach.Theory and Practice of Logic Programming, 4(1) pp. 95-138 (2004).

[7] González, M P; Chesñevar, C; Collazos, C; Simari, R. Modelling Shared Knowledge and Shared Knowledge Awareness in CSCL Scenarios through Automated Argument Systems. LNCS, vol. 4715, pp. 207-222. Springer Verlag (2007).

[8] Gonzalez, M.P., Penichet, V., Simari, G., Tesoriero, R. Development of CSCW Interfaces from a User-Centered Viewpoint: Extending the TOUCHE Process Model through Defeasible Argumentation. LNCS, Springer-Verlag, ISSN 0302-9743, vol 5619, Part VII, pp. 955-965 (2009).

[9] Kirschner, P., Buckingham S., Carr, C. (Eds.). Visualizing Arguments: Software Tools for Collaborative and Educational Sense-Making. Springer-Verlag, London (2003).

[10] Limbourg, Q., et all. UsiXML: a Language Supporting Multi-Path Development of User Interfaces. LNCS, v. 3425, pp. 200-220 (2005)

[11] Liu, X., Raorane, S., Zheng, M.,Leu, M. An Internet Based Intelligent Argument System for Collaborative Engineering

Design. Proc CTS'06, IEEE Comp. Society, ISBN 0-9785699-0-3, pp.318-325 (2006)
[12] Okada, A, Buckingham Shum, S, Sherborne, T (Eds). Knowledge Cartography: Software Tools And Mapping Techniques. Springer, ISBN 1848001487 (2008).
[13] Paternò, F.; Model-based Design and Evaluation of Interactive Applications. Springer, ISBN 1-85233-155-0 (1999)
[14] Penichet, Victor M. R.: Task-Oriented and User-Centred Process Model for Developing Interfaces for Human-Computer-Human Environments. PhD. University of Castilla-La Mancha (2007)
[15] Penichet, Victor M. R.; Lozano, Maria D.; Gallud, Jose A.; Tesoriero, Ricardo: User Interface Analysis for Groupware Applications in the TOUCHE Process Model. International Journal Advances in Engineering Software (ADES) ISSN: 0965-9978, v 40, pp 1212-1222 (2009)
[16] Penichet, Victor M. R.; Lozano, Maria D.; Gallud, Jose A.; Tesoriero, Ricardo: Requirement-based Approach for Groupware Environment Design. Journal of Systems and Software (JSS). ISSN: 0164-1212. DOI: 10.1016/j.jss.2010.03.029 (2010)
[17] Rahwan, I.; Parsons, S.; and Reed, C. (Eds.). Argumentation in Multi-Agent Systems. LNAI (4946), Springer (2008).
[18] Ramesh, B; Dhar, V. Supporting Systems Development by Capturing Deliberations During Requirements Engineering. IEEE Trans. Software Eng., v. 18 (6), pp. 498-51(1992).
[19] Reed, C.; Rowe, G. Araucaria: Software for Argument Analysis, Diagramming and Representation. Int. J. on Artificial Intelligence Tools, Vol. 13(4), pp. 961-979 (2004)
[20] Scheuer, O, Loll, F, Pinkwart, N, McLaren, N. Computer-Supported Argumentation: A Review of the State-of-the-Art. Int. Journal ijCSCL, v5(1) (to appear 2010)
[21] Sidlar, C, Rinner, C. Analyzing the Usability of an Argument Map as a Participatory Spatial Decision Support Tool. Journal URISA, Urban and Regional Information Systems Association, issn 1045-8077, pp. 47-55 (2007).
[22] Sillence, J. Intelligent Argument Systems: Requirements, Models, Research Agenda, and Applications. Encyclopedia of Library and Inf Science, USA, pp. 176 – 217 (1997).
[23] South, M, Vreeswijk, G, Fox, J. Dungine: A Java Dung Reasoner. Proc. of COMMA 2008, IOS press, pp. 360-368 (2008).
[24] Van den Braak, S, Vreeswijk, G, Prakken, H. AVERs: an argument visualization tool for representing stories about evidence. Proc. ICAIL '07, pp. 11-15 (2007)
[25] Verheij B. Virtual Arguments. On the Design of Argument Assistants for Lawyers and Other Arguers. The Information Technology & Law Serie, v6, IteR (2005)
[26] Verheij B. Artificial argument assistants for defeasible argumentation. Journal of Artif. Intell, v. 150 (1-2), pp. 291-324 (2003)