An Integrative Approach to Diagram-based Collaborative Brainstorming

Diogo Azevedo [a], Jordan Janeiro [b], Stephan Lukosch [b], Robert O. Briggs [c], Benjamim Fonseca [a]

[a] School of Science and Technology University of Trás-os-Montes e Alto Douro, [b] Systems Engineering Department Technological University of Delft, [c] MIS Department, San Diego State University, California, USA

al33070@utad.eu, *j.janeiro@tudelft.nl*, *s.g.lukosch@tudelft.nl*, *robertowenbriggs@gmail.com*, *benjaf@utad.pt*

Abstract. The need for computer supported collaboration has grown over the last years and made collaboration processes an important factor within organizations. This trend has resulted in the development of a variety of tools and technologies to support the various forms of collaboration. Many collaborative processes, e.g. strategy building, scenario analysis, root cause analysis and requirements engineering, require various collaboration support tools. Within these synchronous collaborative applications to create, evaluate, elaborate, discuss, and revise graphical models, e.g. data flow, fishbone and brainstorming diagrams, play an important role. Currently, the necessary tools are not integrated and flexible enough to support such processes. In this paper, we introduce a synchronous collaborative brainstorming diagram editor that is integrated in a flexible group support system. By this our approach goes beyond the current state of the art as we can be seamlessly integrated with other collaboration support tools such as text-based brainstorming, voting, etc.

1. Introduction

Working practices had an important growth over the years, especially on group works - a group of people engaged in the execution of several objectives of a common task (Rowley A., 2006, and Frost and Sullivan, 2007). Therefore such an effort should be helped by collaborative practices such as the Computer Support and Cooperative Work - CSCW, which improves the performance of a group in

the execution of tasks, through group work supported by information and communication technologies. Groups can become even more productive when supported by Group Support Systems – GSS. It is decisive that GSS adopt techniques for the development of groupware applications that meet non-functional requirements (quality attributes) such as interoperability, integration, reliability and usability (Ana B. Pelegrina, et al, 2010).

Many collaborative processes, e.g. strategy building, scenario analysis, root cause analysis and requirements engineering, require various collaboration support tools. Within synchronous collaborative applications to create, evaluate, elaborate and revise graphical models by groups, e.g. data flow diagrams, work structure breakdowns and fishbone diagrams. Currently, there is lake of support on GSS for such processes. GSS must therefore offer users collaborative environments where they can interact (Rafael Duque, et al, 2009), however many of these systems fail when providing the right tools for effective collaboration (Grudin J., 1994). Analyze how groups work and evolve is necessary when we consider the social dimension of the collaborative work (Grudin J., 1988).

In this paper we present a Collaborative Line-and-Symbol Diagramming Component – CLSD Component which offers a collaborative environment to manage graphical models and thereby their related collaborative processes. To achieve such a collaborative environment we have been concerned with awareness that as claimed by Dourish and Bellotti (1992) is defined as an understanding of others activities, which provides a context for your own activity. According to (Carl Gutwin, et al, 2004) group awareness information includes knowledge about who is on the collaborative environment, where they are working, what are they doing and their further intentions (Ana B. Pelegrina, et al, 2010). Furthermore, we took into consideration which techniques and diagram types can be used to support collaborative diagramming efforts, and how the features and functions of a single-user differ from a multi-user diagramming tool in order to optimize the values that groups can create through collaborative diagramming. CLSD Component is integrated as a plug-in component within the Computer Assisted Collaboration Engineering (CACE), and thereby can be used in various different processes. CACE approach embeds collaboration expertise with collaboration technologies (Briggs, et al, 2010), so that participants can gain the same benefits without any special training (Mametjanov, et al, 2011).

In the remaining of this paper, we define a set of concepts required within GSS and for Collaboration purpose. After that, we present the requirement analysis giving a scenario of collaborative processes and thereby the set of requirements. In the next chapters the architecture, features and modeling of the CLSD Component are addressed. Before concluding this paper, we fully explain the approach used to implement the CLSD Component.

2. Group Support Systems

A Group Support Systems - GSS consists on a suite of tools for focusing and structuring discussion, while it reduces the cognitive costs of communication and information access among group members making a joint cognitive effort towards a common goal (Robert and Briggs, 2000). Under certain circumstances, industry, military and academic groups who use GSS were able to realize substantial gains in productivity (Fjermestad and Hiltz, 2000). However, the set of tools that GSS can offer are restricted with a limited set of configurable features, for example it can be difficult to fit a collaborative process in a GSS platform. Furthermore, GSS platforms must be flexible to be personalized according to the processes, but to accomplish it they need to follow the component-based software development concepts to become more suitable to different processes' parts. There is a new generation of groupware systems following the component-based approach, such as DACIA (Ladu and Parakash, 2000) to support mobile applications, CoCoWare (Slagter, et al, 2000) to develop applications, and TeamComponents (Jörg and Claus, 2000) to develop either single-user or groupware applications.

3. Requirements Analysis

In the following we elaborate on the requirements that a GSS system has to fulfill allowing collaboration engineers to configure synchronous collaborative applications that actually fit specific collaborative processes, such as strategy building, scenario analysis, root cause analysis and requirements engineering. To illustrate these requirements we present a scenario of a collaborative strategy building process that uses collaborative diagramming and other collaborative applications, e.g. a text-based brainstorming. Two activities that can be considered in this scenario are: 1- a text-based brainstorming for strategy building; 2- a diagram-based brainstorming to organize, connect and manage strategies based on the data gathered in the previous activity (Figure 2).

In the above scenario, we have to support collaboration engineers in (R1) designing collaborative processes, such as strategy building, root cause analysis, and (R2) design suitable collaboration support. For that the GSS needs to support (R3) the integration of components that support collaborative processes, by allowing re-using of existing components (Ana B. Pelegrina, et al, 2010). Furthermore, it must be able to (R4) share, exchange and efficiency (interoperability) of data between components (Hofte H., et al, 1995, and Simone C., et al, 1999, and Ana B. Pelegrina, et al, 2010), in order to re-use the data gathered for example from the first activity (text-based brainstorming) into the second activity (diagram-based brainstorming). Additionally, we do not know all the support that is needed so that (R4) the set of components must be extensible

(Ana B. Pelegrina, et al, 2010) by software developers and (R5) man API to support them (Riehle D., 2000) should be provided. Finally, (R6) our scenario requires collaborative diagramming, and for that we have identified additional requirements.

The list of requirements is based on the analysis of other existing Diagram Software, such as Banxia[1], Smart Ideas[2] and Ext Designer[3]. In this case, the requirements address the interaction that Collaborative Diagramming has to provide to groups while they participate in collaborative environments. It must be possible for group members to (R6.1) insert, import (text-based) and manage ideas into a diagram-based format, like our previous strategy building scenario. Following, ideas are (R6.2) diagram-based organized (clusters and color manager) and (R6.3) connect arrows (connect ideas through arrows). Lastly, group members can unintentionally provoke data conflicts between contributions and therefore it is required to provide (R6.4) feedthrough (Dix A., et al, 1993) - context awareness with the scope (who has been doing what) of other members' activities, consequential communication (Segal L., 1995) - data with their information and the resources that are nearby, and also (R6.5) trigged locking mechanisms when updates occur.

4. Approach

According to Ana B. Pelegrina, et al, 2010 there are GSS systems addressing some of the requirements described above, however for our approach we have chosen a GSS called Action Centers because it addresses all of the above requirements and it fits with our purpose. Two parts form Action Centers: a CACE editor and a Process Support System (PSS). The CACE editor is a tool to design an effective work practice by defining the content and sequence of collaborative activities that are packaged into the PSS (Mametjanov, et al, 2011).

The Action Center therefore does not have any tools, as alternative these tools are plugged into the Action Center as components to simply make them available in the running system. So, the Action Center supports the design of collaborative applications (R1), and allows components (as our CLSD Component) to be assembled by Collaboration Engineers into the CACE editor (R2). These components have access to shared data (R4), are configurable (R3) and can be (re)-designed by other Collaboration Engineers. They usually consist of a user

---

[1] Banxia (Decision Explorer) is a proven tool for managing software issues. Structure and analyse of qualitative information. More information can be found in http://www.banxia.com/dexplore/.

[2] Smart Ideas concept-mapping software brings the power of visual learning to classrooms, through interactive white boards. More information can be found in http://smarttech.com/.

[3] Ext Gui Designer is a graphical user interface builder for web applications. Developed by Sierk Hoeksma. More information can be found in http://www.projectspace.nl/.

interface for displaying data shared in a group, some input mechanism, and business logic.

Furthermore, the Action Center provides two Javascript objects to manage data and their updates – ActionCenterListener, and an ActionCentersAPI (R5) that offers services to create and support the development of collaborative components. Additionally, the data is managed through dynamic communication channels using CometD[4] to a Universal Data Model (Mametjanov, et al, 2011), to dynamically create and store arbitrary relational data. The UDM and the two JavaScript objects offer some mechanism to manage contribution, such as *modifiedBy* to know who (6.4) changed the data, and *lockedBy* to (R6.5) edit-lock entities and their attributes to provide single-user editing. A more detailed description of the system can be found on (Mametjanov, et al, 2011).

Action Center does not address all requirements needed for Collaborative Diagramming. For that purpose, we implemented our (R6) CLSD Component that consists of an XML wrapper and an implementation in JavaScript with Ext JS[5] and an extended library called Joint JS[6]. The JointJS library is used for (R6.1, R6.2 and R6.3) creating diagrams that can be fully interactive for both implementing a diagramming tool (as our CLSD Component) as well as simply for publishing diagrams.

The CLSD Component is a web-based application that supports the cooperation of group participants towards group work. For example, it might support the group in a text-based or a diagram-based brainstorming. Figure 1 shows the overall architecture of our approach.
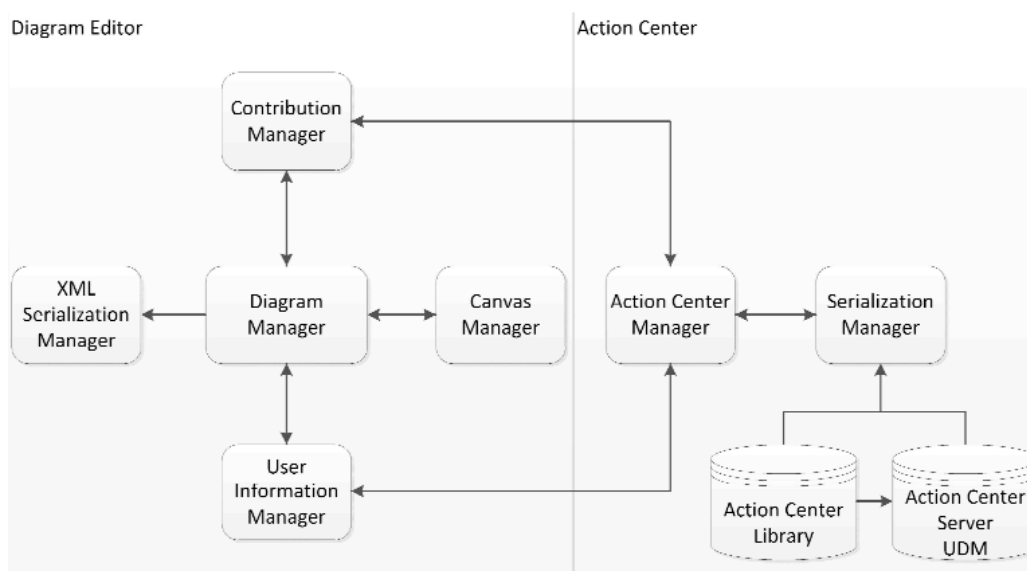

Figure 1. The CLSD Component Architecture coupled to Action Center

---

[4] The Dojo foundation. Cometd. More information can be found in http://cometd.org/.

[5] Ext JS is a javascript framework for developers. More information can be found in http://www.sencha.com/.

[6] Joint JS is a JavaScript library developed by David Durman, More information can be found in http://www.jointjs.com/.

The Diagram Manager is the core manager of our CLSD Component, it is responsible for all processes of input and output and their distribution through the overall system, and for all connections inside the Diagram and between the Action Centers and the Diagram. Additionally, it connects with the Canvas Manager that is the bridge between the core manager of our system and the user - Figure 2. The User Interface (UI) influences its degree of acceptance since it allows communication, collaboration and coordination activities among several users interacting with the system (Victor M. R. P., et al, 2008). The Canvas Manager manages the CLSD Component design, the concepts and their connectors, and the collaborative tools / awareness mechanisms required (Carl Gutwin, et al, 2005), such as the list of users in the session, Telepointers – support actions, intentions and location awareness (Victor M. R. P., et al, 2008), and feedthrough – actions of a particular user can be shown to other users that collaborate in some task (Dix A., et al, 1998 and Carl Gutwin, et al, 2004).
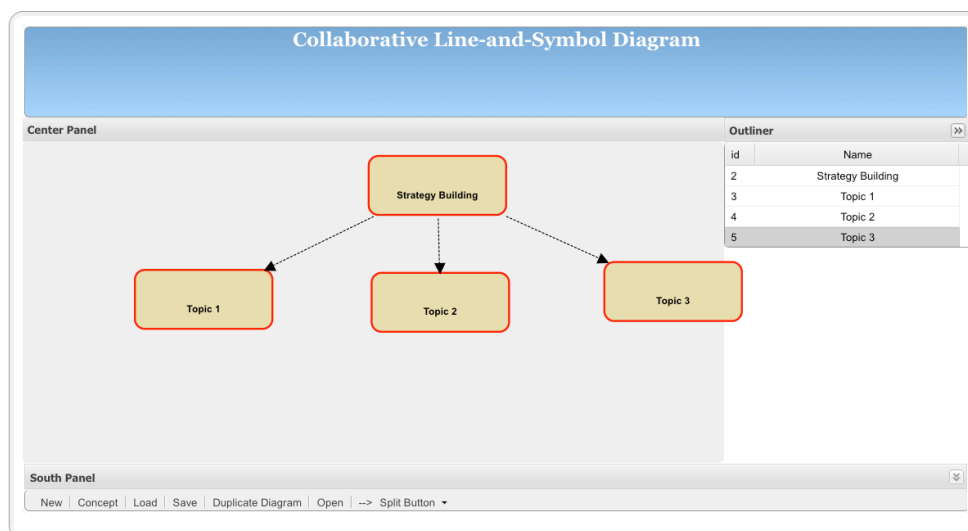

Figure 2. Collaborative Line-and-Symbol Diagramming Component

The Contribution Manager can also be called of Diagram Database Manager since it is responsible for adding, fetching and updating contributions to the Action Center Database. These contributions that are sent to the database can include concepts, arrows, JSON messages or objects and are trigged through notification mechanisms. To manage the information of users that are working in the diagram, such as listening online users, giving personalized information of each of them, and the scope of their activities - group awareness becomes a critical component in successful coordination (Carl Gutwin, et al, 2004) - we have implemented the entity User Information Manager. More information about groupware applications functionalities can be found at (Carl Gutwin, et al, 1998 and Carl Gutwin, et al, 2002).

Finally, another feature developed was the XML Serialization Manager, which is an output file that allows users to visualize their diagrams out of the Action Center.

Action Center in combination with CLSD allows us to support various different processes that require different forms of collaboration. Taking on consideration our previous scenario we take a closer look to the strategy building processes. Data is gathered from a text-based brainstorming (Outliner Component – First Activity) and stored into the UDM – Universal Data Model in Action Centers. The union of the Outliner Component with the CLSD Component (Second Activity) creates an Action Center, where the data, which is selected (identified) based on their relationship types and attributes by the Action Centers, is forward fetched (import) from the UDM and loaded (insert) to the CLSD Component. CLSD transforms it into a diagram-based format where group members can further manage and organize data as collaborative processes. Each single user controls the selection and manipulation of data and until he or she is finished no one else can have access to manipulate that specific data. For that purpose at each moment (through notification mechanisms) concepts shows a locking icon and a scope of action (feedthrough) of the user who is manipulating it.

5. Final Remarks and Future Work

Working practices can become even more productive when supported by GSS. They are becoming widely used thanks to the improvement of network infrastructure, communications, and development tools (Victor M. R. P., et al, 2008). Currently, the necessary collaboration support tools to create, evaluate, elaborate, discuss, and revise graphical models are not integrated or flexible enough (Akhil Mehra, et al, 2005) within GSS to support collaborative processes, such as data flow, fishbone and brainstorming diagrams. According to Bratitsis and Dimitracopoulou (Bratitsis and Dimitracopoulou, 2006), the techniques and information used by awareness mechanisms to the analysis of collaborative processes in which users accomplish common goals is considered the further step. For that there are several models to describe users actions in collaborative environments (Martínez et al, 2003).

In this article, we presented a Collaborative Line-and-Symbol Diagramming Component – CLSD Component assembled in a CACE editor to address the above challenges. A collaboration support tool that consists of a XML wrapper and an implementation for creating diagrams that can be fully interactive for both implementing a diagram-based brainstorming session to manage collaborative processes as well as simply for publishing diagrams. Furthermore, group members can insert, import (text-based) and manage ideas into a diagram-based format through a collaborative environment provided by the GSS system.

In future work, we will observe how practitioners and experts interact with our CLSD Component. We want to use the results to improve the flexibility and usability (Holzinger, 2005) of our component, and further see the exchange of data between components when changing from a text-based brainstorming to a diagram-based brainstorming.

6. Acknowledgments

The authors would like to acknowledge to all colleagues from University of Trás-os-Montes e Alto Douro - UTAD and Technological University of Delft - TU Delft that contribute to the development of our Collaborative Line-and-Symbol Diagramming Component.

This work has been partially supported by the FP7 EU Large-scale Integrating Project SMART VORTEX (Scalable Semantic Product Data Stream Management for Collaboration and Decision Making in Engineering) co-financed by the European Union. For more details, visit http://www.smartvortex.eu/

7. References

Grudin J., (1988): 'Why applications fail: problems in design and evaluation of organization or organizational interfaces', *Proceesdings of the ACM conference on computer-supported cooperative work*, pp. 85-93.

Dourish, P. and Bellotti, V. (1992): 'Awareness and coordination in shared workspaces', *Conference proceedings on Computer-supported cooperative work*, volume 0, pp. 107-114.

Dix, A., Finlay J., Abowd G., and Beale R. (1993): 'Human-Computer Interaction', *Prentice Hall*.

Grudin J. (1994): 'Computer-supported cooperative work: history and focus', *IEEE Comput*, pp. 19-26.

Hofte H., et al, (1995): 'CSCW Infrastructure Research at TRC', *ACM SIGOIS Bulletin*, vol. 15.

Segal L., (1995): 'Designing Team Workstations: The Choreography of Teamwork', *Local Applications of the Ecological Approach to Human-Machine Systems*, pp. 392-415.

Carl Gutwin, Greenberg S., (1998): 'Design for individuals, Design for Groups: Tradeoff between power and workspace awreness', *ACM Computer Supported Cooperative Work*, pp. 383-397.

Dix, A., Finlay J., Abowd G., and Beale R. (1998): 'Human-Computer Interaction', *Prentice Hall*.

Simone C., et al, (1999): 'Interoperability as a means of articulation work', *WACC '99: Proceedings of the international joint conference on Work activities coordination and collaboration*, pp. 39-48.

J. Fjermestad and S. R. Hiltz. (2000): 'A descriptive evaluation of group support systems case and field studies', *Journal of Management Information Systems*, pp. 115-159.

Jörg R., Claus U. (2000): 'Developing synchronous collaborative applications with TeamComponents', *Fourth International Conference on the Design of Cooperative Systems*.

Ladu, R. and Parakash, A. (2000): 'Developing adaptive groupware applications using a mobile component framework', *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, pp. 107-116.

Riehle D., (2000): 'Framework Design: a Role Modeling Approach Dissertation', *ETH Zurich*.

Robert, D. and Briggs, R. (2000): 'GSS for presentation support', *Communications of the ACM*, pp. 43-9197.

Slagter, R., Biemans, M. and Hofte, H. (2001): 'Evolution in use of groupware: facilitating tailoring to the extreme', *Groupware, 2001. Proceedings. Seventh International Workshop*, pp. 68-73.

Carl Gutwin, Greenberg S., (2002): 'A Descriptive Framework of Workspace Awareness for Real-Time Groupware', *Journal Computer Supported Cooperative Work*, pp. 72-81.

Martínez A., et al, (2003): 'Towards an XML-based model for the representation of collaborative action', *conference on computer support for collaborative learning*, pp. 14-18.

Carl Gutwin, Reagan Penner, Kevin A. Schneider (2004): 'Group Awareness in Distributed Software Development', *CSCW*, pp. 72-81.

Carl Gutwin, et al, (2005): 'Supporting Group Awareness in Distributed Software Development', *Engineering Human Computer Interaction and Interactive System*, vol. 3425, pp. 383-397.

Akhil Mehra, et al, (2005): 'A generic approach to supporting diagram differencing and merging for collaborative design', *ASE*, pp. 204-213.

Holzinger, (2005): 'Usability engineering methods for software developers', *Communication of the ACM* vol. 48(1), pp. 71-74.

Bratitsis T., Dimitracopoulou A., (2006): 'Monitoring and analyzing group interactions in asynchronous discussion with the DIAS system', *12th international workshop on groupware*, pp. 54-61.

Frost and Sullivan (2007): 'Meetings around the world: The Impact of Collaboration on Business Performance'.

Schümmer, T. and Lukosch, S. (2007): *Patterns for Computer-Mediated Interaction*, John Wiley & Sons, Ltd.

Victor M. R. P., Maria Dolores Lozano, José A. Gallud, Ricardo T., Maria L. Rodríguez, José L. Garrido, Manuel Noguera, Maria V. Hurtado (2008): 'Extending and Supporting Featured User Interface Models for the Development of Groupware Applications', *J. UCS 14(19)*, pp. 3053-3070.

Rafael Duque, Manuel Noguera, Crescencio Bravo, José Luis Garrido and Maria Luisa Rodríguez (2009): '*Construction of interaction observation systems for collaboration analysis in groupware applications*', Advances in Engineering Software 40(12): pp. 1242-1250.

Ana B. Pelegrina, et al, (2010): 'Integrating Groupware Applications into Shared Workspaces', *RCIS*, pp. 557-568.

Briggs, R., Kolfschoten, G., Vreede, Gert-jan, Albrecht, C., and Lukosch, S. (2010). 'Facilitator in a Box: Computer Assisted Collaboration Engineering and Process Support Systems for Rapid Development of Collaborative Applications for High-Value Tasks' *Information Systems*, pp. 1-10.

Mametjanov, A., Kjeldgaard, D., Pettepier, T., Albrecht, C., Lukosch, S., and Briggs, R. (2011): 'ARCADE: Action-centered Rapid Collaborative Application Development and Execution', *Hawaii International Conference on Systems Sciences,* volume 0, pp. 1-10.