Awareness in Computer-Supported Collaborative Modelling. Application to GenMyModel

Michel Dirix

Axellience 59000 Lille - France
michel.dirix@axellience.com
PhD student since January 2013

Abstract. Costs and markets lead engineering teams to collaborate from different location all over the world. Modelling tools are present in development processes to produce complex software and these tools had to be highly collaborative to permit teams to be productive. For that, teams need information about others and their activities. Currently, the tools are mainly web-based, this is the observed trend in every phase of software development, because it does not require any installation and configuration, but they do not give enough information. This paper will present awareness, which is an answer to the information needed during collaboration, and through a scenario we will see how to answer to awareness questions in UML Case Tools.

Keywords: Collaboration, Awareness, UML

1 Introduction

Since a decade, outsourcing and offshoring projects or part of them have become regular practices in software industry. These ones can have different reasons like the need to cut down costs or to explore new markets. Collaboration between project team members suffers from resulting geographical distribution. First, such situations involve language, cultural and timezone differences. Second, in co-located settings, information is shared by personal interruptions (popping into a colleague office or meeting over coffee), formal face to face meetings, emails, instant messaging. Collaboration is "natural" and each team member can easily find an answer to questions about it: *who is participating to the activity? Where are they? Are they active in the workspace? What are they doing? What are they going to do? What changes are they making?*[1]. The way to find the answers to the previous questions is far more difficult when dealing with geographical distributions.

The topic of my PhD thesis addresses the problem of modelling in distributed teams. Software practitioners usually draw diagrams when they design new applications, maintain existing ones or discuss with their clients using modelling tools [2]. Besides, models are important to manage knowledge and provide an

efficient support to coordinate activities when the projects aims at producing complex software. It is important that modelling tools intent to support collaboration in the context of distributed teams, that is to say, to provide similar working conditions close to co-located settings. Collaborative modelling tools exist (see section 2) but they only focus on and provide basic communication features. They slightly address one of the main aspects of Computer-Support Cooperative Work (CSCW): awareness. Collaboration awareness is an issue that has been considerably studied by the CSCW community. It has been much less studied in software engineering. But as web-based tools currently appear in software projects (e.g. dashboard, code editor) and as they highlight real-time collaboration features, the general interest for collaboration awareness is likely to increase. The goal of my PhD is to study problems related to Collaboration Awareness in collaborative modelling activities/tools, to propose solutions and to experiment them in order to validate them.

The remainder of the paper is organized as follows. In Section 2, we describe what collaboration awareness is and how it is currently handled by collaborative modelling tools. In section 3, we present a scenario that gives more details of what collaboration awareness is within such tools. The section 4 reports how we plan to experiment future results thanks to GenMyModel, a web-based collaborative modelling tool [1].

2 Awareness

2.1 Definitions

The most widely used definition of Awareness is given by Dourish and Bellotti [3] : *'... an understanding of the activities of others, which provides a context for your own activities'*. Schlichter [4] insists few years later on this aspect of CSCW in order to reach and to maintain effective coordination of collaborative work. In his review of awareness in Distributed Collaborative Software Engineering [5], Omoronyia defines five types of awareness as follows:

- Workspace awareness : *the up-to-minute knowledge of other participants interactions with the shared workspace*[6].
- Informal awareness : *the general sense of who is around, what they are doing, and what they are going to do*[1].
- Group-Structural awareness : *knowledge about people roles and responsibilities, their positions on an issue, their status, and group processes*[1].
- Social awareness : *Information about the presence and activities of people in a shared environment*[7].
- Context awareness : *The evolving internal and external state information that fully characterizes the situation of each entity in a shared environment* [5].

[1] www.genmymodel.com

These types refer to a series of questions people ask themselves when they collaborate. Table 1 draws a list of main ones. We illustrate some of these questions in the next section.

Element	**Description**	**Awareness**				
		W	I	GS	S	C
Identity	Who is participating in the activity?	x	x			
Location	Where are they?	x				
Activity Level	How active are they in the workspace	x				
Actions	What are they doing?	x				
Intentions	What are they going to do?		x			
	Where are they going to be?		x			
Changes	What changes are they making?	x				
	Where are the changes being made?	x				
Objects	What objects are they using?	x				
Extents	What can they see?	x				
Abilities	What can they do?	x		x		
Sphere of influence	Where can they have effects?	x		x		
Expectations	What do they need to do next?	x				
Availability	Are they busy, available, can they be disturbed?				x	
Activity History	Which other entities have been involved in the activity?					x
Activity Times	At what times did the activity take place?					x
Activity Duration	How long did the activity last?					x
Concurrent Activities	What concurrent activities were entities involved in?					x

**W : workspace, I : Informal, GS : Group-Structured, S : Social, C : Context*

Table 1. Awareness answers to collaboration raised questions based on [5]

Workspace awareness is the most studied in awareness literature [5] and it seems to be also the most important type in Distributed Collaborative Software Engineering. Even if awareness is the focus of lot of scientific works for decades, it is still a scientific issue. The volume of April 2013 issue of Computer Supported Cooperative Work [8] presents recent scientific results on awareness in different areas (distributed software development, model versioning, virtual team, social media...). Besides, new interesting direction of research relates to recent interactive technologies like large tactile surfaces. Yuill [9] reports case studies about awareness with such technologies. He points out that, for example, one of the originalities of multi-user interfaces is the presence of "mutual" awareness.

Whitehead shows that the trend concerning collaboration support is the arrival of web-based tools in every phase of software development [10]. Their main advantage is that they do not require any installation or configuration: teams are quickly ready to work. So, my first investigations will mainly concern web-based tools. I will analyze desktop tools later.

2.2 In current Collaborative model editor

I have found two web-based tools, they deal with workspace awareness but do not cover other awareness type. In his work on GEMSjax (a meta-modelling tool)[11],

Farwick discuss briefly about awareness. He resumes it to a chat and events to see which elements are currently selected but without give more details. A chat should cover all awareness type in processing content and filtering discussion to have more informations. But it seems this is not the case for GEMSjax. Chat is only used to discuss. Thum with his work on SLIM [12] is akin to Farwick work from awareness point of view. SLIM is an UML Case Tool for synchrounous collaborative modelling based on a lock system to avoid conflicts. When a user click on an element, a lock is put in place to prevent modifications by another user. Concerning awareness, SLIM contains a chat to discuss (content is not processed), with messages and current connected users, and padlocks on model elements locked by other users as seen in figure 1.

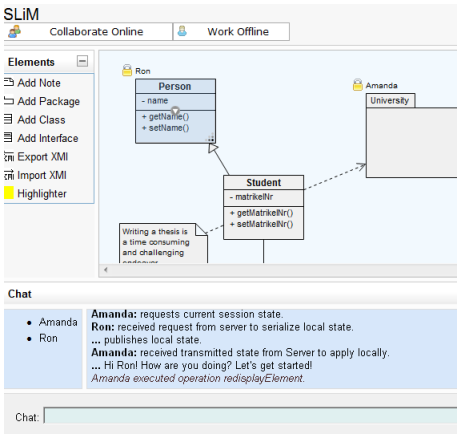

Fig. 1. SLIM awareness

The two found tools focus on synchrounous collaboration but awareness is not enough involved. As I'm PhD student since January 2013, related works have to be more studied and perhaps I will have to study awareness in other collaboration tool types. For the moment, my first work is to see if workspace awareness is the only necessary awareness type needed for collaboration in modeling tools.

3 Illustration of awareness in collaborative modelling

In this section, we illustrate with an example of model (subsection 1) and a related scenario, what kind of information UML practitioners may need about the actions of their co-workers in order to efficiently collaborate when they all work along the same model (subsection 2). We align such information on questions that we have previously mentioned when defining awareness (subsection 3).

3.1 Scenario

The sketch in this paper is an e-shopping project as figure 2 shows. The associated model, named *webcomm*, is composed of several diagrams which represent

the future website from different points of view: a use case diagram with general actions that a customer can do, a global class diagram for the application and a small class diagram which focuses on one functionality. The latter will be improved to put in place *Order* (We will suppose that this diagram is just created and will be refined). As we can see, class *Order* is shared between the two class diagrams.

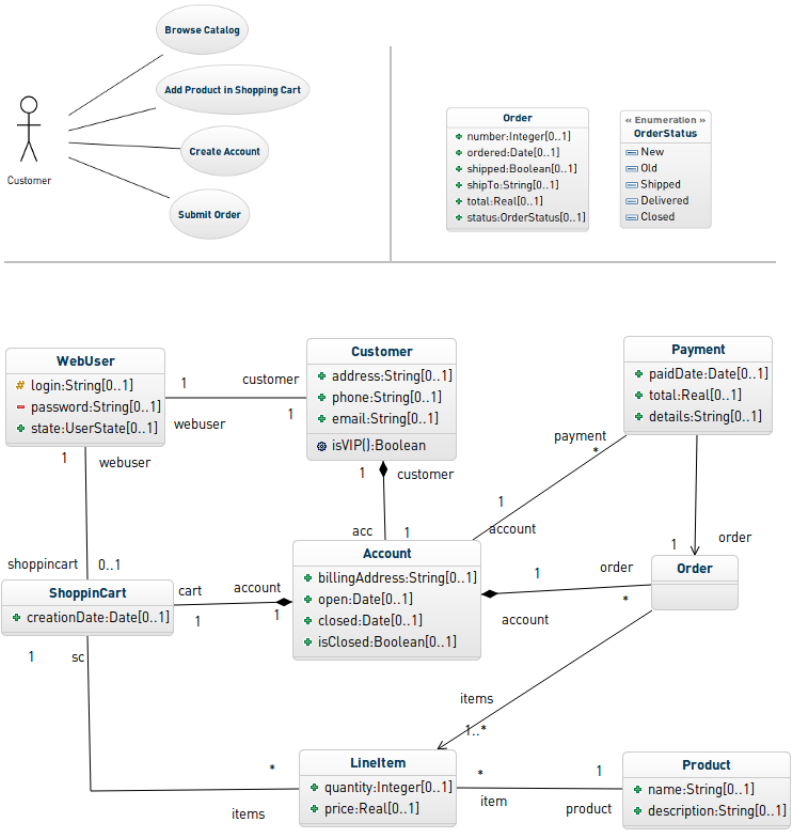

Fig. 2. Diagrams for webcomm model

This model and its diagrams are shared among several persons through an hypothetical collaborative UML case tool. We have two teams where first one (Alice, Bob and Ted) is in France and the other one (Jiao and Cheng) is in China as seen in figure 3. This implies that there is a time difference of 7 hours. When it is 9am at Paris, it is 4pm at Shanghai, that means the temporal window of real-time collaboration between the two teams is reduced. But even reduced, it exists and team members have to take advantage of it.

Figure 3 also shows actions of Bob, Alice and Jiao. We describe them in the next subsection.

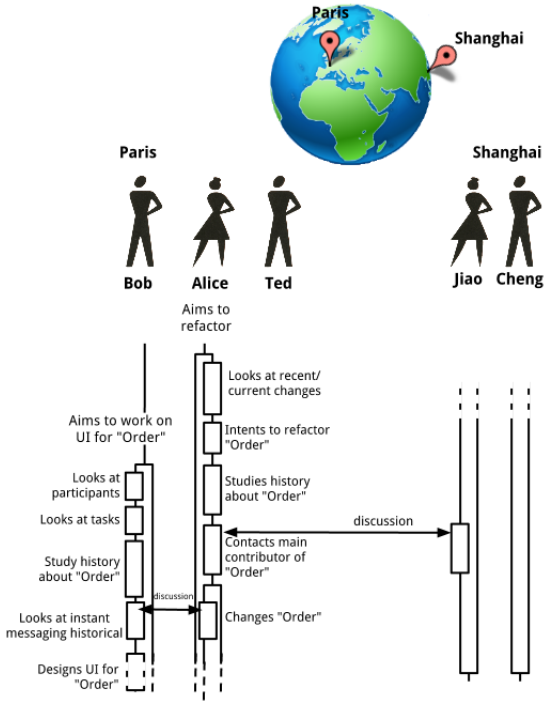

Fig. 3. Scenario

3.2 In action

First, Alice connects to the online model editor and opens *webcomm* model. She has in mind to do refactoring because, at last meeting, one of the discussed problems concerns some classes that becomes too big. So, she looks for diagram and last changes. She thinks that *Order* is a little bit big and may be reduced by dispatching *shipTo* into a new class *Address*. She wants to discuss with the main contributor of *Order* to know if the change she plans to do has sense. She analyses *Order* history to find this contributor. This is Jiao. So, Alice contacts her to discuss. Jiao approve Alice's idea. So, Alice applies changes about *shipTo*.

Meanwhile, Bob arrives at work. The goal of his current task is to create Graphical User Interfaces (GUI) for *Order*. So, he connects to the editor, and opens *webcomm* model. He looks for current changes and tasks to know if his work will impact others. He knows that Cheng and him use to work on the same concerns, and that Cheng usually does not like being disturbed by external changes. By looking the activities of the others, he sees that Cheng is working on *Customer* but does not have information about Alice and Jiao because they do not tell which activity they are working on. Thanks to instant messaging history, he knows that Alice and Jiao have discussed about *Order*. With changes history, he sees that Alice has made modifications on *Order*. He contacts her

to have in-depth explanations. However, refinement that Alice has made has no impact for him because he just begins to build the GUI and then he only focuses on the main features of *Order*. But if Alice modifies this class in the future, he will be notified through a visual notification about likely impacts. So, after his quick investigation, he can begin to work.

3.3 Scenario and awareness

This scenario highlights the fact that awareness-support is required to work in good conditions: each collaborator need many information about tasks that his/her co-workers aims to fulfill, actions they performed (or ones they plan to) or who currently participate to modelling. An instant messaging is not sufficient. Table 2 shows actions of the scenario for which the model editor has to provide related information to users in order they are efficiently aware of the whole activity of their team.

Action of scenario	**Awareness**					**Related element**
	W	I	GS	S	C	
List of present users	✓	✓				Identity
What users are doing	✓					Location/Activity Level/Actions
Prior modifications	✓					Activity Level/Changes
Current modifications	✓	✓	✓			Location/Activity Level/Actions/Changes/Sphere of Influence
Impact on elements Impact visualization	✓ ✓	 ✓	 ✓			Location Intentions/Objects/Sphere of Influence
Project involvement	✓					Activity Level
On what is collaborator working		✓				Intentions
Discuss for organization						
History					✓	Activity History/Concurrent Activities
Task understanding, need discuss if available				✓		Availability

**W : workspace, I : Informal, GS : Group-Structured, S : Social, C : Context*

Table 2. Needed information for collaboration in modeling tools

Scenario covers all awareness types but not all their refinements as seen in table 2. For uncovered refinements (*extents*, *abilities*, *activity time* & *duration*), we may easily foresee situations where they would be useful. For example, for *extents* and *abilities*, a client may have to give his/her feedback about model, so he/she can have a restricted access on certain parts with just the read and comment permission. For *activity time* and *duration*, we could considerate that each time a user performs a action, this one is associated with her/his current task.

3.4 Cognitive overload

Providing support for different types of awareness is a positive point for users to reach real-life collaboration. Nevertheless, displaying and streaming too much informations may cause them a cognitive overload [9]. Users should finally lose important information. So there are some display rules to define in order to avoid cognitive load. For example, the actions history may be filtered in different ways: hiding move actions (not always useful to understand model construction), grouping elementary actions in an more abstract one (maybe with pattern detection mechanism)...

4 Conclusion and Future Work

In this paper, we have seen that a software engineering project work could be a highly cooperative and distributed activity. Modelling tools are used during the development process in order to manage knowledge, coordinate activities, design a new application, maintain existing ones or discuss with their clients. To collaborate, engineers need information about others and their activities as identity, actions on model, intentions, changes. This information is related to awareness. Through literature, only few modelling tools support collaboration. Among them, only partial workspace awareness is supported while the scenario presented in the paper highlights that workspace awareness is not the only one required to provide a strong collaboration feature in UML case tools.

Axellience currently develops a such UML Case Tool named GenMyModel [2](see figure 4). GenMyModel is a web-based tool so it does not require any installation or configuration. It is planned to be an industrial product that will be able to support very large projects (in term of model elements and team size) with several innovative features (tablets or smart boards support, model transformation, code generation, round tripping...). Axellience has began to communicate about a beta-version of GenMyModel since January 2013, and there are already more than 1500 of users distributed in more than 90 countries although collaboration is missing. Some models already reach more than 50 classes elements and includes several hundreds of model elements. For GenMyModel users, collaboration is the most asked feature. They see it as a real advantage for a successful project. Thanks to this ecosystem and users needs, I will benefit from a realistic context where I will apply my validation process. This ecosystem is a strong side of my work.

Until now, I studied works about awareness in collaboration on tactile surfaces. From this study, I identified awareness types and needed information that seem to be required for collaboration in UML Case Tools. Then, I compared this information with those present in few UML web-based tools. For the next step, I must identify in the CSCW literature all scientific results which can be useful and mapped in our context. Then, a study will be realized on existing UML Case Tools (web-based and desktop). For each, I will observe if they have

[2] www.genmymodel.com

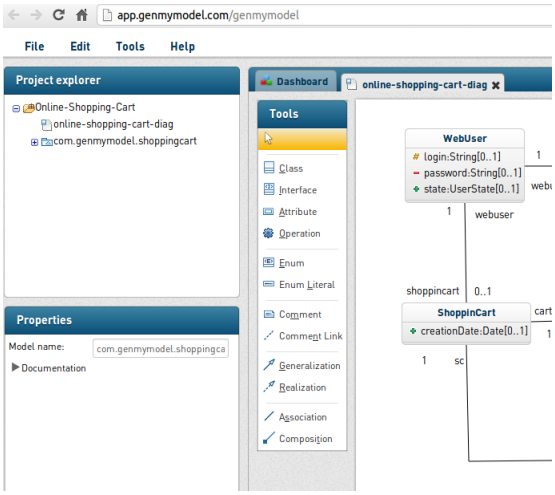

Fig. 4. GenMyModel

a collaboration mode and what kind (real-time, negotiation between users...). If any collaboration mode is supported, I will search to what extent the information highlighted by CSCW literature results exists inside and what kind of information is displayed to users. This study should be realized just after this summer.

After the list of tools which support awareness is established, I will discuss with communities to discover the reasons why they use this tool, get the advantages and disadvantages of each and what are the collaboration features they need. From this information, we will work with our users in an agile-like mode to deliver them expected features according to their need. For that, we will do a proposal, build a sketch by an implementation and provide it to the users to get their feedback. To do so, we will give volunteer users a customizable environment with those features. Their actions will be tracked to highlight which information is really needed during collaboration. After that, they will be sent a survey based on identified observations in order to confirm or not the observations. From this result, we will do a new proposal according to the figure 5. This cycle with users should begin in October.

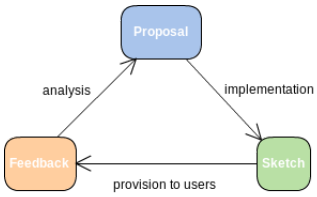

Fig. 5. Methodology

References

1. Gutwin, C., Greenberg, S., Roseman, M.: Workspace awareness in real-time distributed groupware: Framework, widgets, and evaluation. In: Proceedings of HCI on People and Computers XI. HCI '96, London, UK, UK, Springer-Verlag (1996) 281–298
2. Grossman, M., Aronson, J.E., McCarthy, R.V.: Does uml make the grade? insights from the software development community. Inf. Softw. Technol. **47**(6) (April 2005) 383–397
3. Dourish, P., Bellotti, V.: Awareness and coordination in shared workspaces. In: Proceedings of the 1992 ACM conference on Computer-supported cooperative work. CSCW '92, New York, NY, USA, ACM (1992) 107–114
4. Schlichter, J., Koch, M., Xu, C.: Awareness - the common link between groupware and community support systems. In Ishida, T., ed.: Community Computing and Support Systems. Volume 1519 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (1998) 77–93
5. Omoronyia, I., Ferguson, J., Roper, M., Wood, M.: A review of awareness in distributed collaborative software engineering. Software: Practice and Experience **40**(12) (2010) 1107–1133
6. Gutwin, C., Stark, G., Greenberg, S.: Support for workspace awareness in educational groupware. In: The first international conference on Computer support for collaborative learning. CSCL '95, Hillsdale, NJ, USA, L. Erlbaum Associates Inc. (1995) 147–156
7. Prinz, W.: Nessie: An awareness environment for cooperative settings. In Bdker, S., Kyng, M., Schmidt, K., eds.: ECSCW 99. Springer Netherlands (1999) 391–410
8. Kolfschoten, G., Herrmann, T., Lukosch, S.: Differentiated awareness-support in computer supported collaborative work. Computer Supported Cooperative Work (CSCW) **22**(2-3) (2013) 107–112
9. Yuill, N., Rogers, Y.: Mechanisms for collaboration: A design and evaluation framework for multi-user interfaces. ACM Trans. Comput.-Hum. Interact. **19**(1) (May 2012) 1:1–1:25
10. Whitehead, J.: Collaboration in software engineering: A roadmap. In: Future of Software Engineering, 2007. FOSE'07, IEEE (2007) 214–225
11. Farwick, M., Agreiter, B., White, J., Forster, S., Lanzanasto, N., Breu, R.: A web-based collaborative metamodeling environment with secure remote model access. In Benatallah, B., Casati, F., Kappel, G., Rossi, G., eds.: Web Engineering. Volume 6189 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2010) 278–291
12. Thum, C., Schwind, M., Schader, M.: Slima lightweight environment for synchronous collaborative modeling. In Schrr, A., Selic, B., eds.: Model Driven Engineering Languages and Systems. Volume 5795 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 137–151