Visual Mediation Mechanisms for Collaborative Design and Development

Carmelo Ardito[1], Barbara Rita Barricelli[2], Paolo Buono[1], Maria Francesca Costabile[1], Antonio Piccinno[1], Stefano Valtolina[2], and Li Zhu[2]

[1] Dipartimento di Informatica, Università degli Studi di Bari,
Via Orabona 4, 70125 Bari, Italy
{ardito,buono,costabile,piccinno}@di.uniba.it
[2] Dipartimento di Informatica e Comunicazione, Università degli Studi di Milano,
Via Comelico 39/41, 20135 Milano, Italy
{barricelli,valtolin,zhu}@dico.unimi.it

Abstract. Collaborative design involving end users has emerged as a response to the needs felt by various organizations of adapting software to specific environments and users. During time, users and environments evolve; this is another reason why software has to be modified. Different stakeholders, including consultants, designers internal to the organization and, recently, end users, have to collaborate among themselves, and possibly with the software providers, to shape software. Such stakeholders face fundamental challenges in learning how to communicate and in building a shared understanding. Researchers are now addressing such challenges. This paper contributes to this innovative research by formally defining visual mediation mechanisms for collaborative design. A case study illustrating their application is discussed.

Keywords: collaborative design, mediation mechanisms, end-user development, meta-design, communities of practice.

1 Introduction

The evolution of interactive software has determined the need of collaborative design involving stakeholders having different backgrounds and expertise. Software products must be modified in order to be adapted to specific environments and users. The software should also be modified at use time to comply with users' and environments' changes. Product developers already cooperate with end users and implementation consultants when developing or adapting software. But we are now assisting to a paradigm change in the way software is designed, developed, used and evolved. The report from a study commissioned by the US Department of Defense to define a research agenda on 'Ultra-Large-Scale Systems' (ULSS report) confirms this change and states that software will evolve during usage at all levels: system software, middleware, and the applications themselves [1]. Development activities will be distributed over the whole lifecycle and initiated by various stakeholders, including end users. The report also uses the term socio-technical ecosystem, which indicates the complex mutual dependencies among different parts of the software development

C. Stephanidis (Ed.): Universal Access in HCI, Part I, HCII 2011, LNCS 6765, pp. 3–11, 2011.

infrastructure. Researchers are now exploring different ways for effectively supporting the collaborative activities of the diverse stakeholders in such ecosystems. Particular attention is devoted to the development of software environments and tools, which enable end users to effectively participate in this collaboration and in adapting software at use time.

Recent developments based on the Web 2.0 and semantic Web, like weblogs, podcasts, RSS feeds, social software as wikis and social networking, are already examples of collaborative design environments that permit user-generated contents. Fischer refers to a new world based on cultures of participation, in which end users evolve from being passive software consumers to active producers [2], and are involved in various activities of end-user development [3].

In several development practices, communication and collaboration with end users take place through channels that are separated from the actual software, e.g., phone, e-mail, thus limiting end users' participation in collaborative design. Another main problem is that stakeholders are very diverse, characterized by different cultures and skills, they use different languages and notations, adopt different documentation styles, i.e. they belong to different Communities of Practice (CoPs). According to Wenger, we refer to a CoP as a group of people who share a common practice and address a common set of problems [4]. CoPs develop their own languages and notations to express and communicate their knowledge, problems and solutions. Examples of CoPs are software developers, software consultants, end users. CoPs involved in design, development and evolution of a certain software system represent a Community of Interest (CoI), defined in [5] as a community of communities brought together to solve a problem of common concern. CoIs stress the importance of combining voices from different CoPs; however, they face fundamental challenges in communicating among the CoPs and in building a shared understanding, which is the basis for their collaboration.

Members of the CoI keep collaborating at use time, whenever there is the need to modify or evolve the software; this forces the development of technical means to relate and integrate users' and developers' views in order to provide a seamless way of moving between use and design of software, facilitating its adaptation to users' needs and environments. Such technical means include new modeling languages, architectures that support multilevel design and development, but also mediation mechanisms, which permit the communication between professional developers' environments and end users' environments across the ecosystem. The novel contribution of this paper is the formal definition of visual mediation mechanisms supporting collaborative design. Their application to a case study referring to a Web portal, which advertises products of shops of various types, is also discussed.

The paper is organized as follows. Section 2 discusses the mediation process. Section 3 presents the formal specification of visual mediation mechanisms, and Section 4 illustrates how they are applied to a case study. Finally, Section 5 reports conclusions.

2 Mediation Process

A mediation process allows two human actors (shortly actors) to reach a common understanding, related to a specific domain, by the support of an agent, the mediator [6].

Wiederhold defined a mediator as "a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications" [7]. The concept of mediator has been used in the field of Web services to manage the interoperability among either software agents and Web services or Web services themselves [8], [9].

In the collaborative design context analyzed in this paper, the mediation process consists of exchanging messages between two actors playing a certain role in the collaboration [10]. These actors are generally members of different CoPs; they use dedicated interactive environments to reason on and perform their activities. Similarly to what is described in [11], such environments exploit an interaction language, which comply with the CoP notations, culture and role in the collaboration, in order to be usable for the CoP members. Each environment is equipped with an engine that acts as mediator by translating the incoming messages into the CoP's interaction language.

The first two steps of a generic mediation process are illustrated in Fig. 1. The human actor H1 sends a message ($Message_1$) to another human actor H2. Before reaching H2, $Message_1$ is captured and managed by "Mediator2" (the engine of H2's interactive environment that acts as mediator for it) that, by exploiting the knowledge Base (KB) for the current domain, translates it into the interaction language used by H2, so that H2 can understand it. The translated message, $Message_1'$, is then delivered to H2. In the second step represented in Fig. 1 H2 replies to H1's message by sending a new message ($Message_2$) to H1. In analogy with what happened in the first step, the message is captured and managed by "Mediator 1" (the mediator for H1's interactive environment) that, by exploiting the knowledge base, translates it into the interaction language used by H1. The translated message ($Message_2'$) is then delivered to H1.

The CoPs' environments support the actors involved in the communication process by allowing them to create and exchange boundary objects and their annotations on them. Boundary objects are artifacts of the interactive system and are part of the message sent by a human actor (a member of a CoP), but received and interpreted differently by another human actor according to the background and expertise of the CoP s/he belongs to [12]. Boundary objects are used all the time to sustain communication in face-to-face collaboration; for example, blueprints, sketches and drawings are used during discussion in design engineering; similarly, digital images

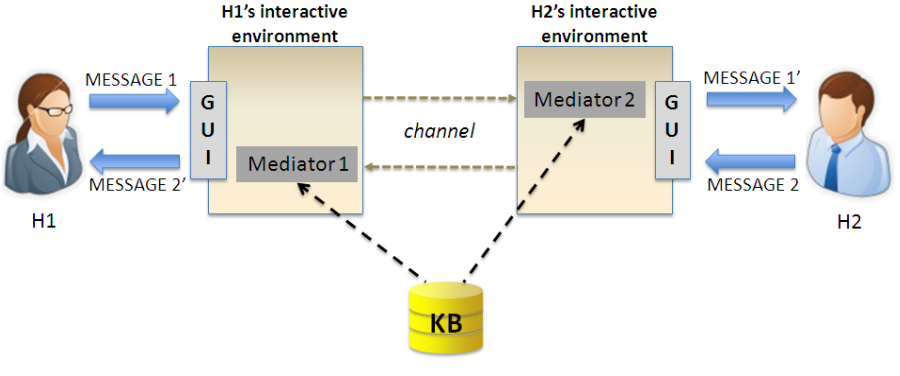

Fig. 1. Mediation process between two human actors (*H1* and *H2*)

are used during discussions in medicine and natural sciences. Discussants may express their annotations on a boundary object by using sticky notes.

Going to interactive software systems, boundary objects are used to support computer-mediated communication among members of CoIs, who are geographically distributed and work asynchronously on the same project. The diversity of the members of the CoI is addressed by making available boundary objects that are adaptable to the various cultures and contexts of use.

Members of the CoIs which constitute the design team interact and negotiate a concept by exchanging messages based on boundary objects as a concrete representation of what they mean. Boundary objects serve as externalizations that capture distinct domains of human knowledge and hold the potential to lead to an increase in socially shared cognition and practice [13], [14]. They carry information and context and can be used to translate, transfer and transform knowledge between CoIs members [12]. These objects are dynamic; they can be changed and manipulated to carry more information. In a collaborative design context, the effectiveness of a boundary object is directly related to how it is translated from tacit knowledge to explicit knowledge and translated back from explicit knowledge to tacit knowledge between different actors [15]. Since the information carried by boundary objects can be implicit, the annotations allow each actor to explicitly explain the modification s/he introduces in the boundary objects.

3 Mediation Mechanisms

The elements involved in a mediation process constitute a Mediation Mechanism (MM), defined as:

$$\text{MM} = (\textit{Mediator}, \textit{KB}, \textit{MVL})$$

where:

- *Mediator* is the agent that supports the two human actors in reaching an agreement through a mediation process;
- *KB* is the knowledge base accumulated in a specific domain in which the actors collaborate;
- *MVL* (Mediation Visual Language) is the visual language constituted by the set of messages exchanged by the two human actors by means of the Mediator. MVL is defined as follows:

$$\text{MVL} = \{\text{MVLmsg}_1, \ldots, \text{MVLmsg}_n\}$$

A MVLmsg is a message defined as

$$\text{MVLmsg} = \langle \text{data, metadata} \rangle$$

Data describe the object of the mediation; metadata specify some characteristics of the sending and receiving actors and of the digital platform used in the communication.

When an actor sends a message, a mediation process starts. Such message is constituted as follows.

- Data:
  - *EP*, an executable program, that is the software artifact that the two actors are collaboratively designing or part of it;
  - *A*, the annotation that the sender attached to the EP in order to communicate with the receiver;
- Metadata:
  - *S*, the profile of the human actor that acts as a sender;
  - *R*, the profile of the human actor that acts as a receiver;
  - *Pl*, the specification of the hw/sw platform being used to access the program

All messages following the first one convey the contributions of the involved actors. The metadata related to the profiles of the communicants are sent only in the first MVL message and not repeated in the next ones. In the messages, boundary objects are augmented with multimedia annotations to support negotiation among the different actors. Boundary objects and annotations are explicitly represented by MVL.

A mediation mechanism enables actors of different CoPs, working with different software environments, to properly communicate and collaborate within the ecosystem, since it makes concepts expressed in the language of a CoP understandable by the members of another CoP.

In [11], examples of message exchanges between actors of different CoPs, each working with the software environment specific for the CoP the actor belongs to, are provided. That paper also describes the Software Shaping Workshop (SSW) methodology, whose main idea is that all stakeholders, including end users, are "owners" of a part of the problem (e.g. software engineers are experts of technology, end users know the application domain, human-computer interaction experts deal with human factors). They all bring their own expertise into the collaboration and exchange ideas in order to converge toward a common design. Different software environments (called Software Shaping Workshops), are provided to each community of stakeholders (CoPs) in order to allow them to actively participate in system design, development and evolution. The case study illustrated in the next section has been designed according to the SSW methodology.

4 A Case Study: The Virtual Showroom

The case study described in this paper refers to a Web portal of a company that provides virtual windows for shops of various natures. Through its portal, the company sells advertisement spaces (the virtual shop windows) to shop owners, in order to allow them to advertise their goods. One of the novelties of the system is that such owners are allowed to create and manage the content of the virtual windows, so that they can update them as they like at any time. The virtual windows may be created using different templates, which are made available by the company and sold at different prices: the more the complexity (in terms of combined multimedia elements) grows, the higher the virtual window price becomes. Fig. 2 depicts a virtual window for a women's accessories shop, whose template is composed of a left side with textual description and a right side with a photo gallery showing four pictures.

Some stakeholders in this case study are grouped into four distinct CoPs:

- **Customers,** i.e. Web surfers, who interact with and/or browse the virtual shop windows;
- **Content providers,** i.e. shop owners, who provide contents to be shown in their virtual shop window(s);
- **Editorial staff**, who create virtual windows' templates to be used by the content providers;
- **Administrators**, who shape the software environment in which the editorial staff designs the virtual windows' templates.

Administrators, editorial staff, and content providers collaborate in designing the portal, by interacting with software environments specifically developed for them. Editorial staff and content providers are not required to be computer experts. On the other hand, administrators have to be familiar with the application domain and also should have some experience in Web development. The software environment used by the administrators is designed by professional software developers, but this is out of the scope of this paper so that this CoP is not considered here. The reader interested in the whole meta-design process can refer to [11].

In order to provide an example of a mediation process in this case study, the situation in which the content provider wants a feature modification in her/his virtual shop window is considered. As illustrated in Fig. 3, s/he uses the annotation tool available in the system to annotate that specific feature explaining the changes s/he wants. The feature is the photo gallery that is highlighted with a dashed border. Through this annotation, which in Fig. 3 overlaps the main picture, the content provider requests to the editorial staff a modification in the virtual shop window template in order to be able to show a higher number of pictures.

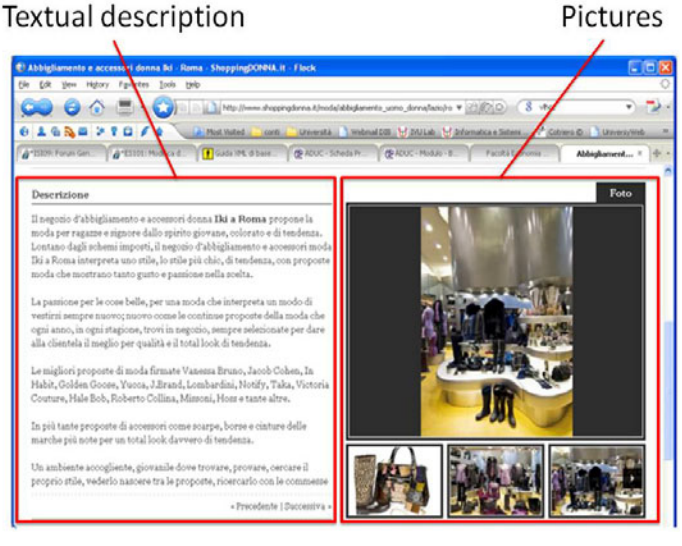

Fig. 2. An example of virtual window of a women's accessories shop. The template is composed of a textual description on the left and a photo gallery on the right.

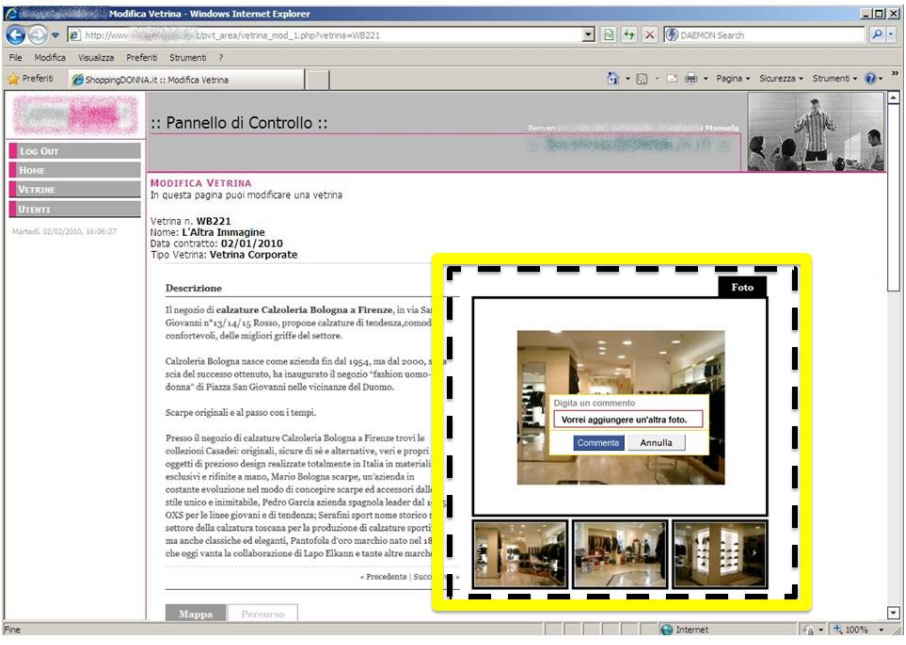

Fig. 3. An example of annotation containing a request of the content provider. The photo gallery is highlighted by surrounding it with a dashed border and the annotation explaining the requested modification is overlapped on the main picture.

A mediation process is thus activated. The mediation mechanism involved in this process consists of the three components defined in Section 3; specifically, *Mediator* is an engine that is part of the environment used by the content provider to make her/his requests of changing the template of the virtual shop window; *KB* contains all the information necessary to translate the content of the message (i.e. EP and A) in a specific request for the editorial staff; *MVL* is the visual language composed by the messages exchanged during the whole mediation process between content provider and editorial staff. The first message is constituted as follows.

- Data:
  - *EP*, the virtual shop window environment that the content provider is asking to modify;
  - *A*, the annotation that the content provider attached to the EP in order to communicate her/his request to the editorial staff;
- Metadata:
  - *S*, the profile of the content provider (the sender in this mediation process);
  - *R*, the profile of the editorial staff (the receiver in this mediation process);
  - *Pl*, the specification of the hw/sw platform used to access the Web portal.

The receiver, a member of the editorial staff, gets the message translated according to the language of her/his software environment, which is customized to the needs of that CoP. As shown in Fig. 4, the content of the message is rendered in a table at the bottom of the screen, which communicates the same meaning intended by the content provider who sent the message, but using a different visual representation, in which some codes are used, that the editorial staff well understands.

If the editorial staff can directly manage the request, s/he performs the necessary software modification and communicates it to the content provider; otherwise, s/he activates a second mediation process with a member of the administrator CoP, to whom s/he reports the request for modifications. In the first mediation process, the reply message to the content provider consists of the virtual shop window application *EP*, modified according to the content provider request, and the annotation *A* created by the editorial staff to explain the performed modifications. If the content provider is satisfied by the solution s/he gets, the mediation process is concluded; otherwise it keeps going on iteratively until the content provider and the editorial staff reaches an agreement.

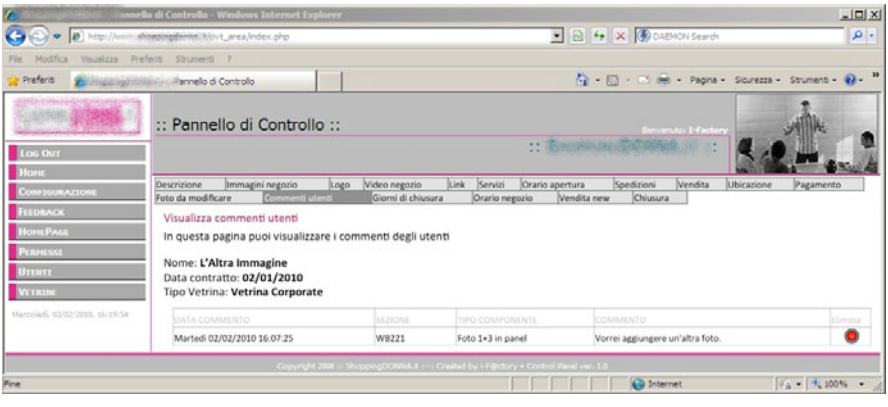

Fig. 4. Editorial staff environment: the request sent by the content provider is shown

5 Conclusions

This paper has discussed and provided a formal definition of visual mediation mechanisms for collaborative design, development and evolution of software. Mediation mechanisms provide a means to improve communication and cooperation among all stakeholders involved in the design of software artifacts, including end users. This communication is fundamental in order to cooperatively create software adapted to user needs and context of work.

A case study referring to a Web portal that provides advertisement for shop of various natures has been presented; it provides an example of how visual mediation mechanisms are used to permit the collaboration among the different stakeholders.

Acknowledgments. This work is supported by Italian MIUR and by grant PS_092 DIPIS. Li Zhu acknowledges the support of the *Initial Training Network* "Marie Curie

Actions", funded by the FP 7 - People Programme with reference PITN-GA-2008-215446 "DESIRE: Creative Design for Innovation in Science and Technology". The authors thank Nicola C. Cellamare for his collaboration in the case study.

References

1. Software Engineering Institute: Ultra-Large-Scale Systems: The Software Challenge of the Future, http://www.sei.cmu.edu (last access on February 22, 2010)
2. Fischer, G.: Cultures of Participation and Social Computing: Rethinking and Reinventing Learning and Education. In: Ninth IEEE International Conference on Advanced Learning Technologies (ICALT), pp. 1–5. IEEE Computer Society, Riga (2009)
3. Lieberman, H., Paternò, F., Wulf, V. (eds.): End User Development, vol. 9. Springer, Dordrecht (2006)
4. Wenger, E.: Communities of Practice: Learning, Meaning, and Identity. Cambridge University Press, London (1998)
5. Fischer, G.: Extending Boundaries with Meta-Design and Cultures of Participation. In: 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (NordiCHI 2010), pp. 168–177. ACM, Reykjavik (2010)
6. Boulle, L.: Mediation: Principles, Process, Practice. LexisNexis Butterworths, Chatswood (2005)
7. Wiederhold, G.: Mediators in the Architecture of Future Information Systems. Computer 25(3), 38–49 (1992)
8. Burstein, M.H., McDermott, D.V.: Ontology Translation for Interoperability among Semantic Web Services. AI Magazine - Special issue on Semantic Integration 26(1), 71–82 (2005)
9. Vaculin, R., Neruda, R., Sycara, K.: The Process Mediation Framework for Semantic Web Services. International Journal of Agent-Oriented Software Engineering 3(1), 27–58 (2009)
10. Zhu, L., Mussio, P., Barricelli, B.R.: Hive-Mind Space Model for Creative, Collaborative Design. In: 1st DESIRE Network Conference on Creativity and Innovation in Design, Desire Network, Aarhus, Denmark, pp. 121–130 (2010)
11. Costabile, M.F., Fogli, D., Mussio, P., Piccinno, A.: Visual Interactive Systems for End-User Development: A Model-Based Design Methodology. IEEE T. Syst. Man Cy. A 37(6), 1029–1046 (2007)
12. Carlile, P.R.: A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development. Organization Science 13(4), 442–455 (2002)
13. Jennings, P.: Tangible Social Interfaces: Critical Theory, Boundary Objects and Interdisciplinary Design Methods. In: 5th Conference on Creativity & Cognition (C&C), pp. 176–186. ACM, London (2005)
14. Resnick, L., Levine, J., Teasley, S.: Perspectives on Socially Shared Cognition. American Psychological Association, APA (1991)
15. Fong, A., Valerdi, R., Srinivasan, J.: Boundary Objects as a Framework to Understand the Role of Systems Integrators. Systems Research Forum 2(1), 11–18 (2007)