Towards a Framework for the Development of CSCW Systems

Sergio López Antonaya and Crescencio Bravo Santos

School of Computer Science, University of Castilla - La Mancha,
Paseo de la Universidad, 4-13071 Ciudad Real, Spain
SergioL@jccm.es, Crescencio.Bravo@uclm.es

Abstract. This paper proposes a conceptual and technological framework for the improvement of the development processes and the quality of collaborative systems, starting from a previous study of the most notable lacks identified in the field of CSCW regarding implementation issues, and the analysis of all the inherent aspects of organizational contexts such as internal structures, resources management or workflow definitions.

Keywords: Groupware Applications, Reusability, Tailorability, Model-Driven Development, Service-Oriented Architecture.

1 Introduction

Although collaborative systems have increased their presence in our lives through the integration of diverse tools, such as instant messaging, shared calendars or popular Web 2.0 technologies, into many different environments [1], the ultimate goal pursued by the CSCW field of study of transforming the work carried out with computers from an individual use to a collective one, still appears as a difficult achievement in a distant future. The effort and costs involved in the development of a groupware application is still very high compared to traditional software [2, 3], although some proposals [4, 5] have provided theoretical and practical guides to simplify these development processes and improve the quality of final products, dealing with not only the technical complexity of such systems, but also the psychological and social aspects identified [6]. However, the problem is far from being completely solved.

New software development paradigms like Model-Driven Development (MDD) or architectures like Service-Oriented Architecture (SOA) provide new valuable tools in regard to the building of CSCW systems [7]. On the one hand, MDD provides a higher degree of abstraction in the development of software and different views for a system, depending on the interest of each user or participant. On the other hand, SOA architectures provide high maintainability, reusability and tailorability due to the inner nature of the Web services in which they are based on. Both technologies allow developers to simplify the development processes, and users to modify and adapt the software environments in which they are immersed when working.

Y. Luo (Ed.): CDVE 2010, LNCS 6240, pp. 117–120, 2010.

2 Framework and Case Study

The conceptual framework proposed in this paper, allows to specify any type of CSCW system for any kind of organizational environment such as universities, corporations, governments or hospitals, considering the use of high level of abstraction models and dividing a full system into three independent and complementary subsystems (Figure 1) similar to other works in literature [8, 9, 10]. The system's s*tructure* is defined in terms of *organizations*, *societies*, *groups* and *roles*. The *behaviour* is expressed through g*oals*, *activities* and *tasks*. In the *instrumentation* subsystem, g*roupware applications* and different technological r*esources* are used.

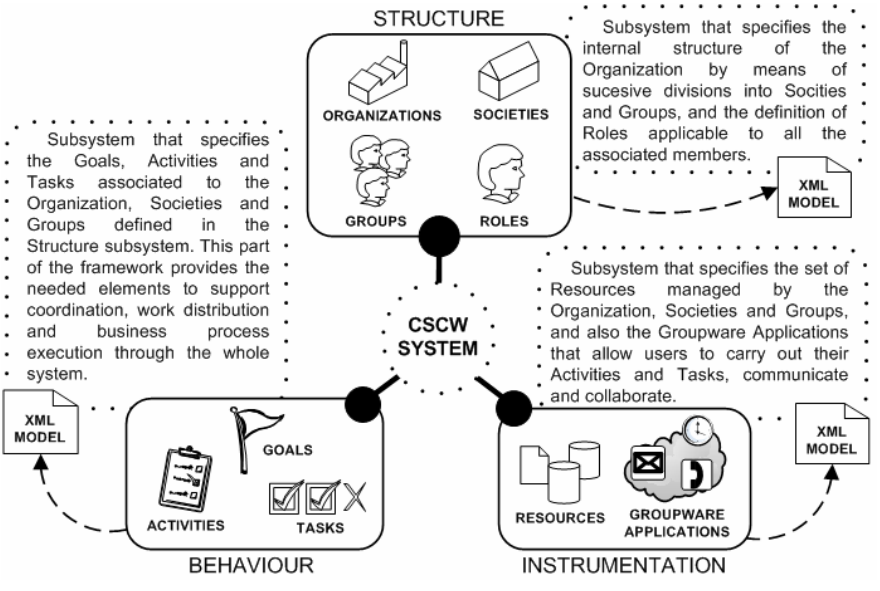

Fig. 1. CSCW Subsystems according to the proposed conceptual framework

Like other environments or specific-purpose libraries for the development of groupware applications [11, 12], our proposal also defines a technological framework that helps simplify the building processes of new CSCW systems, and to ensure high levels of reusability and tailorability in the final implementations, something specially relevant for the specific case of collaborative software, as appointed in several studies [13]. The proposed technology framework provides a Client/Server architecture in which communications between the users are carried out by exposing and invoking distributed Web Services.

The framework additionally provides on-the-fly mechanisms for the analysis, interpretation, compilation and execution of new groupware applications by means of dynamic source code generation tools and internal reflection properties that allow user interfaces and business logic to be altered in real time, just replacing or modifying the XML specification files [14] for the s*tructure*, *behaviour*, and *instrumentation* subsystems (see Figure 1).

We illustrate all these concepts in the case study of health care area. Hospitals are well-defined organizational entities with a clear overall *goal*: provide a good health care system to citizens. Additionally, a hospital can be seen as a more complex structure consisting of several *societies* dedicated to *activities* associated with the different areas of current medicine: Cardiology, Oncology, Neurology, etc. At the same time, each of these *societies* can be divided into different *groups*, composed of individuals with specific *roles* such as doctors or nurses, who perform more concrete *tasks* such as diagnosis or surgery interventions. Thus, we can realize how a common organizational environment as a hospital can be described and modeled by elements defined in the subsystems of *structure* and *behaviour* of the proposed conceptual framework.

On the other hand, the *instrumentation* subsystem provides the *resources* and *groupware* applications needed for the execution of the *tasks* assigned to each *group*. For instance, a *group* composed of three doctors involved in diagnostics *tasks*, may need to check a patient's medical history at a given moment from distant locations of a hospital. An initial *groupware* application for this simple purpose could be made up of a documentation viewer and a video-conference component, which would be appropriate enough for this kind of context. However, if one of the doctors would decide to replace the video-conference component with instant messaging due to privacy issues, he would only have to replace the *instrumentation* XML specification (see Figure 1) that he was using with a suitable one, in order to change the user interface and the functions provided by the system without altering the final functionality offered.

Figure 2 shows the explained runtime tailorability features and how all the technologies exposed are used to fulfill the usability and tailorability requirements.

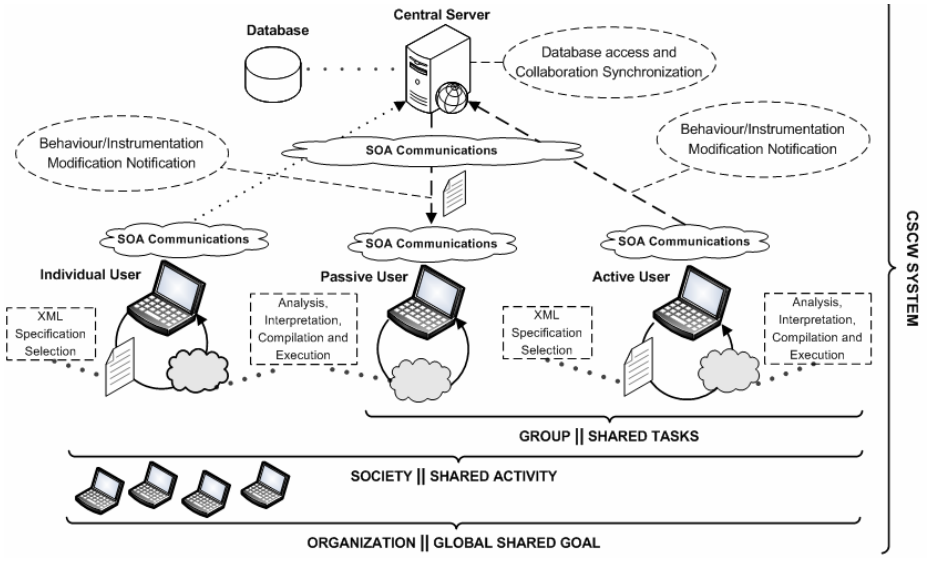

Fig. 2. Runtime tailorability features for groupware applications

3 Conclusions and Future Work

This article has established the principles of a new development framework by means of a conceptual model and a technological architecture that cover all aspects and fundamentals involved in CSCW systems development processes, including groupware applications, taking into account the characteristics of organizational internal structures or the development of business processes, and facilitating the proper levels of maintainability, reusability and tailorability for this kind of software.

The work to be performed in the immediate future will focus on formalizing the entire conceptual framework by means of ontologies and meta-models, developing a visual modeling language and devising a suitable development methodology for CSCW systems based on the exposed framework.

References

1. Rodden, T.: A Survey of CSCW Systems. Interacting with Computers 3 (1991)
2. Grundy, J., Graham, T.C.N.: External Requirements of Groupware Development Tools. Engineering for Human-Computer Interaction 1998 (1998)
3. Wu, J., Graham, T.C.N.: Toward Quality-Centered Design of Groupware Architectures. In: Engineering Interactive Systems. Springer, Heidelberg (2008)
4. Garrido, J.L., Noguera, M.: Un Esquema Basado en Ontologías para la Especificación de Sistemas Cooperativos. XV Jornadas de Ingeniería del Software y Bases de Datos (2006)
5. Molina, A.I., Ortega, M.: CIAM: A Methodology for the Development of Groupware User Interfaces. Journal of Computer Science 14(9) (2008)
6. Grudin, J.: Groupware and Social Dynamics - Eight Challenges for Developers. Communications of the ACM 37(1) (1994)
7. Bravo, C., Gallardo, J.: Developing Modeling Systems Following a Model-Driven Engineering Approach. In: 7th International Workshop on Software Architectures (2008)
8. Gutwin, C., Pinelle, D.: Task Analysis for Groupware Usability Evaluations: Modeling Shared-Workspaces Tasks with the Mechanics of Collaboration. ACM Transactions on Computer-Human Interaction 10(4) (2003)
9. Bravo, C.: Protocolos de Coordinación y Comunicación Estructurada en Entornos CSCL. VII Simposio Internacional de Informática Educativa (2005)
10. Gutwin, C., Greenberg, S.: The Mechanics of Collaboration – Developing Low Cost Usability Methods for Shared Workspaces. In: IEEE 9th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (2000)
11. Chabert, A., Grossman, E.: Java Object Sharing in Habanero. Communications of the ACM 41(6) (1998)
12. Roseman, M., Greenberg, S.: Building Real Time Groupware with Groupkit, A Groupware Toolkit. ACM Transactions on Computer Human Interactions (1996)
13. Slagter, R., Biemans, M.: Evolution in Use of Groupware: Facilitating Tailoring to the Extreme. In: Proceedings of the 7th International Workshop on Groupware (2001)
14. Bravo, C., Gallardo, J.: Defining Tasks, Domains and Conversational Acts in CSCW Systems: the SPACE-DESIGN case study. Journal of Universal Computer Science 4 (2008)