Science of Computer Programming 89 (2014) 320–349

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

Model-driven development of interactive groupware systems: Integration into the software development process

Ana I. Molina [a,*], William J. Giraldo [b], Manuel Ortega [a], Miguel A. Redondo [a], César A. Collazos [c]

[a] Department of Information Technologies and Systems, University of Castilla–La Mancha, Spain
[b] System and Computer Engineering, University of Quindío, Quindío, Colombia
[c] Departamento de Sistemas – FIET, University of Cauca, Colombia

HIGHLIGHTS

- The design of interactive groupware systems is a complex and multidisciplinary task.
- Different *stakeholders* and viewpoints should be supported.
- We propose to integrate CIAM (for collaboration and interaction modeling) and OpenUp (for functionality modeling).
- We provide a theoretical and computational solution to this necessity.
- With this approach all modeling viewpoints and stakeholders can be gathered together using a unique environment.

ARTICLE INFO

Article history:
Received 18 January 2012
Received in revised form 27 February 2014
Accepted 27 February 2014
Available online 12 March 2014

Keywords:
Software Engineering (SE)
Human–Computer Interaction (HCI)
Groupware design
Model Driven Development (MDD)
Models integration

ABSTRACT

Modeling interactive groupware systems is a complex and multi-disciplinary task. It is necessary to provide designers and engineers with a set of methods, notations and tools to specify the different aspects to consider when designing this type of systems. In this work we present a methodological framework based on the integration of several notations and processes for modeling some of these aspects, in particular: interaction, collaboration and functionality. The objective of this work is to provide a more complete support to the design of groupware systems, considering different viewpoints and modeling perspectives of the several stakeholders involved in the development of such applications.

1. Introduction

Traditionally, the development of systems supporting group work has been considered a complex task [1,2]. In addition, and in order to support effective collaborative work it is also necessary to provide usable user interfaces and promote *group awareness* [3]. The design of both aspects (group work and human–computer interaction issues) is a multi-disciplinary process that may involve experts in various fields of knowledge (*stakeholders*): graphic designers, computer engineers, ethnographers, sociologists, etc. [4]. Each specialist makes use of specific terminology and notations of their work. For example, a usability engineer's main design and communication artifact is a set of models in CTT notation [5]. On the other hand, functional aspects of the system can be addressed by software engineers using the UML standard. Each *stakeholder* has its

* Corresponding author.
E-mail addresses: AnaIsabel.Molina@uclm.es (A.I. Molina), wjgiraldo@uniquindio.edu.co (W.J. Giraldo), Manuel.Ortega@uclm.es (M. Ortega), Miguel.Redondo@uclm.es (M.A. Redondo), ccollazo@unicauca.edu.co (C.A. Collazos).

http://dx.doi.org/10.1016/j.scico.2014.02.030
0167-6423/

own perspective or point of view regarding the analysis and modeling of the final system, and prefers the use of a concrete specification technique (or notation). Therefore, it is necessary to provide these multi-disciplinary groups with tools to support their work. These tools could enable designers to specify their models in different notations, collecting and mapping the common information provided by each.

As mentioned above, if we want to address the comprehensive modeling of a groupware system we must consider multiple aspects [1]: functionality, distribution, usability, security [6], among others. In this work we address the modeling of some of them, in particular: interaction, collaboration and functionality. We only consider these three aspects, because they cover part of two layers of the "three-tier architecture" when creating software [17]. Using our proposal we try to support the presentation layer (interaction), the processing layer (functionality) and the collaboration (due to we focus on groupware systems). We address these aspects, at the moment, but in the future we plan to cover additional aspects.

Regarding the specification of these aspects, it is necessary to mention that we find certain problems or needs. UML, widely used by software engineers, addresses the design of the functional part of applications, leaving aside the aspects related to human–computer interaction (HCI) [7] or person-to-person interaction (supported by *groupware* systems). There are some proposals that address the modeling of the interactive aspects of the systems [8], but that do not contemplate the group work as one of their requirements. We note, therefore, that modeling of these three aspects (functionality, interaction and group work) have been addressed separately from different areas of study, but not in a holistic way.

For supporting a more complete modeling of interactive groupware systems we can adopt two approaches: (a) define a new dedicated notation (that allows modeling of the three aforementioned aspects) or (b) reuse existing and complementary notations, supporting the mapping and integration between them. In this work we adopt the second approach. In this way, each stakeholder can use the most suitable notation for designing a particular issue (UML for processing layer, CTT for human–computer interaction issues, ...). We base this choice on one of the principles provided by Moody [9]. This author proposes a set of principles for designing cognitively effective visual modeling notations. These principles are based on theory and empirical evidence. One of these principles is the called "*Principle of Cognitive Fit*" that proposes the use of different visual dialects for different tasks and audiences. This principle is based on the *cognitive fit theory*, a widely accepted theory in the information systems field, which has been validated in a wide range of domains [10–12]. In contrast with most Software Engineering notations, that exhibit visual monolinguism (they use a single visual representation for all purposes), cognitive fit theory suggests that different visual representations may be required for different tasks and different audiences [13]. This implies the use of complementary visual dialects. Thus, in this work we propose a model-driven framework for development of interactive collaborative systems based on the integration of different visual dialects or notations [14]. Thus, the various *stakeholders* involved in the development of these applications can use the most suitable notation for each of them (in their own *workspace*).

As we have indicated, in this work we address the modeling of functionality, collaboration and interaction. In order to support both interaction and group work design jointly we propose a methodology called CIAM [15] and a notation called CIAN [16]. This methodology focuses on the design and development of the presentation layer [17] (that is, the user interface) of these applications, but it does not take into account the design of their functional aspects (processing layer). For modeling of functionality we can use UML. UML and CIAN have different purposes but they can be considered as complementary. There are some modeling elements interrelated in both notations, being necessary to have certain coherence and mutual feedback between the models specified by using each of them. This idea of integration, classification and common information collection is expressed graphically in Fig. 1. In this scenario it is also necessary to provide computational support to classify and map the information provided by each of the notations.

Therefore, the main **objective** in this work is to study the integration of specifications to create in the CIAM framework and the conventional software development process. We will analyze the relationship between these models and we will deal with the technological and computational support for editing and integrating their notations (CIAN and UML). Additionally, we will address the relationship between their processes and we will explore their integration. This process integration should also be computationally supported. The overall process proposed will support the design of three of the aspects to consider in an interactive *groupware* system: functionality, interaction and collaboration. This proposal mainly aims to reduce the *gap* between CIAN (a proposal from the HCI area) and the UML standard (widely used in Software Engineering) [18,19].

This article is structured into the following sections: first, we present a review of proposals that address the modeling of group work and interactive aspects. A brief review of the proposals addressing the integration of notations belonging to the Human–Computer Interaction (HCI) and Software Engineering (SE) fields follows. Section 3 focuses on presenting one of the two methodological approaches taken as a starting point in our proposal of integration (CIAM). In the next section we explain our proposal of the integration of CIAN and UML notations and the corresponding processes (CIAM and a software development process). To do this, we present the stages followed in our integration proposal as well as its computational support (Section 5). Finally, we discuss the conclusions arising from this work and the future work derived from it.

2. Related works

This section begins with a brief review of proposals for modeling group work and human–computer interaction aspects. The following will address the problems of integration of proposals from SE and HCI, and the main approaches and contributions in this field.