A Software Architecture for Defining a Methodological Approach to Develop Collaborative Applications

Mario Anzures-García[1], Luz A. Sánchez-Gálvez[1], Miguel J. Hornos[2],
Patricia Paderewski-Rodríguez[2]

[1] Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Computacion, Puebla,
Mexico

[2] Universidad de Granada, Departamento de Lenguajes y Sistemas Informáticos,
E.T.S.I. Informática y de Telecomunicación, Granada,
Spain

{mario.anzures, sanchez.galvez}@correo.buap.mx, {mhornos,patricia}@ugr.es

Abstract. This paper presents a software architecture-based methodological approach to develop collaborative applications. Today, the use of collaborative applications has spread to various domains, as they facilitate communication, collaboration, and coordination between several users. These applications require mechanisms to support and model communication activities and processing of information, vital in the dynamic nature to the group. In this paper, the use of a software architecture is recommended to develop collaborative applications. This architecture for specifying the structure and behavior through the application, providing a shared meeting space to simplify and agile the group work. Thus, it is possible to support dynamic group structure. In addition, specification tables are proposed to simplify the development of this kind of applications; since the developers to complete the table are analyzing the necessary elements required to build an application, so performing requirements analysis, design, and displayed as would the final application. A case study to validate the software architecture is proposed.

Keywords: Methodological Approach, Software Architecture, Collaborative Applications, Specification Tables, Group Work.

1 Introduction

The use of a software architecture allows us to have a global perspective of the software applications, since it knows its components what do them and how are related, as well as, the environment in which interacts these. This knowledge leads us to identify and analyze the necessary components of the application to develop. In this way, it is possible to get any application requirements. Therefore, it can be handled to the development of collaborative applications, which provides a shared interface that allows a group of people to achieve a common goal. Consequently, in this paper, a collaborative application for managing the departmental tests is developed as a case study to implement a software architecture.

The necessary elements to create a collaborative application are specified by this model in four layers; which provide four essentials aspects: the group, the cornerstone of the group work; the interaction to control and manage the shared objects to the application and between different users of the group: the application presents several views to visualize the interaction carried out by the group in the stages that application contains; and the adaptation to adjust the application with respect to the produced changes through group interaction.

In order to facilitate the development of the collaborative applications, this model supplies specification tables, so it is possible to define which elements will have the application of an intuitive manner, even this can be made by any inexperienced person in this domain. Thus, this model can be used to specify requirements, to outline the design and implementation.

These requirements identified in the table inform how the application elements will be distributed and executed in each involved stage in this. In this paper, the table elements are the base of the requirements analysis, since each they are part of the application for managing the departmental tests, and therefore, these determine the design and implementation of the same. Thus, the software architecture can be used how a methodological approach to develop this type of applications. This approach is made up by four parts: requirements specification, sketch creation, code production, and application test.

The rest of the paper is organized as follows: Section 2 describes briefly the collaborative applications; Section 3 explains the used software architecture, and the derived specification table of the same; Section 4 presents the case study, in which software architecture is implemented using a methodological approach for building a collaborative application for managing the departmental tests. Section 5 outlines the conclusions and future work.

2 Collaborative Applications

A collaborative application is a computer-based application that supports a group of people to achieve a common goal and provides services to support the work of users through a shared environment interface [1]. Collaborative applications provide the shared workspace, where they will perform group work; therefore, it must provide the communication, collaboration, and coordination of the users. Different terms to denote the shared workspace have been used, such as conversations [2], local [3], places [4], spaces [5], conferences [6, 7], and meetings [8, 9]. In general, all these terms denote a group of individuals, geographically distributed, which share a common interest to perform common tasks. In this paper the term session to denote the shared workspace is used.

Collaborative applications provide a mechanism to control and manage sessions, called session management, which allows you to define sessions via a user interface, through which users establish a connection; that is, users to join, leave, invite someone to, and exclude someone from a session. Generally, these mechanisms only specify how the group work will be organized. However, it is important to support and define different styles for group work. Thus, if the style imposed by the system is accepted or unsuitable for group work, you should be changed to one that meets your needs. For this reason, this model uses an ontology to model the session management policies [10] that allows to support different styles of group work.

A variety of tools (such as Groupkit [11], ANTS [9], and SAGA [12]), architectures (e.g., Clock [13], and Clover [14]), and methodologies (AMENITIES [15], ClAM [16], and TOUCHE [17]), which allows to develop collaborative applications. However, these do not specify the steps to develop this kind of applications, and they are not flexible enough to adjust to the group changing needs.

3 Software Architecture

Software architecture is defined as the fundamental organization of a system, embodied in its components, their relationships to each other, to the environment, and the principles governing its design and evolution [18]. A variety of architectural styles, can be identified in a software architecture. A style is each recognized generic pattern in relation to systems group; of another manner, a style describes and provides the basic property of an architecture, as well as; it establishes the limits for its evolution. One example of architectural style is a layered style, which is organized hierarchically, and it is characterized by a sense of development "bottom-up", so that lower layers provide resources that are used by upper layers, according to their particular needs. A layer is a software technique for structuring the software architecture that can be used to reflect different abstraction levels in the architecture.

A layered style is ideal for supporting the development of collaborative applications, since it leads to break down a complex problem into a set of smaller problems and simpler to solve. Therefore, in this paper a layered software architecture will be used to develop the distributed components of a management system for departmental tests.

The layered software architecture (see Figure 1) has been derived from performed analysis about: Task Analysis [19], Activity Theory [20], Coordination Theory [21], Conceptual Model [22]; and Distributed Cognition [23]. These related works supply a set of ideas and concepts to manage the group interaction of the collaborative applications. Fundamentally, these studies consider four principal aspects in a collaborative application: group, their interaction, the application itself, and its adaptation. Therefore, the software architecture contains four layers: Group Layer, Interaction Layer, Application Layer, and Adaptation Layer.

The first aspect is a key to the performance of the work carried out to achieve the common goal. The group must present an organizational structure to support the *division of labour*, which indicate the actions that the group members (users) should make in relation to the established roles for each of them. This organization must be governed by a police, which defines the roles (Role) that users can play. These roles establish the set of rights/obligations (R/O) and status (St) of the user; whom can execute tasks (T), which are comprised of Activities (A) that use the prevailing shared resources (R).

The second aspect is elemental to provide the communication, coordination, and collaboration between the users. Accordingly, it must establish the session (Ss), which is the shared workspace where the interaction is carried out. Furthermore, it must make available for the awareness group and group memory through a notification (Nt) mechanism, which informs users of and registers every change in the shared resources used in each activity. Finally, it must ensure the consistency of the resources being shared, facilitating the manipulation of the users' permissions, which are granted, in accordance with established organizational structure, and a concurrency (Cc) mechanism.

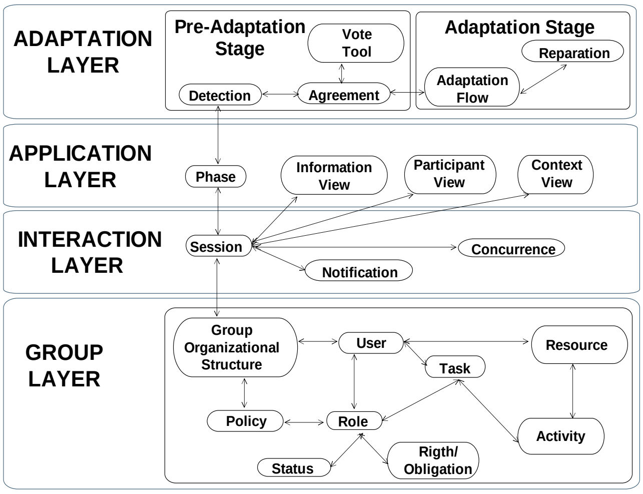


Fig. 1. Layered software architecture for building collaborative applications.

The third aspect allows us to show the generated information and interaction in the collaborative application. This is presented in stages (they are defined as each of the collaboration moments [22]) on views (which are user interfaces). Three views are considered in this model: Information View (IV) that displays the user information, Participant View (PV), exhibiting the changes in shared resources and, therefore, provide group awareness, and Context View (CV) shows the group memory, i.e., the change history of shared resources.

In the fourth aspect, the views are adjusted to produced changes by interaction between users and of these with the own application. For doing this adaptation, a detection process monitors the session, determining whether an activity requires to carry out the adaptation. Only if it is an adaptable process, in a non-hierarchical organizational style, an agreement process is executed, and a vote tool is used for reaching a consensus on whether an adaptation process should be performed. When an adaptation or adaptive process should be executed, an adaptation flow process and one reparation —which returns each component to their previous state and notifies users that adaptation cannot take place— will be executed.

The software architecture is mainly focused on the design and implementation of software structures, abstractly defining components that perform a task, their interfaces, and communication between them, in order to meet adequately functional and non-functional requirements of an application. For this reason, this software architecture facilitates the requirements' specification, which will do by a table; which is based principally on the architectural model proposed here, MetaOntology [27], and agile methodologies [28].

Specification Table allows us to: collect all the requirements and agile the design of collaborative applications; reduce the learning curve in the process of creating of this kind of applications, since it is only necessary to complete the table with elements that are intuitive even for any inexperienced person in the domain CSCW; establish how will be access control to application collaborative since these tables are classified by stages, delineating the roles that can participate in each of visualize the collaborative application, since it is possible to define which elements will have the application user interfaces.

The table (see figure 2) contains the elements' specification of the Group Layer (except the organizational structure of the group, policy and user); Interaction Layer, Application Layer. With respect to Adaptation Layer, two columns only are set, one to indicate whether "there or not adaptation” (TA), and another to describe "What is this?” (W?).

4 Case Study

In the Autonomous University of Puebla (BUAP) have sought different ways to improve or increase the quality of student learning, one of these mechanisms is the realization of departmental tests. Which aim to homogenize the teaching of a subject, i.e., that all teachers will cover the same percentage of the academic program. The Faculty of Computer Science carries out departmental tests in different areas of knowledge; however, a departmental test requires a shared workspace to that involved teachers perform group work. For this reason, this paper proposes the development of a collaborative application for managing departmental tests using a software architecture. This application is intended to minimize the time and effort that engaged teachers in the enforcement of departmental tests. Several actors involved in this type of tests are considered: Manager (Mg —he/she is responsible to configure the application, establishing who plays the other four roles, existing areas and what subjects are part of these—); Area Coordinator (AC —he/she registers to EC, and schedules the professors' meetings related with the same subject—); Test Coordinator (TC —he/she organizes the completion of each test, requesting and agreeing the tests number to make, dates and questions of these; then he/she posting the test and the classroom where each Professor will apply it—); Professor (P —he/she proposes and vote date in that the test will be performed, as well as the exercises that it will contain—), and Student (Su — he/she consults date and classroom where the test will make, as well as its scores of each subject—). In general, the five roles must register to join at the session, which is provided by application user interfaces. The collaborative application for managing the departmental tests presents four stages: Application Configuration, Test Preparation, Test Elaborating, and Test Results.

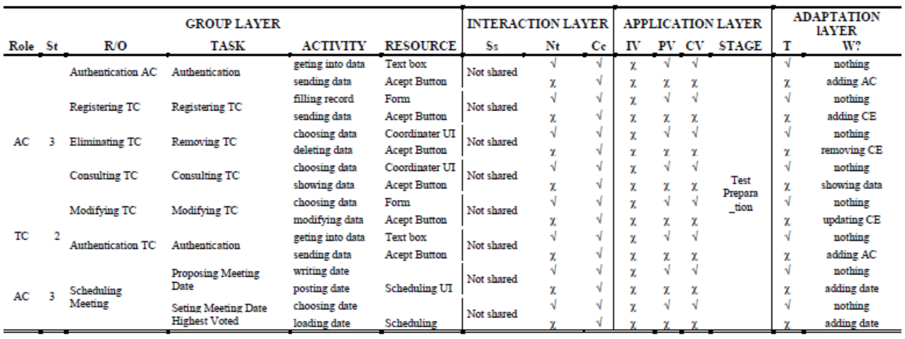
Once it has been explained the case study, then it will prove a software architecture-based methodological approach to develop collaborative applications.

4.1. A Software Architecture-based Methodological Approach

A methodological approach is proposed to simplify and agile the development of a collaborative application. This approach derives of the software architecture mentioned above, and consists in the following steps:

- To elaborate of the requirements specification.
    - Specifying the elements of the Group Layer, for this, the ontological model of the session management policies can be applied. However, for some developers complete an ontology is difficult. For this reason, it is convenient to use a specification table, in which these elements can be laid.
    - Identifying the elements of the Interaction Layer, which must be listed in the specification table.
    - Recognizing the elements of the Application Layer, which must be registered in the specification table.
    - Determining the elements of the Adaptation Layer, which must be enumerated in the specification table.
    - Generate a unique specification table containing the elements corresponding to the four layers of software architecture proposed here.
- To create a sketch of how the application would display.
    - Organizing of the specification table by stages.
    - Determining the user's access control according to the roles that can participate in each stage.
    - Establishing the elements of each user interface, considering the resources and users interact in it.
    - Defining what and how data must be stored.
    - Carrying out a schema of the user interfaces and stored data.
- To implement the collaborative application.
    - Making the schema where will be data stored.
    - Developing the necessary user interfaces.
    - Building each of the web services required to implement the collaborative application.
    - Making the composition of these web services.
- To test of application.
    - Performing the necessary proofs to deliver the required application.

Table 1. Requirements specification of the test preparation stage.



GROUP LAYER						INTERACTION LAYER			APPLICATION LAYER				ADAPTATION IAYER	
Role	St	R/O	TASK	ACTIVITY	RESOURCE	Ss	Nt	Cc	IV	PV	CV	STAGE	T	W?
AC	3	Authentication AC	Authentication	geting into data	Text box	Not shared	√	√	χ	√	√	Test Prepara _tion	√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding AC
		Registering TC	Registering TC	filling record	Form	Not shared	√	√	χ	√	√		√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding CE
		Eliminating TC	Removing TC	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				deleting data	Acept Button		χ	√	χ	χ	χ		χ	removing CE
		Consulting TC	Consulting TC	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				showing data	Acept Button		χ	√	χ	χ	χ		χ	showing data
		Modifying TC	Modifying TC	choosing data	Form	Not shared	√	√	χ	√	√		√	nothing
				modifying data	Acept Button		χ	√	χ	χ	χ		χ	updating CE
TC	2	Authentication TC	Authentication	geting into data	Text box	Not shared	√	√	χ	√	√		√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding AC
AC	3	Scheduling Meeting	Proposing Meeting Date	writing date		Not shared	√	√	χ	√	√		√	nothing
				posting date	Scheduling UI		χ	√	χ	χ	χ		χ	adding date
			Seting Meeting Date Highest Voted	choosing date		Not shared	√	√	χ	√	√		√	nothing
				loading date	Scheduling		χ	√	χ	χ	χ		χ	adding date

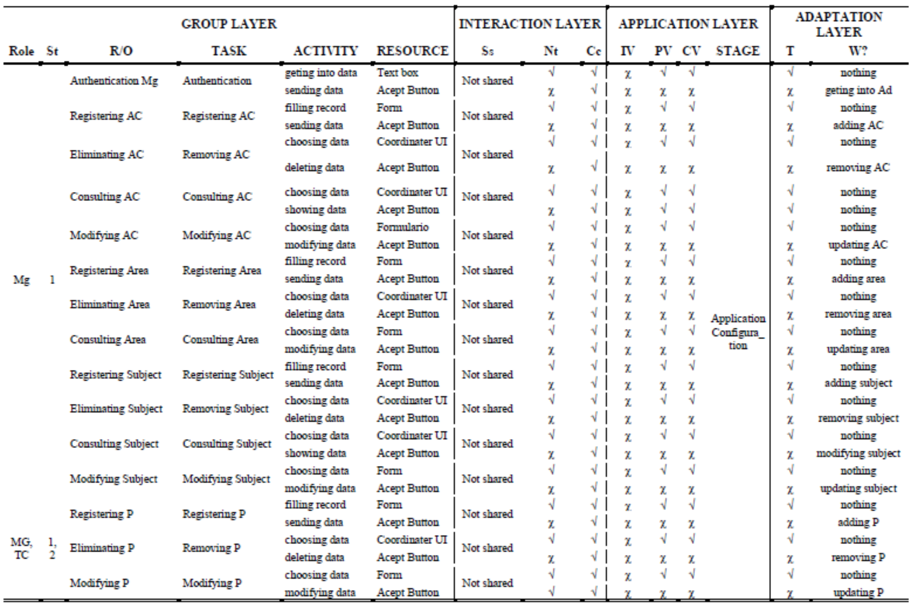
4.1.1. Requirements Specification

This is the first step of the methodological approach, for which will used the specification table. In this paper, the tables of each stage (see table 1 to 4) are shown directly by space issues.

4.1.2. Creation of an Application Sketch

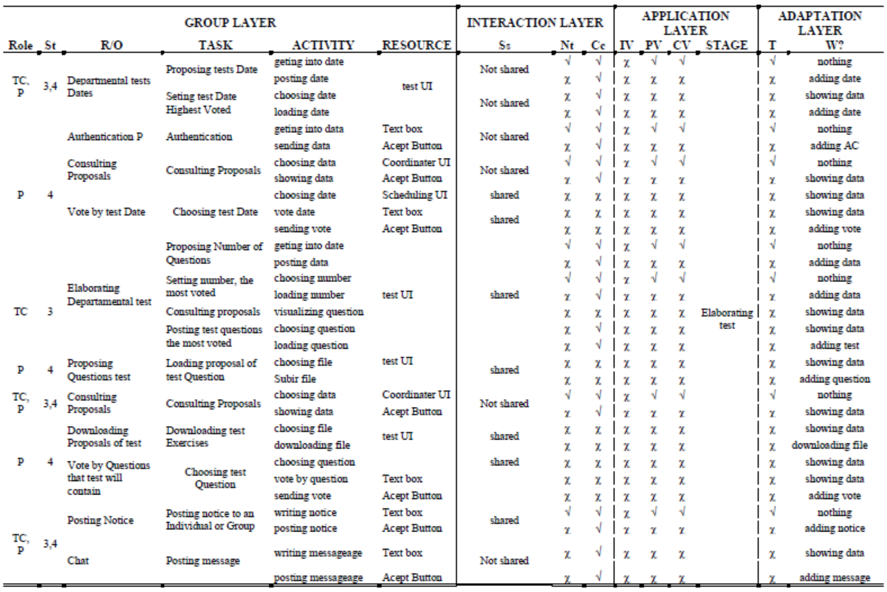
This is the second step of the methodological approach. Only, the tables of each stage (see table 1 to 4) are displayed, it is not possible to exhibit the other elements referent to application sketch by space issues.

Table 2. Requirements specification of the application configuration stage.



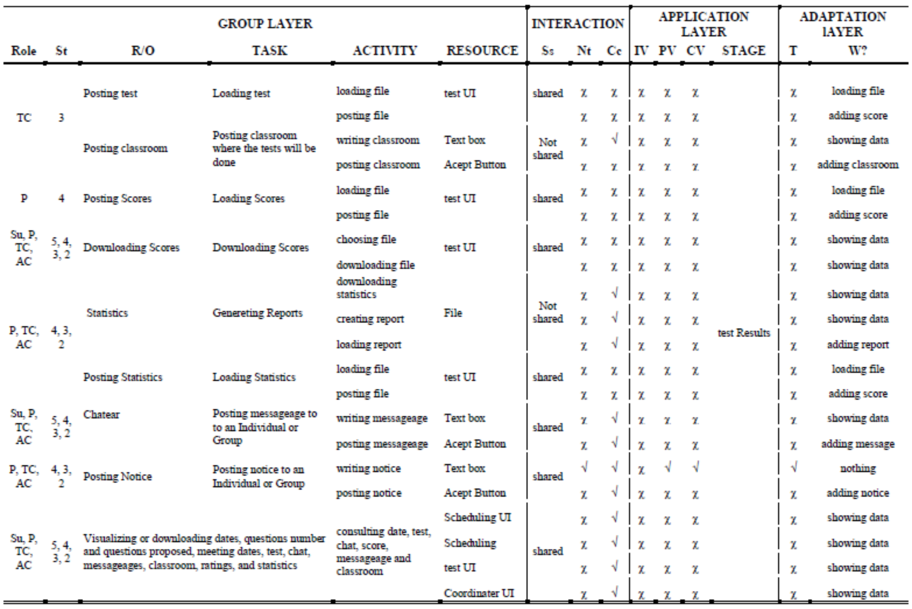
GROUP LAYER						INTERACTION LAYER			APPLICATION LAYER				ADAPTATION LAYER	
Role	St	R/O	TASK	ACTIVITY	RESOURCE	Ss	Nt	Cc	IV	PV	CV	STAGE	T	W?
Mg	1	Authentication Mg	Authentication	geting into data	Text box	Not shared	√	√	χ	√	√	Application Configura_ tion	√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	geting into Ad
		Registering AC	Registering AC	filling record	Form	Not shared	√	√	χ	√	√		√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding AC
		Eliminating AC	Removing AC	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				deleting data	Acept Button		χ	√	χ	χ	χ		χ	removing AC
		Consulting AC	Consulting AC	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				showing data	Acept Button		χ	√	χ	√	√		√	nothing
		Modifying AC	Modifying AC	choosing data	Formulario	Not shared	√	√	χ	√	√		√	nothing
				modifying data	Acept Button		χ	√	χ	χ	χ		χ	updating AC
		Registering Area	Registering Area	filling record	Form	Not shared	√	√	χ	√	√		√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding area
		Eliminating Area	Removing Area	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				deleting data	Acept Button		χ	√	χ	χ	χ		χ	removing area
		Consulting Area	Consulting Area	choosing data	Form	Not shared	√	√	χ	√	√		√	nothing
				modifying data	Acept Button		χ	√	χ	χ	χ		χ	updating area
		Registering Subject	Registering Subject	filling record	Form	Not shared	√	√	χ	√	√		√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding subject
		Eliminating Subject	Removing Subject	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				deleting data	Acept Button		χ	√	χ	χ	χ		χ	removing subject
		Consulting Subject	Consulting Subject	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				showing data	Acept Button		χ	√	χ	χ	χ		χ	modifying subject
		Modifying Subject	Modifying Subject	choosing data	Form	Not shared	√	√	χ	√	√		√	nothing
				modifying data	Acept Button		χ	√	χ	χ	χ		χ	updating subject
MG, TC	1, 2	Registering P	Registering P	filling record	Form	Not shared	√	√	χ	√	√		√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding P
		Eliminating P	Removing P	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				deleting data	Acept Button		χ	√	χ	χ	χ		χ	removing P
		Modifying P	Modifying P	choosing data	Form	Not shared	√	√	χ	√	√		√	nothing
				modifying data	Acept Button		√	√	χ	χ	χ		χ	updating P

Table 3. Requirements specification of the elaborating test stage.



GROUP LAYER						INTERACTION LAYER			APPLICATION LAYER				ADAPTATION LAYER	
Role	St	R/O	TASK	ACTIVITY	RESOURCE	Ss	Nt	Cc	IV	PV	CV	STAGE	T	W?
TC, P	3,4	Departmental tests Dates	Proposing tests Date	geting into date	test UI	Not shared	√	√	χ	√	√	Elaborating test	√	nothing
				posting date			χ	√	χ	χ	χ		χ	adding date
			Seting test Date Highest Voted	choosing date		Not shared	χ	√	χ	χ	χ		χ	showing data
				loading date			χ	√	χ	χ	χ		χ	adding date
P	4	Authentication P	Authentication	geting into data	Text box	Not shared	√	√	χ	√	√		√	nothing
				sending data	Acept Button		χ	√	χ	χ	χ		χ	adding AC
		Consulting Proposals	Consulting Proposals	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				showing data	Acept Button		χ	√	χ	χ	χ		χ	showing data
		Vote by test Date	Choosing test Date	choosing date	Scheduling UI	shared	χ	χ	χ	χ	χ		χ	showing data
				vote date	Text box	shared	χ	χ	χ	χ	χ		χ	showing data
				sending vote	Acept Button		χ	χ	χ	χ	χ		χ	adding vote
TC	3	Elaborating Departamental test	Proposing Number of Questions	geting into date		shared	√	√	χ	√	√		√	nothing
				posting data			χ	√	χ	χ	χ		χ	adding data
			Setting number, the most voted	choosing number			√	√	χ	√	√		√	nothing
				loading number	test UI		χ	√	χ	χ	χ		χ	adding data
			Consulting proposals	visualizing question			χ	χ	χ	χ	χ		χ	showing data
			Posting test questions the most voted	choosing question			χ	√	χ	χ	χ		χ	showing data
				loading question			χ	√	χ	χ	χ		χ	adding test
P	4	Proposing Questions test	Loading proposal of test Question	choosing file	test UI	shared	χ	χ	χ	χ	χ		χ	showing data
				Subir file			χ	χ	χ	χ	χ		χ	adding question
TC, P	3,4	Consulting Proposals	Consulting Proposals	choosing data	Coordinater UI	Not shared	√	√	χ	√	√		√	nothing
				showing data	Acept Button		χ	√	χ	χ	χ		χ	showing data
P	4	Downloading Proposals of test	Downloading test Exercises	choosing file	test UI	shared	χ	χ	χ	χ	χ		χ	showing data
				downloading file			χ	χ	χ	χ	χ		χ	downloading file
		Vote by Questions that test will contain	Choosing test Question	choosing question		shared	χ	χ	χ	χ	χ		χ	showing data
				vote by question	Text box		χ	χ	χ	χ	χ		χ	showing data
				sending vote	Acept Button		χ	χ	χ	χ	χ		χ	adding vote
TC, P	3,4	Posting Notice	Posting notice to an Individual or Group	writing notice	Text box	shared	√	√	χ	√	√		√	nothing
				posting notice	Acept Button		χ	√	χ	χ	χ		χ	adding notice
		Chat	Posting message	writing messageage	Text box	Not shared	χ	√	χ	χ	χ		χ	showing data
				posting messageage	Acept Button		χ	√	χ	χ	χ		χ	adding message

Table 4. Requirements specification of the test results stage.

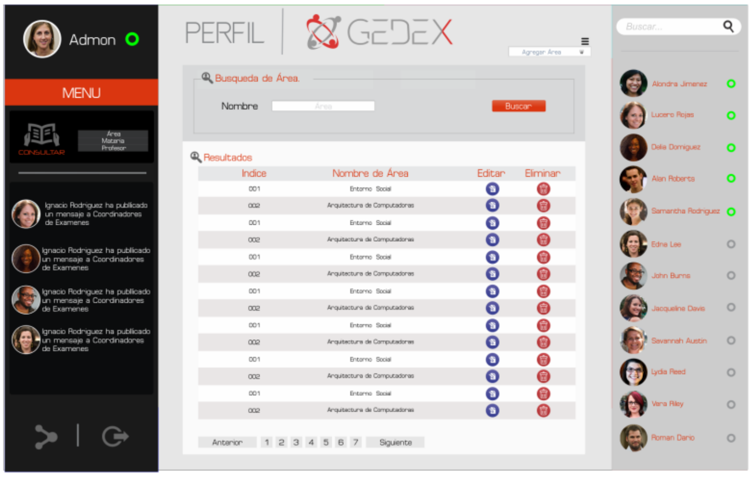


GROUP LAYER						INTERACTION			APPLICATION LAYER				ADAPTATION lAYER	
Role	St	R/O	TASK	ACTIVITY	RESOURCE	Ss	Nt	Cc	IV	PV	CV	STAGE	T	W?
TC	3	Posting test	Loading test	loading file	test UI	shared	χ	χ	χ	χ	χ	test Results	χ	loading file
				posting file			χ	χ	χ	χ	χ		χ	adding score
		Posting classroom	Posting classroom where the tests will be done	writing classroom	Text box	Not shared	χ	√	χ	χ	χ		χ	showing data
				posting classroom	Acept Button		χ	χ	χ	χ	χ		χ	adding classroom
P	4	Posting Scores	Loading Scores	loading file	test UI	shared	χ	χ	χ	χ	χ		χ	loading file
				posting file			χ	χ	χ	χ	χ		χ	adding score
Su, P, TC, AC	5, 4, 3, 2	Downloading Scores	Downloading Scores	choosing file	test UI	shared	χ	χ	χ	χ	χ		χ	showing data
				downloading file			χ	χ	χ	χ	χ		χ	showing data
P, TC, AC	4, 3, 2	Statistics	Genereting Reports	downloading statistics	File	Not shared	χ	√	χ	χ	χ		χ	showing data
				creating report			χ	√	χ	χ	χ		χ	showing data
				loading report			χ	√	χ	χ	χ		χ	adding report
		Posting Statistics	Loading Statistics	loading file	test UI	shared	χ	χ	χ	χ	χ		χ	loading file
				posting file			χ	χ	χ	χ	χ		χ	adding score
Su, P, TC, AC	5, 4, 3, 2	Chatear	Posting messageage to to an Individual or Group	writing messageage	Text box	shared	χ	√	χ	χ	χ		χ	showing data
				posting messageage	Acept Button		χ	√	χ	χ	χ		χ	adding message
P, TC, AC	4, 3, 2	Posting Notice	Posting notice to an Individual or Group	writing notice	Text box	shared	√	√	χ	√	√		√	nothing
				posting notice	Acept Button		χ	√	χ	χ	χ		χ	adding notice
Su, P, TC, AC	5, 4, 3, 2	Visualizing or downloading dates, questions number and questions proposed, meeting dates, test, chat, messageages, classroom, ratings, and statistics		consulting date, test, chat, score, messageage and classroom	Scheduling UI	shared	χ	√	χ	χ	χ		χ	showing data
					Scheduling		χ	√	χ	χ	χ		χ	showing data
					test UI		χ	√	χ	χ	χ		χ	showing data
					Coordinater UI		χ	√	χ	χ	χ		χ	showing data

4.1.3. Implementation of the Collaborative Application

This is the third step of the methodological approach. Only, some user interfaces see Figure 2 to 4) of the application are presented by space issues. As seen in Figures 2 to

4, user interfaces are the result of sketch derivative of the elements placed on the specification table. Although, the application for managing the departmental tests is developing, the proofs already have been performed.



Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

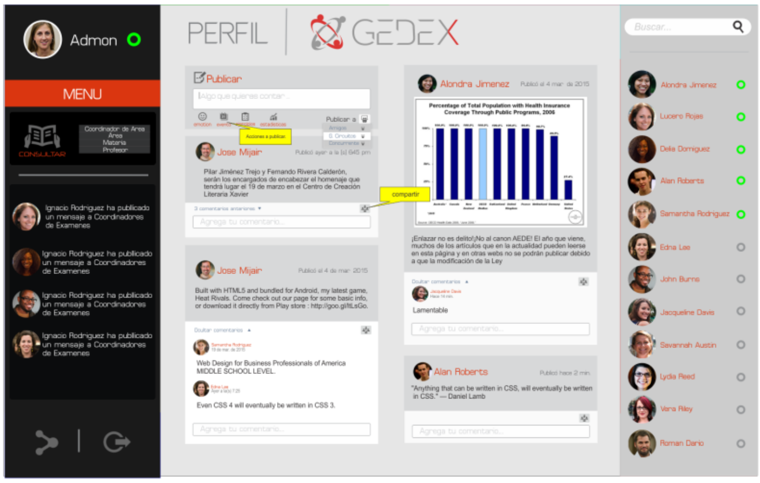
Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Indice	Nombre de Área	Editar	Eliminar
001	Entorno Social		
002	Arquitectura de Computadoras		
001	Entorno Social		
002	Arquitectura de Computadoras		
001	Entorno Social		
002	Arquitectura de Computadoras		
001	Entorno Social		
002	Arquitectura de Computadoras		
001	Entorno Social		
002	Arquitectura de Computadoras		
001	Entorno Social		
002	Arquitectura de Computadoras		
001	Entorno Social		
002	Arquitectura de Computadoras		

Fig. 2. User Interface of the subjects by area.

5 Conclusions and Future Work

This paper has presented a methodological approach based on layered software architecture for developing Collaborative Applications. The approach is enriched with layered software architecture, which offers the sufficient guidelines to build this kind of applications by four layers. These separate this construction on four concerns: group, interaction, application, and adaptation.



Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Jose Mijair — Publicó ayer a la(s) 645 pm

Pilar Jiménez Trejo y Fernando Rivera Calderón, serán los encargados de encabezar el homenaje que tendrá lugar el 19 de marzo en el Centro de Creación Literaria Xavier

3 comentarios anteriores

Agrega tu comentario...

Jose Mijair — Publicó el 4 de mar 2015

Built with HTML5 and bundled for Android, my latest game, Heat Rivals. Come check out our page for some basic info, or download it directly from Play store : http://goo.gl/ltLsGo.

Ocultar comentarios

Samantha Rodriguez — 19 de mar de 2015

Web Design for Business Professionals of America MIDDLE SCHOOL LEVEL.

Edna Lee

Even CSS 4 will eventually be written in CSS 3.

Agrega tu comentario...

Alondra Jimenez — Publicó el 4 mar de 2015

¡Enlazar no es delito!¡No al canon AEDE! El año que viene, muchos de los artículos que en la actualidad pueden leerse en esta página y en otras webs no se podrán publicar debido a que la modificación de la Ley

Ocultar comentarios

Jacqueline Davis — Hace 14 min.

Lamentable

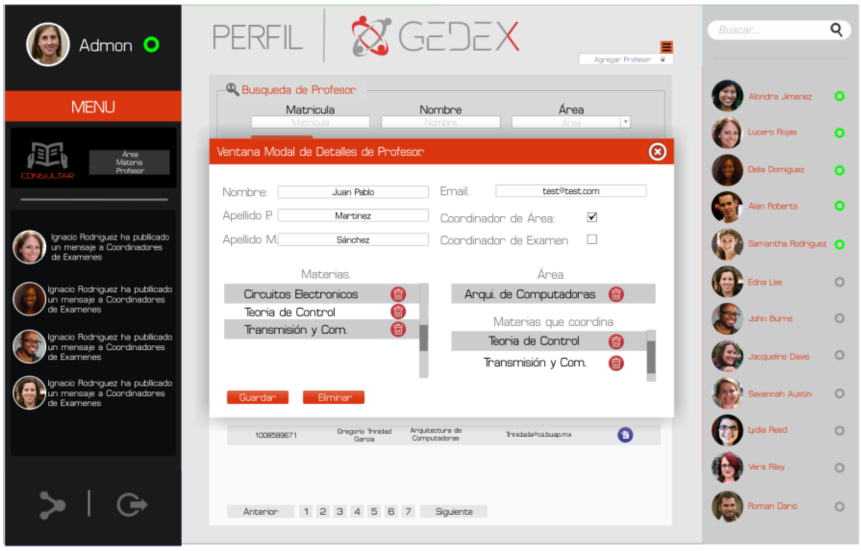
Agrega tu comentario...

Alan Roberts — Publicó hace 2 min.

"Anything that can be written in CSS, will eventually be written in CSS." — Daniel Lamb

Agrega tu comentario...

Fig. 3. User Interface of the professor profile and chat.



Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Ignacio Rodriguez ha publicado un mensaje a Coordinadores de Examenes

Nombre: Juan Pablo
Apellido P Martinez
Apellido M. Sánchez
Email. test@test.com
Coordinador de Área: ☑
Coordinador de Examen □

Materias.
Circuitos Electronicos
Teoria de Control
Transmisión y Com.

Área
Arqui. de Computadoras

Materias que coordina
Teoria de Control
Transmisión y Com.

1008589671 | Gregorio Trinidad Garcia | Arquitectura de Computadoras | Trinidada@cs.buap.mx

Fig. 4. User Interface of the searching of professor.

The methodological approach proposes four phases: requirements specification, sketch creation, implementation, and proof. Which is founded on specification tables that define the elements that will have the application of an intuitive manner, even this can be made by any inexperienced person in this domain, but with application knowledge to carry out. By applying this methodological approach has been implemented the collaborative application of management of departmental tests.

The future work is orientated to establish in a manner detailed this methodological approach.

References

1. Ellis, C.A., Gibbs, S.J., Rein, G.L.: Groupware: some issues and experiences. Communications of the ACM, Vol. 34-1, pp. 39–58 (1991)
2. Kaplan, S.M., Carroll, A.M.: Supporting collaborative processes with conversation builder. Computer Communications, 15(8), pp. 489–501 (1992)
3. Fitzpatrick, G., Kaplan, S.M., Tolone, J.: Work, locales and distributed social worlds. In: Proceedings ECSCW, pp. 1–16 (1995)
4. Fitzpatrick, G., Kaplan, S.M., Mansfield, T.: Physical spaces, virtual places and social worlds: A study of work in the virtual. In: Proceedings CSCW, pp. 334–343 (1996)
5. Beaudouin-Lafon, M.: Beyond the workstation: Mediaspaces and augmented reality. In: Proceedings of the Conference on People and computers IX, 9, pp. 9–18 (1994)
6. Rajan, S., Venkat, R.P., Vin, H.M.: A formal basis for structured multimedia collaborations. In: Proceedings of the 2nd IEEE International Conference on Multimedia Computing and Systems, pp. 194–201 (1995)

7. Venkat R.P., Vin, H.M.: Multimedia conferencing as a universal paradigm for collaboration. In: L. Kjelldahl (ed.), Multimedia: Systems, Interaction and Application, 1st Eurographics Workshop, Springer-Verlag, pp. 173–185 (1991)
8. Edwards, W.K.: Session management for collaborative applications. In: Proceedings CSCW, pp. 323–330 (1994)
9. García, P., Gómez, A.: ANTS framework for cooperative work environments. IEEE Computer Society Press, 36(3), 56–62 (2003)
10. Anzures-García, M., Sánchez-Gálvez, L.A., Hornos, M., Paderewski-Rodríguez, P.: Ontology-Based Modelling of Session Management Policies for Groupware Applications. Lecture Notes on Computer Science, Vol. 4739, pp. 57–64, Springer-Verlag, (2007)
11. Roseman, M., Greenberg, S.: Building Real-time Groupware with GroupKit, a Groupware ToolKit. ACM Trans. Computer-Human Interaction, Vol. 3, 66–106 (1996)
12. Fonseca, B., Carrapatoso, E.: SAGA: A Web Services Architecture for Groupware Applications. In: Proc. of the CRIWG, LNCS 4154, Springer-Verlag, pp. 246–261, (2006)
13. Graham, T.C.N., Urnes, T.: Integrating Support for Temporal Media in to an Architecture for Graphical User Interfaces. In: Proc. of the International Conference on Software Engineering (ICSE'97), ACM Press, Boston, USA, pp. 172–182 (1997)
14. Laurillau, Y., Nigay, L.: Clover Architecture for Groupware. In: Proc. of the ACM Conference on CSCW, New Orleans, Louisiana, USA, pp. 236–245 (2002)
15. Gea, M., Gutierrez, F.L., Garrido, J.L., Canas, J.J.: AMENITIES: Metodología de Modelado de Sistemas Cooperativos. In: COLINE02, Workshop de Investigaci6n sobre nuevos paradigmas de interacción en entornos colaborativos aplicados a la gestión y difusión del Patrimonio cultural, Granada, Spain (2002)
16. Molina, A.I., Redondo, M.A., Ortega, M., Hope, U.: ClAM: A methodology for the development of groupware user interfaces. Journal of Universal Computer Science (2007)
17. Penichet, V.M.R., Lozano, M.D., Gallud. J.A.: An Ontology to Model Collaborative Organizational Structures in CSCW Systems. In: Engineering the User Interface, Springer, pp. 127–139 (2008)
18. Garlan, D., Shaw, M.: An introduction to software architecture. Advances in Software Engineering and Knowledge Engineering, 1, pp. 1–39 (1994)
19. Van Welie, M., van der Veer, G.C., Eliëns, A.: An Ontology for Task World Models, Design, Specification and Verification of Interactive System. Springer Computer Science, 57–70 (1998)
20. Kuutti K.: The concept of activity as a basic unit of analysis for CSCW research. In: Proceedings of the Second European Conference on CSCW (1991)
21. Ellis, C., Wainer, J.A.: Conceptual model of groupware. In: Proceedings of the 1994 ACM Conference on CSCW, pp. 79–88 (1994)
22. Hollan, J., Hutchins, E., Kirsh, D.: Distributed cognition: toward a new foundation for human-computer interaction research. ACM Transactions on Computer-Human Interaction (TOCHI) Special issue on HCI in the new millennium, Vol. 7-2 (2000)

23. Fernández-López, M., Gómez-Pérez, A., Juristo, N.: Methontology: From Ontological Art Towards Ontological Engineering. In: Spring Symposium on Ontological Engineering of AAAI, Stanford University, California, pp. 33–40 (1997)
24. Abrahamsson, P., Salo, O., Ronkainen, J.: Agile software development methods: Review and analysis. VTT Electronics (2002)