CSCW systems in virtual environments: a general development framework

W. J. Sarmiento*‡ and César A. Collazos†

*IDIS Research Group, Universidad del Cauca, wilsonj@unicauca.edu.co
†Full Professor, IDIS Research Group, Universidad del Cauca, ccollazo@unicauca.edu.co
‡Multimedia Research Group, Universidad Militar Nueva Granada, wilson.sarmiento@unimilitar.edu.co

Abstract—Collaborative Virtual Environments (CVEs), defined as systems that allow performing some collaborative tasks in a virtual world, were proposed since 1990. From the beginning, these systems were developed as particular applications of Computer-Supported Cooperative Work (CSCW) and virtual reality concepts about problems with specific restrictions on both collaborative work and technological (hardware and software) requirements. Therefore, design guidelines which could be applied to new developments have not been proposed. This paper presents a general framework for designing and developing CSCW systems in virtual 3D environments, which integrates new methodologies that provide clear and formal techniques to develop CSCW, interactive and 3D systems.

Keywords—Virtual environments; groupware; framework and design methodology.

I. INTRODUCTION

Technological development aims to supply computational tools for supporting tasks developed by humans. Clear examples of this fact are the Computer-Aided Design (CAD) systems, which provide tools to facilitate the work of designer, whilst they perform the creative labor. Similarly, Computer-Supported Cooperative Work (CSCW) systems provide computational tools that looking for supporting and facilitating the work group, i.e. tasks that are carried out cooperatively by various users.

Therefore, CSCW requires more than computational tools, because it requires understanding all aspects of the work group. Consequently, several areas such as psychology, sociology, organizational theory, among others are also involved in the CSCW research [1]. Research on these areas allow identify some key processes in the work group, which must be taking into account for developing CSCW systems: 1) Communication, between team members; 2) coordination, under a set of clear rules, and 3) information sharing in a democratic mode [1], [2]. Other important components in CSCW are the computational tools; they are the channels that allow the users working as a team for achieving a common goal. For doing so, special characteristics should be included in those computational tools, which are commonly called groupware or simply CSCW systems [1]. Therefore, the development of CSCW systems require to take into account the key processes of the work group, therefore, firstable, it is necessary to define particular features of this process into a specific task or work [1].

It is known that observation is one of the most useful tools for characterizing any process done by a user, but, not any observation process could allow obtaining valid conclusions, formal and documented process is also required. In the particular case of working group, observing the group of people working and cooperating in a natural environment allows to identify mechanisms that facilitate to achieve the objectives of the cooperative tasks. To underline a text, to write footnotes, to draft some ideas, are some examples of those mechanisms. In CSCW systems these mechanisms are known as 'sharing artifacts', and could be implemented in two ways either attempting to reproduce a real context or proposing new mechanisms according to the used technology.

The sharing artifacts are used to facilitate the key processes of work group, and also used for achieving the main goal of the CSCW systems, as generating awareness in a cooperative task. The users must have awareness that are part of a team, consequently that their actions affect to others [1]. Also users must have awareness about how technological system mediates between them, by which it is possible to observe latency in the action of the other users, associated to communication channel restrictions [2]. Finally, the users must have awareness that they are executing a task and need information about the task activity [1].

However, technological restrictions are in some cases oppose to the cooperative work of users; for example, in a natural environment, communication process involve natural interaction between team members such as tacit and no verbal messages; coordination process includes monitoring the relevance of the participant feels (enthusiasm, compromise, etc.) and information sharing does not have time or physic space limitations, which must be done in the dinner or in the corridor.

The Collaborative Virtual Environments (CVE) has been proposed as an alternative for improving those aspects in CSCW systems. A CVE is a system that allows carrying out a cooperative task in a 3D virtual world, which provides the possibility of involve several aspects of a natural coll-
laborative scenario in a CSCW system. These systems were proposed since 1990, and its advantages have been exposed and verified by several applications [3]. A simple but revolutionary aspect of the CVE is the user interaction, which is carried out by a graphic embodiment, called avatar. The use of avatars provides alternatives of visual appearance, facial expressions and physical interactions near to reality that allows involving gestural and no tacit communication options [4]. Other relevant aspect is the possibility to offer a space potentially infinite, a virtual world, where the collaborative work can be carried out. This virtual world can be designed for generating empathy with the working team and include several spaces for the task may be done [3].

Likewise, the sharing artifacts can be designed following metaphors near to reality; for example, it is possible see a team member doing a presentation in an environment similar to an office, or see how the avatar of a team member disappears when he/she is disconnected, among others [5]. On the other hand, many other mechanisms have been developed for improving the sensation of realism such as enveloping visualization devices, which increases the immersive sensation improving the mode as users are involved with the task [3]; visual tools for monitoring the work of the group and following sharing tasks, personalized detail views [4] and, more recently, haptic and multimodal interfaces [5].

As it expected, the introduction of those new aspects increases the complexity of those systems, resulting in more and more design requirements that include graphic design, user interfaces, objects and 3D navigation spaces choice, usability and ergonomic conditions, and many other features of the collaborative work. Previous works have proposed software architectures for building of CVE [6], [5], [7], aiming its proposals to software and hardware components and intercommunication between these, leaving the side important issues as design of collaborative work, 3D interaction, sharing artifacts, among others. Other proposals have been centered in 3D interaction models in Virtual Environments (VE) [8], [9]; emphasized in the problems of 3D navigation, selection, manipulation and system control, and in the concepts of 3D awareness. Also, there are works that present the CVE systems, its architecture, advantages and features [10], [11], [12], even some describe the design process used [13], but this experience are not fully replicable, as a guide that can be applied as a formal method. Therefore, it is necessary defining a methodology that guide the process of development of a CVE, from the design, to the evaluation, passing through to the development of software as such. This methodology should be the proposals integration around the design and development so much of system CSCW as applications 3D, basing in CVE concepts, this paper proposes a general framework as an approximation of this methodology.

The framework proposed in this paper allow the design of CSCW systems in virtual environments, which attempt to integrate new methodologies that provide clear and formal techniques to develop CSCW and interactive and 3D systems. The next section includes a brief review of related works. In the section 3 details of the proposed framework are described, the next section shows framework implementation details through an example, and the last section presents general conclusions and some reflections about future work.

II. RELATED WORKS

This section shows the background about design and development of CVE, the first part presents a description of more representative CVE, and the next explains other proposals in order to design these systems.

A. CVE developed examples

Since the CVE were proposed, some examples have showed their design and development. Classic examples as MASSIVE [11], Spline [12] y DIVE [14], were the fists that include the main concepts as multiuser support, avatars, interactions with world objects, etc. But the more important contribution of these classic systems was the basics concepts of 3D awareness: aura, focus and nimbus. Aura is defined as a volume around an object that border its presence in virtual world and allows the interaction with others [10], [14], focus is a region were an observer can put its attention, and nimbus is a region surrounding an object that limit its observability [11], [15]. Thus, interaction between two objects could be possible if the auras are overlapping, and so an observer can put his/her attention in an object. Previous examples showed the advantages and the potential of the CVE, but do not allow to transfer the experience in other contexts because no describe the process used for their development; furthermore, these systems have been designed for general purposes with technological exploration as main target.

Some more recent examples have extended the functionality of systems developed previously; as sTeam3D, which is a a web X3D client that allows transparent connection to a sTeam server. This system takes advantage of sTeam, an open source groupware that provides some mechanisms to communication and cooperation, so much synchronous as asynchronous mode, and combines these mechanisms with hypermedia document management [16]. A similar example is SLMMeeting, which implements some general purpose collaborative activities in Second Life platform, as agenda, meeting chat board, booking list, among others. This system is supported chiefly by ability already included in Second Life for creation, administration, navigation and interaction with virtual worlds [17]. However, these systems not present a formal guide for integrating CSCW systems with VE applications.

MaDViWorld is other representative example; this system is a framework for the development of VE with some collaborative features [18]. This work allows manage multiple
users session in a virtual room through avatars, where the users can be interacting with objects inside room, even allows multiple virtual room simultaneously. However, this model is only a viewpoint of the user sessions management, proposing the use of “virtual world paradigm” against position to the traditional “document paradigm”. MaDViWorld does not offer any support to management or design of the VE, avatars and 3D interaction, since the authors indicate literally “These aspects, however, are not our first preoccupation and are rather considered as improvements that could be added later within the system implementation layer of the framework” [18].

B. Other methodological approaches

Benford et al. presented the design of a CVE; they used the ethnography as methodology for the formal observation process of users. The report published describes the results of ethnographic study carried to the interior of TRC Company, where they identified that the documents generated by the company were the core of the collaborative processes. Thus, documents were modeled as “shared objects” over which they designed the shared artifacts. The authors show the importance of the design of the collaborative processes as central point of CVE development [13]. Although the report presents some recommendations of design, this not is a methodology of design and development.

Some work proposes general architectures in order to develop CVE. Dai et al suggest a model that integrates the conference function of NetMeeting with VE capabilities using VRML [6], the proposal describe details of the software components, which are using COM as interface for the system integration. the proposal describe details of the software components, which are using COM as interface for the system integration. Gomes et al. propose an integration framework for extended CVE with third party tool, the proposal core is “integration module” between CoLab, Platine and VNet, where CoLab allows collaborative capabilities, Platine provides communication tools (chat, audio and videoconference) and VNet is a free multi-user VRML system [5]. These proposals show systems architecture but do not include guidelines to design or to develop CVE.

Other proposals aim to design 3D interaction in CVE, for example, Otmane et al. present a formal method to specify and mathematical modeling three 3D interaction functions: navigation, selection and manipulation, and their relation with the specific concepts of 3D awareness (aura, nimbus and focus) [8]. Hrimech and Merienne suggest a method for evaluating the impact of 3D interaction in CVE, in order to develop a guide of design. They proposed three factors for a system evaluation: co-presence, involvement/awareness and collaborative effort apply to the 3D interaction functions. The factors are evaluated with a set of metrics acquired by user observation, monitoring system and applied questionnaire [9]. Nonetheless, there are methodologies, successful tested, directed to design and development CSCW systems; as SHAstra [19], cooperation scenarios [20], designing with ethnography [21], AMENITES [22], ontology-driven analysis [23], among others. These methodologies, each one following itself approach, show a formal process in order to building groupware, with common emphasis: collaboration work as the main goal to CSCW; but these methodologies have not been proposed to develop CVE systems.

In other hand, the development of some 3D software need to consider algorithms and techniques of real time render, 3D interaction, and specific data representation, in the software engineering process [24]. Also, in some particular case, i.e. game 3D development, activities as conceptual design, write history, modeling, animation of characters and world, must be integrating in the formal development process [25]. Some methods de facto have been created in the game development industry; there are modifications of agile methodologies where conceptual and artistic activities have been included in design process [24], [25]. In order to integrate these activities in the software engineering process, some approach proposes to use the Unified Modeling Language (UML) for documenting 3D modeling and animation activities [26]; and other proposals use no rigid documents as character sheets, design document, concept document, project plan, etc. [24], [25].

III. PROPOSED FRAMEWORK

Nonetheless, there are methodologies, successful tested, directed to design and development CSCW systems; as SHAstra [19], cooperation scenarios [20], designing with ethnography [21], AMENITES [22], ontology-driven analysis [23], among others. These methodologies, each one following itself approach, show a formal process in order to building groupware, with common emphasis: collaboration work as the main goal to CSCW; but these methodologies have not been proposed to develop CVE systems.

In other hand, the development of some 3D software need to consider algorithms and techniques of real time render, 3D interaction, and specific data representation, in the software engineering process [24]. Also, in some particular case, i.e. game 3D development, activities as conceptual design, write history, modeling, animation of characters and world, must be integrating in the formal development process [25]. Some methods de facto have been created in the game development industry; there are modifications of agile methodologies where conceptual and artistic activities have been included in design process [24], [25]. In order to integrate these activities in the software engineering process, some approach proposes to use the Unified Modeling Language (UML) for documenting 3D modeling and animation activities [26]; and other proposals use no rigid documents as character sheets, design document, concept document, project plan, etc. [24], [25].

Taking into account the specific characteristics of CSCW systems in VE explained above, our framework proposes to involve aspects related to the design of both 3D software
and CSCW systems in a common approach. Therefore, our framework preserves the main goal of CSCW, named collaborative work, and the main features of 3D software i.e. the well-known interaction possibilities. The figure 1 illustrates a sketch of the proposed framework, which is a cycle composed of seven stages: requirements engineering, 3D design, collaborative model, interaction and visualization model, software development, collaboration monitoring and awareness monitoring, which should be performed iteratively. Note that, the proposed framework starts and finalizes with the “User observation” process, which must be the core of any development process in systems that involve user interaction.

A. Software requirements (Step ①)

As it is usual, requirements engineering provides methods and techniques that allows a formal and standardized elicitation, specification and analysis of needs of users [22], [23]. Herein, we emphasize that this process must be based on the user observation; particularly CSCW systems require observation of a team collaborating for developing a specific task in the “real conditions”.

B. 3D design (Step ②)

The main advantage of work in a virtual world is the amount of communication and interaction possibilities. However, these possibilities only can be provided if VE designs generate empathy with the users. For example, development of video games includes a conceptual design (scenarios, characters and history) that allows reaching a hard sense of immersion in the gamers [25]. In order to apply this concept, the proposed framework defines a specific step for obtaining a high level, or conceptual, 3D design. This conceptual design must to include scenarios, avatars, shared artifacts and interaction modes and to be results of a formal user observation process. Furthermore, these designs must be integrated with the software development process by using documentation and formal languages such as UML [26].

C. Collaboration Model (Step ③)

When user requirements and 3D interaction modes have been specified, the collaboration model defines a formal schema that abstracts the key processes of work group (communication, coordination and sharing) in the domain problem. Thus, this step delivers a dynamic collaboration model that includes workflow, user roles, assigning-splitting task and negotiation processes. Additionally, this model also must to define low level processes such as session users, information persistence, concurrence and synchronism, among others. Taking into account that the collaboration model is the main aspect of most of the CSCW development methodologies, this stage could include approaches proposed by any of those methodologies [19], [22], [21].

D. Visualization and Interaction Model (Step ④)

3D visualization and interaction model should be designed based on the dynamic collaboration model defined in the previous stage. Whilst the 3D design was focused on the conceptual design of the CVE, the visualization and interaction model aims to specify how the users will interact in that VE. For doing so, it is essential to define the mechanisms for generating 3D awareness such as aurea, nimbus and focus, which define the characteristics of relevant VE elements that should be designed [3]. First, it is necessary to define the navigation and object selection schemes, for instance avatar movement options (modes and roads), task visualization levels, and view customization alternatives, among others. The tacit and explicit avatar communication modes, the specific sharing artifacts and how they are used must be also specified and designed in this stage.

This stage includes also activities related to 3D modeling and animation of VE elements, which must be coherent with the conceptual 3D design defined in the step ②. Additionally, hardware devices that allow that users visualize and interact with the 3D virtual world must be choices. Although, device selection attempt to achieve the interaction requirements, it is possible that new software requirements should be specified for supporting that in the system.

E. Software development (Step ⑤)

Once the particular application design models (3D, collaboration, visualization and interaction models) have been elaborated, computational applications should be implemented. For doing so, several methodologies have been proposed, which generally compress both architectural design and codification. In the architectural design, low level representation of VE elements (3D objects, persistent data, collaborative tasks, etc.) and their dynamic behavior should be modeled [24]. Then, this architecture is used a base for building a software application that should to achieve all requirements defined in previous stages. Herein, it is important to emphasize that the whole framework is proposed as an iterative process. So, software development will also be in an iterative process, when each iteration allows fulfilling a set of requirements, resulting in several software prototypes.

F. Collaboration Monitoring (Step ⑥)

One of the most important steps in software development is the test process, which allows introducing user feedback for improving details in the next iteration. As was described in the previous sections, requirements of CSCW systems there are not only related to functional and interface capabilities but is also relevant to assess the accomplishment of its main goal i.e. to facilitate the development of collaborative tasks. For achieving that, the framework specifics an activity named collaboration monitoring, which should be carried out mean of a formal and documented user observation process when they are interacting with the system. Measurement
of collaboration degree is yet an open research problem. Some aspects that should be evaluated in the CSCW systems are the cohesion between team members, the kind and quantity of messages and information shared between them, the efficiency and effectiveness task developments, among others [19].

G. Awareness Monitoring (Step ③)

Another fundamental aspect that must be evaluated is the degree of user awareness. This monitoring process finds to measuring the relevance of the interaction and visualization schemes provided by the VE for generating a high level of user awareness in different levels such as self awareness in a collaborative team, awareness of the other and the work group. In this stage some questions should be asked, for instance: how the interaction and visualization devices are relevant to these awareness senses? what is the comfort of users with the avatars and world aspects?, what is the degree of user fatigue produced by the interaction and visualization devices?, what is the degree of ergonomics and usability of the interaction models?, etc. It is to say; in this stage the quality of experience in the collaborative work should be determined [8], [9]. Newly, development of objective measures that allow evaluating the user awareness is also an open research problem.

IV. IMPLEMENTATION DETAILS

This section illustrates the main implementation details of the proposed framework using a simple example. We assume that it is necessary to develop a CVE for promoting the working group in a company in order to improve the labor climate. For doing so, the CVE must to achieve the following characteristics 1.

It is necessary to design a contest in a virtual world. Several teams will carry out the game, each of them composed by four gamers. The main goal will be to find ten keys that will be hidden in ten virtual rooms, which will be unconnected. Each virtual room will has several boxes, one of them containing the searched key. Ten clues, one by virtual room, will be provided in order to facilitate the searching task. The team that finds the ten keys in the smaller time will win the game. Game start by placing three team members in three different virtual rooms, the fourth member will act as an observer that will be able to interact with the others using a communication artifact such as a chat room. Gamers will have two minutes for exploring the room, after that time they will be translated to other rooms and roles will be changed i.e. other gamer will be selected as observer. Therefore, the observer role will be essential for supporting the exploration task in rooms observed or visited previously by him.

In order to improve the system acceptance by the users, it is necessary to guarantee a visual coherence that allows generating a good empathy user-system. For doing so, the proposed framework suggest to develop the 3D design stage (step ②) which involve the conceptual design of 3D components that will define the avatar visual appearance. Figure 3 shows an initial conceptual design in character sheets. In this case a simple but friendly appearance is defined. It is expected that proposed design generate acceptance between the users. An important aspect is the coherence that maintains the avatars and virtual world appearances. Additionally, note that it is required to consider several alternatives, which should be approved by the stakeholders. For example in the figure the visual effect of changing a simple feature, as is the color environment is also illustrated. For our example the visual concept (b) was selected, this seems to be most friendly, less traditional and break out traditional working schemes. Furthermore, these designs must be integrated with the software development process by using documentation and formal graphical languages such as UML.

After that visual appearance is defined, the collaborative design process should be performed. As was mentioned above, it is the base for developing interaction and communication processes. Therefore, at the stage ③ (collaborative-
tion model) we propose to appropriate previous concepts and methodologies that have shown be useful in CSCW system designs. In particular, in this example we use the COoperative MOdel notation based on UML, COMO-UM, which are a modification of the well-known state diagrams of UML 1.5. Details of that notation can be found in [23]. shows the collaboration models designed for our system, in (a) the organization diagram and in (b) the collaborative task “Key Finding”. In these diagrams the different user roles and the interactions between them can be observed, in (a) it is possible see roles transition by time out event, and (b) shows the role actions in the collaborative task.

The stage refers the development of the visualization and the interaction models. These results from the application of the collaboration model to the 3D environment based on the visual parameters arose in the conceptual model. This stage also involves the functional and visual GUI design. Figure 5 shows the design of a virtual room in the 3D world, an avatar model. Note that these models are coherent to the conceptual design selected in step. The GUI design must be include two different schemes, the first one to the active gamer and the second one for the passive gamer (the observer). The awareness elements involve the animation of the iconic avatar to the passive gamer, which will be used for indicating that this is writing a message, audio alerts for indicating that a message has been sent, visual alerts for indicating to active gamer that the passive gamer requires your attention, among others. In this example some concepts such as aura, focus and nimbus were not used because the users not share a common space.

Figure 6 illustrate the interaction model with interactions diagrams for each role, in this example we use the CurTaskTrees (CTT) notation [28], that is commonly used in user interface specification. This model involves a chat system but also non-verbal schemes such as attention calling and other awareness elements. As it is expected that this CVE be executed in corporative environments, we use a navigation pattern that is well known and used in several video games, described in the IV. This is characterized because only uses mouse and keyboard as input devices.

Once design stages have been performed follows the software development stage (Ⅲ). This stage starts with the design of a low level architecture that it is then taken to a programming language for generating a software appli-
In this stage is also important to define a logic representation model of the 3D elements. In this case, we use a graph representation of the scene, which make easier the navigation process, the animation of the avatars and the detection of possible collisions.

The proposed framework defines two last stages dedicated to evaluating the system. The first one named collaboration monitoring (step 6) is focused in to evaluate how the team performs the communication, coordination and sharing information processes. In this case, some metrics such as number of sent messages, mean answer time, number of attention calls, mean time for founding each key, effectiveness of the collaboration process developed by the passive gamer, which could be measured doing an analysis of the system log. The second one 7 should evaluate the user immersion based on the monitoring of the awareness elements.

Finally, the cyclic model of the proposed framework suggest that several adjusts could be applied on a specific system element such as the visual characteristics, awareness elements, collaboration model, etc., but also could be applied to the global system according to an analysis process.

V. CONCLUSIONS AND FUTURE WORK
In this paper a general framework for developing CSCW system in VE have been proposed. The main aspect of this framework is the fact that requirements specification and evaluation processes are centered in observation of user activities when developing collaborative tasks.

The proposed framework is defined as an iterative process composed by 7 steps: CSCW and 3D software design models. The requirements specification has been divided in two independent processes: software requirements and 3D design. On the other hand, the system modeling is composed of three steps called collaboration model, visualization and interaction model and software development. These three models are integrated for generating a final computational application that ideally achieves all requirements of the system. Finally, evaluation process has been divided in two monitoring processes that attempt to assess the main aspects of this kind of systems i.e. collaboration and awareness capabilities. Due that this framework has been proposed as an iterative process, problems detected in the first iteration will be fixed in the next iteration process.

Future work will be focused in the definition of detailed artifacts for developing each step of the proposed framework. In that way, we will be interested in the definition of an implementable instance of the proposed framework and the evaluation in some specific collaborative applications.

ACKNOWLEDGMENT
Wilson Sarmiento is fully supported by the Universidad Militar Nueva Granada at Colombia.

REFERENCES


