# Tracking The Evolution of Collaborative Virtual Environments

*Rubén Mondéjar-Andreu, Pedro García-López, Carles Pairot-Gavaldà, and Antonio F. Gómez-Skarmeta*

*Collaborative virtual environments have been traditionally influenced by virtual worlds and collaborative distributed environments. They provide more realistic collaborative and user-oriented tools thanks to these systems' immersion and interaction skills. Construction of such environments on top of different middleware or framework solutions is of particular interest, since the environment can benefit from the abstractions provided by the underlying layers, as well as their reusability and extensibility features. This approach can help these systems to easily adapt to new trends and novel technological challenges. In this article, we explore the foundations of collaborative virtual environments, focusing on innovative building blocks like decentralized distributed infrastructures. Finally, we analyze different key application fields and draw some conclusions.*

**Keywords:** Distributed Systems, Frameworks, Grid, Groupware, P2P, Virtual Environments.

## 1 Introduction

Groupware or **collaborative environments** are defined as any environment used by some number of concurrent users located elsewhere, that perform tasks in cooperation to achieve common goals. Moreover, it provides a way of interaction between these users and the shared scenario. The CSCW (Computer Supported Cooperative Work) term was coined to reference such software, although there is some blurring between the different meaning of the terms collaborative and cooperative. Though both these terms are used as synonyms, there are some subtle distinctions. Cooperation and collaboration do not differ in terms of whether or not the task is distributed, but by virtue of the way in which the task is divided. In cooperation the task is split (hierarchically) into independent subtasks. In collaboration, processes may be (hierarchically) divided into intertwined layers. In cooperation, coordination is only required when assembling partial results, while collaboration is a coordinated, synchronous activity that is the result of a continued attempt to construct and maintain a shared conception of a problem.

There exist a vast number of different collaborative environments, each of them bringing its own vision, with a set of **collaborative tools**. Such tools are classified into three different types according to the time frame and user's presence: those that involve asynchronous communication (e-mail or discussion forums), those that involve synchronous communication, meaning that users must be connected to collaborate (chat or instant messaging), and finally those that support both synchronous and asynchronous group activity, like electronic calendars, where events and important dates are signalled.

In the last years we have experienced a huge leap in communication technologies. The augment of network bandwidth has promoted world wide resource sharing and thus enabling novel killer applications which have boosted Internet's use. BSCW [1] and Groove [2] are two significant examples of collaboration-oriented tools, which follow the workspace concept. BSCW is a classic in the collaborative environments software, and as with many others, it is based on client-server web technology. It allows workspace sharing between different users. On the other hand, Groove is a shared desktop application including file

**Authors**

**Rubén Mondéjar-Andreu** is a PhD student in the Dept. of Computer Science and Mathematics at *Universitat Rovira i Virgili*, Spain. His research interests include distributed systems and collaborative middleware over structure peer-to-peer networks. He has BSc and MSc degrees in Computer Engineering from *Universitat Rovira i Virgili*. <ruben. mondejar@urv.net>

**Pedro García-López** is an Assistant Pprofessor in the Dept. of Computer Science and Mathematics at *Universitat Rovira i Virgili*, Spain. His research interests include computer-supported cooperative work and distributed systems. He has a PhD in Computer Engineering from *Universidad de Murcia*, Spain. He is a member of the IEEE. <pedro.garcia@urv.net>

**Carles Pairot-Gavaldà** is a PhD student in the Dept. of Computer Science and Mathematics at *Universitat Rovira i Virgili*, Spain. His research interests include distributed systems and middleware infrastructure for peer-to-peer networks. He has BSc and MSc degrees in Computer Engineering from *Universitat Rovira i Virgili*. <carles. pairot@urv.net>

**Antonio F. Gómez-Skarmeta** is an Assistant Professor in the Dept. of Information and Communications Engineering of the *Universidad de Murcia*, Spain. He received an MSc in Computer Science from the *Universidad de Granada*, Spain, and a PhD in Computer Science from the *Universidad de Murcia*. His research interests include distributed systems and security and mobility in computer networks. <skarmeta@dif.um.es>

sharing services, messages, chat, synchronized web browsing, or calendar. It is based on a peer-to-peer (P2P) architecture, meaning that it does not rely on a single server.

**Three-dimensional (3D) virtual environments** were conceived a long time ago and have also evolved as different applications, like simulators, 3D chats or videogames. 3D Virtual worlds' origins are found in military simulation, more precisely in flight simulators. Consequently, public commercialization of such technology was open to all audiences, and originated the model which is still valid nowadays: 3D graphics in immersive environments trying to offer more degrees of freedom within the virtual scenario

Besides networking advances, existing systems are gaining in computing power as well as in graphics technology. Consequently, **Collaborative Virtual Environments** (CVEs) are the next step in collaborative applications, since their aim is to change the way in which we perform our daily work, study, play, communicate, or, in general, interact with others.

## 2 Background

Let us discuss what we consider the basic foundations of collaborative virtual environments. Firstly, **3D virtual worlds** provide the three-dimensional view, and secondly, **distributed systems** are necessary to offer multi-user and collaborative tools capabilities.

### 2.1 Three-Dimensional Virtual Worlds

A 3D virtual world can be seen as an environment where not only the appearance but its components' inner representation belongs to a **three-dimensional** space. A **virtual world** provides a way for the users to interact. For example, modifying its components, interacting with them, etc.

Additionally, these modifications must be autonomous, allowing the user itself to decide what kind of **interaction** to choose. Therefore, it is an **immersive** environment, meaning that user is capable of recognizing that he is inside a 3D environment, assimilating all temporal and spatial aspects related to it. Such environments need to work in **real time**, requiring good response times, so that users do not notice lags between actions and the immediate effect which affects their context. Lastly, it must be a realistic environment, where appearance quality must reflect as much as possible the real world.

### 2.2 Distributed Systems

Distributed computing studies the coordinated use of physically separated computers. As stated by Andrew S. Tanenbaum, "*Distributed systems need radically different software than centralized systems do*." Independent of its architecture or the coupling degree between the different network members, we want to highlight the event propagation service and persistence service, which we consider as key services for application development on top of these environments. They provide necessary abstractions to disseminate and store the system's state in multi-user collaborative applications.

Both of the dominant distributed systems' architectural approaches are discussed as follows:

a) Client-Server Model

This is the classic and most used model, because it is the easiest to implement. There are just two types of different entities: the server, which offers all services, and the client, which uses them. In this model, all services, like state propagation or data persistence, are offered by the server itself, being responsible to propagate and store information.

When a server is unable to withstand system's load, more servers can be added, adopting *clustering* strategies or server federation techniques, mainly depending on the local or remote machines location. In either case, adoption of these strategies implies an economic overhead, charged to the institution which hosts the servers.

b) Decentralized Models

The alternative to the previous model is decentralization. The idea is that all components in the distributed system have the same responsibilities acting both as clients and servers. These systems also satisfy the following features:

- **Load balancing**. It is not necessary to calculate *a priori* the systems' capacity depending on working peaks. Capacity can be reassigned on demand.
- **High availability**. They are fault tolerant because if a node fails, services are immediately rescheduled to another network node.
- **Low cost**. The services offered by these architectures are managed in a way that obviates the need for powerful servers.
- The most wide-known decentralized models available nowadays are the **Grid** (Global Resource Information Database) and the **peer-to-peer model**.

The **Grid** evolves the cluster and server federation concepts. In this model, homogeneous nodes are grouped to solve complex problems that require more resources or scalability to a high number of users. This architecture allows coordination and resource sharing between remote users. The Globus [4] project is an implementation of the Grid model.

The other approach is the **peer-to-peer model**, with a greater degree of decentralization than the Grid. It can be defined as an architecture for distributed systems which typically has lots of heterogeneous nodes which cannot be trusted. It is fault tolerant and self-organizing; operating in dynamic scenarios with frequent node joins and leaves. Its typical objective is to benefit from shared resources (like CPU, network bandwidth or storage) all along the different network nodes which form the overlay network.

## 3 Collaborative Virtual Environments

As we explained in the previous section, the virtual worlds evolved to multi-user virtual environments using distributed systems technology. Moreover, we can consider Distributed Virtual Environments as the origin of Collaborative Virtual Environments. To better understand such environments, we present a group of key requirements that are necessary to build such multi-user applications.

### 3.1 Requirements

***a) Session Administration and Zone***

To determine the structure of the environment we can use the **zone** concept. A zone is a part of the virtual environment where users can interact and use the different available objects. However, when there are many state changes in a short period (movements, actions, arrivals and departures), the system reach a state where it cannot meet the requirements in real time.

Here is where it becomes necessary to use mechanisms such as **zone partitioning,** reducing the virtual space and the number of users, so decrementing the maximum number of simultaneous events needing to be processed. A very simple algorithm consists on discarding the events at a distance threshold from us.

***b) Shared Objects and State Propagation***

Another essential concept is the explicit support of shared objects. Usually, users engage in collaborative interactions with other users in the shared context, or with accessible collaborative components. These collaborative components, also called artefacts, range from synchronous to asynchronous, depending on the interaction time frame. There are many types of artefacts e.g. voting tools, slide presenter tools, documents, audio/video players, etc.

To ensure the state propagation required to guarantee the persistence in the overall system, each change should be spread and later on persisted to be properly restored in another moment. This is one of the biggest problems in distributed systems because these systems must maintain the state of the components in the virtual environment in a seamless and coordinated way.

***c) Coordination and Consistency***

Another feature that should be present in any collaborative environment is the explicit support for coordination policies. These policies can be categorized in roles for access control and concurrency control. The platform should provide extension hooks to insert custom coordination components.

The concurrency control mechanism should provide mutual exclusion services for shared artefacts, so that the users do not produce inconsistencies in the use of these tools. An example to illustrate this problem could be the transport of an artefact from one zone to another of the virtual world.

***d) Awareness***

Awareness is another important feature in the virtual environments. We can define it as an understanding of the activities of others, which provides a context of your own activity. An awareness platform should provide data acquisition from the running environment. The importance of a well-defined awareness model must be emphasized.

### 3.2 Distributed Virtual Environments

Distributed Virtual Environments are the bridge between virtual worlds and collaborative virtual environments. One of the firsts 3D distributed environments was DIVE (Distributed Virtual Interactive Environment) [5], created in Sweden by SICS (Swedish Institute of Computer Science) in 1991. It provided representation by avatars, different browsing and interaction types, and audio and video communication.

The DIVE environment consisted of a group of distributed processes that represented avatars or objects, which could modify the database that contained the representation of the world. The communication among these processes was based on multicast protocols and thus did not rely on a central server architecture. In fact, each user possessed a copy of the world in which they collaborated.

We can also find other simpler examples, such as 3D chat communities like ActiveWorlds [6]. It belongs to a company located in Massachusetts, USA, that provides a platform and the necessary software to interact with other users in a 3D environment. ActiveWorlds also gives the possibility to rent a space in its server so that the user manages his own world.

### 3.3 Collaborative 3D Frameworks

After DIVE, other environments like SPLINE, NPSNET or MASSIVE appeared. Many of them have successfully solved problems like the structuring of the world and the zone partitioning, the state propagation, the coordination and the consistency, and the awareness. They also introduced concepts like the portals between worlds, to be able to move between them.

Those collaborative environments offered richer user interactions by means of flexible collaboration tools. Nevertheless, the complexity of the domain has led to huge and intricate software systems that are difficult to extend and reuse, so precluding system interoperability and flexibility. To cope with these problems, the trend is to augment modularization in the design of such infrastructures. Many systems are following a framework-based approach to augment modularization and provide extension points to develop new applications on top of them.

In the domain that we are studying, we highlight two examples of framework-oriented platforms enabling the development of collaborative virtual applications. One example is Ants CSCW [7], a generic multi-user collaborative framework. The ANTS system is a component-based application framework that simplifies development of distributed collaborative components in the Java language. It provides a client side container for JavaBean components that hides complexity from middleware layers by providing a remote persistent property mechanism and a distributed event service. The key architectural decision of creating the distributed event service on top of a publish/subscribe notification system permits a very powerful and flexible approach. On the one hand, state propagation is efficiently accomplished by the decoupled notification system.


**Figure 1:** Students Watching a Video in MOVE.

On the other hand, a seamless awareness model is achieved that directly benefits from filtering capabilities offered by the messaging middleware.

MOVE (see Figure 1) is a Collaborative Virtual Environment constructed on top of the ANTS component groupware framework. The system fully benefits from the underlying collaborative services provided by the ANTS infrastructure. MOVE provides essential collaborative services like shared sessions, support for synchronous and asynchronous components, security, coordination and a server-side awareness infrastructure. One of the key aspects of this work is to provide extensibility at all levels of the system. It is thus possible to create new components or artefacts, to develop new coordination mechanisms, to interface with different middleware services and propagation channels, and to develop new actuators for the awareness service. This creates an interesting playfield for research in Virtual Environments by opening the system to third-party extensions willing to solve specific problems in this wide domain.

Another framework, Croquet [9], provides a virtual environment enabling third-party development of collaborative tools. Croquet supports collaborative access to shared resources in a wide-area scale (see Figure 2). Croquet aims to provide an extensible substrate for the construction of collaborative virtual environments. The users and the user groups can build private or shared worlds and to make them accessible by means of *interdimensional* portals. Croquet is based on a peer-to-peer system and uses replication mechanisms to guarantee the maintenance of the virtual world state. Also, if a machine client is too slow to compute the state, it is discarded for this task.

Both MOVE and Croquet represent academic approaches to the complex setting of collaborative virtual environments. None of them achieved to provide a simple extensible model attracting many third-party extensions. A lot of research work and commercial support is still required to produce a true killer application in the arena of 3D collaborative virtual environments.

## 4 CVEs Application Fields

To stress the utility of CVEs, we conclude by discussing three of their main application fields. We also enumerate example products and, in this way, we can see how research projects and commercial products are constructed, and how million-people communities can interact in a single virtual world.

### 4.1 Collaborative Work

When talking about collaborative environments, enterprise groupwork is one of the most common and interesting aspects to study. Working towards a common project, which can be divided into subtasks, is a definition which fits perfectly in the collaborative work field.

As an example to illustrate this section, we talk about Workspace 3D [10] from Tixeo, located in Montpellier, France. Workspace 3D offers a three-dimensional workspace where all enterprise members can talk together by video conference and collaborate in this environment represented by their avatar (see Figure 3). The environment provides portals where collaborative applications are loaded. Current tools are: slide projector, shared folders, desktop applications, whiteboard, shared browsing, and 3D model viewer.

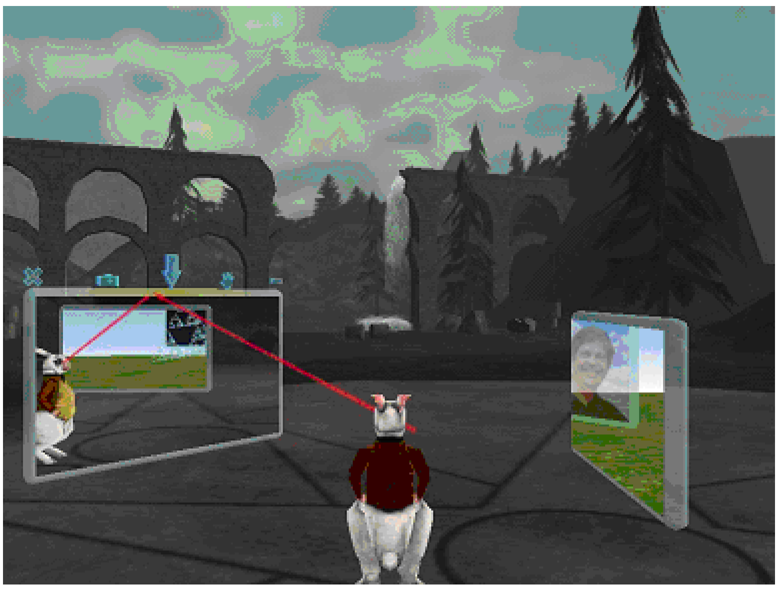

**Figure 2:** Avatar Displaying Portals to Other Worlds in Croquet

 

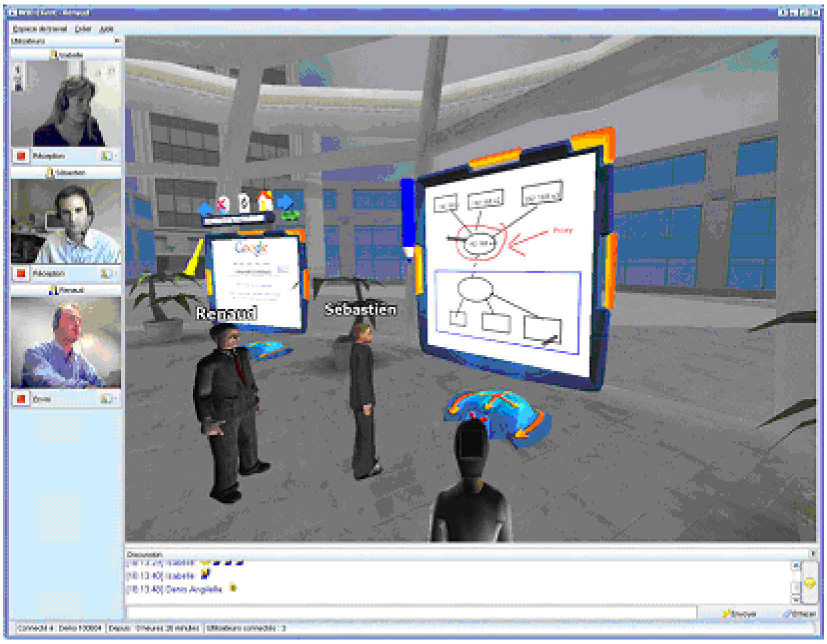


**Figure 3:** Videoconference with Browser and Whiteboard in Workspace 3D.

This is a proprietary solution, started from scratch, being an *ad-hoc* solution following the client-server model.

### 4.2 Learning

The learning field is one of the most explored when innovating in the collaborative setting. This also applies to collaborative virtual environments where we can find interesting contributions. From these projects we highlight MOVE, initially designed to recreate collaborative virtual environments in the learning scenario. MOVE distinguishes teacher and student roles and provides interesting tools like professor-guided navigation or moderation capabilities. MOVE was used in medical educational settings and a whole hospital was recreated in the virtual world.

Besides MOVE, there is the NICE (Narrative Based, Immersive, Constructionist/Collaborative Environment for Children) [11] project from the University of Illinois, USA. In this project, a virtual environment for children was created, enabling them to collaborate and talk between them. They can also find the teacher or other avatars controlled by the artificial intelligence of the system. The garden world is dealt in a persistent way, storing changes for next sessions. One feature to highlight in this project is the utilization of virtual reality devices, like head mounted displays (HMD), and data-gloves within a CAVE [11] environment.

### 4.3 Entertainment

Multiplayer games can be considered as a simple form of collaboration. Games have not only been one of the fundamental pillars for the development of graphical tools, but they are also one of the biggest focuses of virtual communities, since they have demonstrated that they can be used by more than five million users.

In MMORPG (Massive (ly) Online Multiplayer Network role-Playing Game) games, users can find a richer collaboration. This point is of special interest to us, not for the degree of collaboration, but for the great number of users that interact. It all began with Ultima Online and now hundreds of games of this kind already exist. Nowadays, games like Lineage II or World of Warcraft (see Figure 4) are a good example of the success of these persistent virtual worlds. Normally, a monthly fee is needed to support server infrastructure maintenance for these games.

On the other hand, every videogame company supports the multiplayer service in their releases. However, in more homogeneous platforms, like video consoles, more global solutions are adopted. One well-known solution which was started in 2003 for network based games in the Sony PlayStation 2 platform is the IBM Butterfly Grid, based on the OGSA (Open Grid Services Architecture). Butterfly controls the load of the server processes and when it determines that there are too many players connected to the same server, it automatically reconfigures servers with lesser use to support the most demanded games, and transfer players to these systems.

## 5 Conclusions and Future Directions

Throughout this article we have analyzed different projects, and we can now draw some conclusions. A logical one is that research projects experiment with more innovative approaches and technologies. This is the case of the system decentralization in Croquet. In stark contrast, commercial applications still deal with more traditional solutions (i.e. Workspace 3D).

These centralized solutions have proved to be insufficient in some cases, and enterprises sometimes cannot maintain infrastructures to support their products. For example, using a single server to synchronize the state of a virtual world means that all participants are sending their messages to the server from anywhere in the network. The traffic in this network spot is critical and it becomes a bottleneck, causing congestion. The enterprise thus needs to invest in expensive infrastructures to be able to sustain the load. These situations are happening now and there are clear examples in the videogame world. Some MMORPGs tend to be unsuccessful due to this fact, and sometimes even the developer company can go bankrupt due to high server costs.

As we have seen in the Butterfly Grid, with an initial funding provision, a more adequate infrastructure can be designed, providing a more scalable and robust service. However, these Grid services must be maintained. Therefore, to adopt a low cost solution, we need to adopt the **peer-to-peer** model. Moreover, this model is nowadays being much researched and there exist many innovative contributions, like Mercury [12], where a decentralized infrastructure is proposed, allowing zone partitioning.

Thanks to the important increments in graphic power and network bandwidth in user machines, the decentralization trend is viable and can compete with centralized systems with a higher cost and maintenance. Moreover, we believe that there will appear commercial products based on the peer-to-peer technology, following the line started by Microsoft in acquiring Groove's peer-to-peer collaboration desktop.

Nevertheless, 3D collaborative virtual environments are still less mature than their 2D counterparts. Some key reasons are the high cost of virtual reality devices, their overall bad quality and the costly expense of maintaining server farms for multi-user support. We expect that Collaborative Virtual Environments will reach the masses in the next few years thanks to advances in decentralized affordable technologies, low-cost virtual reality devices and probably with the help of less-intrusive mixed reality research.

**Figure 4:** Meeting in A Forest at World of Warcraft.

*Translation by the authors*

**Acknowledgements**

This work has been partially funded by the Spanish Ministry of Science and Technology through project TIC-2003-09288-C02-00.

**References**

[1] F.FIT and O.S. GmbH. BSCW (Basic Support for Cooperative Work), 2005. <http://bscw.fit.fraunhofer.de/>.
[2] Groove Networks. <http://www.groove.net/home/index.cfm>.
[3] George Coulouris, Jean Dollimore, and Tim Kindberg. Distributed Systems - Concepts and Design. Addison-Wesley, 2nd edition, 1994.
[4] Globus Toolkit. <http://www.globus.org/toolkit>.
[5] Distributed Interactive Virtual Environment (DIVE). <http://www.sics.se/dive>.
[6] Active Worlds. <http://www.activeworlds.com>.
[7] Pedro García and Antonio Gómez Skarmeta. "ANTS Framework for Cooperative Work Environments". IEEE Computer, pp 56-62, March 2003.
[8] Pedro García, Oriol Montalà, Carles Pairot, Robert Rallo, and Antonio F. Gómez Skarmeta, "MOVE: Component Groupware Foundations for Collaborative Virtual Environments". Proceedings of the 4th International Conference on Collaborative Virtual Environments (CVE), Bonn, Germany, September - October 2002, pp. 55 - 62.
[9] David A. Smith, Alan Kay, Andreas Raab, David P. Reed. "Croquet - A Collaboration System Architecture," First Conference on Creating, Connecting and Collaborating through Computing, 2003, c5, p. 2.
[10] Tixeo Soft. <http://www.workspace3d.com>.
[11] Roussos, M., Johnson, A., Moher, T., Leigh, J., Vasilakis, C., and Barnes, C. (1998). "Learning and Building Together in an Immersive Virtual World In Presence" vol 8, no 3, June, 1999, special issue on Virtual Environments and Learning; edited by William Winn and Michale J Moshell., MIT Press, 1998, pp. 247-263 and cover page.
[12] Ashwin Bharambe and Srinivasan Seshan. "MERCURY: A Scalable Publish-Subscribe System for Internet Games", First International Workshop on Network and System Support for Games, Braunschweig, Germany, April 2002.

 