Collaborative Virtual Environments – From Birth to Standardization

Chris Joslin, Thomas Di Giacomo, Nadia Magnenat-Thalmann
MIRALab - University of Geneva
{Joslin, Giacomo, Thalmann}@miralab.unige.ch

Abstract

Collaborative Virtual Environments, which have been referred to by many different names over the years, have been around since the early 90's, and some even before as hardware dedicated versions; now they are being standardized by the Motion Picture Experts Group (MPEG). This paper examines most of the significant systems over the past 13 years, discussing some of the most pioneering systems and the main methodologies implied, and whether their standardization is feasible.

Introduction

Fluid, unrestrictive, and realistic collaboration, between two or more people, in a virtual environment has been the dream of many possibly since computers (and emphasized by the improvement of graphics) were conceived. To be able to interact with people virtually, whilst unable to determine that it really is just bits and bytes, is the an ultimate goal. However to provide such a functionality, the kind of technologies required takes a lot of craft. It is not just a case of virtual reality that is simpler to set up compared to creating a collaborative virtual environment, it is virtual reality plus a whole lot more features: multiple users, interaction, networking and communication, servers and clients, virtual and real spaces, accuracy and multi-modal communication on levels of significant complexity etc.

CVE Systems, a History

These types of collaborative virtual workspaces come under many different guises: collaborative virtual environments (CVE), networked (NVE or NET-VE), collaborative workspaces (CW) and many more, but they all take on the same generally tasks; albeit for different purposes.

Leaving aside some of the simpler systems involving just text based collaboration (such as Multi-User Dungeons or MUDs) we discuss CVE research and development from the early systems through to the latest developments, the trends that have emerged and the concepts that disappeared. We have examined the mass of papers specifically related to CVE systems from the last 13 years, starting around 1990 with a system called “Reality Built for Two” [1]; there is one system in 1987 by Sim et al [2] which can be classified as a CVE, but was built using dedicated hardware for military training purposes. Since then there have been a multitude of systems developed, popularity peaked in 1997 with new developments generally falling off ever since; this is exemplified in Figure 1.

It is atypical in many latter papers to claim credit for innovations passed over many times; Area of Interest Management, which has been called many things in the past – but uses the same concept, has been reinvented almost as many times as there have been years. Scene segmentation suffers from similar problems, but here authors claim different methodologies. In this paper we attempt to examine them all, and provide an overview of CVE History as it has taken place over the last 13 years.

We conclude the paper with a discussion on MPEG; after so much innovation it is interesting to observe how and in fact what is being standardised and also how it might affect the average user of an MPEG system.
CVE Systems, a Beginning

Figure 1 shows a chronological list of all systems that have been extracted from the papers detailing the global CVE technology. Different versions of each system have also been added in order that developments and improvements can be easily visualised.

As previously mentioned, it is not our objective to analyse each system paper by paper, this has been done many times before and offers the reader no additional benefit; instead we focus on how each CVE system deals with the increasing requirements of connectivity (allowing more users to connect, improving collaboration and overall the user immersion etc). To this effect we examine the key areas and methodologies involved in CVE systems.

Sharing and Communication

Apart from some system specifics and topics which also relate to general virtual environments, which are out of the scope here, many of the work, carried over the duration of CVE systems, has been focused on either sharing or reducing the overall communication between clients – both are concerned with maintaining a general consistency between clients.

Talking about maintaining consistency refer to the task of ensuring that all natural and synthetic objects, as well as virtual humans (often the representation of each user, i.e. avatars) are in exactly the same position and orientation on all the clients connected to the same virtual world.

Object and environment sharing has its own core problems, which are not part of the focus on this section, being on communication. There are possibly five main categories for improving communication between clients (read here end-users), these are Network Topology, Dead Reckoning, Area of Interest Management (or AOIM), Scene Segmentation, and Compression; all are detailed in the following sections.
**Network Topology**

Discussed in most work devoted to CVE systems, and comparisons made in almost as many papers, the network topology of a CVE system is an issue that has never been really completely solved. The main contenders are as follows:

- **Peer-To-Peer** – Each client sends a separate message to all other clients collaborating in the same world.

- **Client/Server** – Each client transmits messages to a central server, which distributes messages to all the clients in the same world.

- **Multicast** – Each client transmits a message to a multicast address, to which all clients in the same world are connected.

The Peer-to-Peer topology is probably the oldest type of configuration (circa 1987), introduced when using a dedicated server was very expensive. This topology suffers the most from the drawback that for each client a separate message must be sent. This topology is now rarely used or discussed in articles dealing with CVE, but it does now and again appear in some form.

The Client/Server topology was introduced thereafter (circa 1992), it suffers from practically the same problems as peer-to-peer on local area networks (LAN), but for wide area networks (WAN) distribution is much better. It also offers several other advantages due to better handling of consistency and also often due to the server acting as a master over all connecting clients, which thus provides resolution over conflicts. Multicast was introduced later on when network infrastructures were improved (circa 1994); a superior topology in many ways apart from one: its adoption. Multicast requires that only one message is sent for all the clients connected, regardless of how many are connected; however, in many instances multicast traffic is dropped, and it is not standard in IPv4 so routers are not expected to support it. IPv6 should change all that as multicast support is included as standard. It also suffers slightly from a lack of domain master, but this has been solved.

---

**Figure 2. Network Topology Trend over Time**
by introducing a multicast server into the topology.

Figure 2 shows the trend for a topology normalised for the total number of papers presented during a particular year. Many combined systems (Client/Server and Multicast) are now emerging as expected, taking the advantages and best elements of both.

**Dead Reckoning**

Although possibly one of the first methodologies that has been introduced (circa. 1991) into CVE systems [3], it is possibly the least known to many developers. Dead Reckoning mainly concerns using negative correction scheme for the movement of objects (and in some cases virtual humans) whereby an object’s trajectory and velocity remains the same, unless a client is told otherwise.

This method works well for simple objects, especially slow objects, which are appropriate since initially this method has been developed for tank movements in military situations. Its main drawback has been that it suffers from problems associated with network latency, due to the fact that the time between transmission from the sending client, to the reception of update by the receiving client is the additional time an object will advance at its current course (and obviously in error), as exemplified in Figure 3.

Therefore for a fast moving object (such as a racing car) object travelling at 80KM/h going round a bend, the object will always have an approximate error of two meters. In addition, for complex interactions, such as virtual humans, the number of updates would be similar to the normal, continuous, update of positions. Dead reckoning does have strong advantages in computer games where interactions are much simpler. Although, apart from discussions based on complex mechanisms for dead-reckoning algorithms for virtual humans, the method is rarely mentioned.

**Area of Interest Management**

Area of Interest Management (often referred to as AOIM) has been apparently reinvented several times over its history; possibly due to its naming convention being changed. Introduced around 1993 [4], AOIM management deals with attempting to channel network traffic for a particular user with regards to other users – which also has some concepts of sharing inherently embedded into the system. AOIM basically consists of three main elements: Focus, Aura and Nimbus, which are shown in Figure 3.

- **Aura** – this is used to define a region which to a user desires interaction; this can be set for different mediums (such as audio, graphics etc).

- **Focus** – this is a sub-region that defines the actual focal point of a user; this could be considered in some respects as the view frustum of the user, but can be controlled on a more social level.

- **Nimbus** – this is the region which a user wishes be made themselves known; a smaller nimbus can be considered as a low desire to interact, whereas a large nimbus indicates a high interaction desire.
AOIM can be used for many aspects, even social limits within collaborative sessions. In terms of data transmission AOIM can be used to limit the data transmitted between clients to only data which is relevant (i.e. “of interest”) according to each client’s awareness and focus. As another example, AOIM can be used, in Computer Graphics, to dynamically set a particular level of detail for the geometric resolution of objects. The main drawback of AOIM is that it is often more difficult to control, and define how to limit the transmission – i.e. where, when, how, and how to do it smoothly are all considerations for AOIM.

AOIM still remains a popular method and has been featured in more than a quarter of all papers reviewed, albeit in different forms with some focusing on sub-elements (such as Focus or Awareness, the latter being not detailed here).

**Scene Segmentation**

This is mentioned separately from AOIM because, although they have similar characteristics, Scene Segmentation [5] is really focused on keeping users connected on completely different networks. Introduced around the same time as AOIM, scene segmentation is basically separating a large scene into smaller pieces and allowing clients to connect to the smaller sections and then move to another network or server as the user moves around the scene, as shown in Figure 5. Most work has been focused on size of segments, and how to segment efficiently. Some systems try and retain clients on the same server, but only provide a small segmentation of the global world, whilst others completely change the network address or server, depending on their topology.

Apart from reducing the overall network traffic by only considering clients in the same scene space, segmentation is also useful for retaining accuracy. Barrus et al [6] discuss this topic more extensively, but the basic premise is that as environments grow, the values used to represent the absolute positions of both objects and virtual humans needs to either increase, or to be less accurate. The outcome of this is users performing very close collaborative work far from the world centre will find the step between one position and another being quite large.

Scene segmentation is by far the most popular method of controlling network traffic, it is used extensively in many of the on-line games that are available today, which are discussed later, and can be found in nearly one third of all papers reviewed, in some form or another. This is possibly no wonder as it represents the segmented real world that we live in; it is not really a question of whether to segment or not, but how to do it, generically or dependently of the world, for the best usage.

**Compression**

Surprisingly one of the least mentioned topics with relation to collaboration is compression; however this is possibly due to
the lack of concern for such requirements. Unless there is a complex interaction involved between clients (such as the use of virtual humans), the transmission of updates is usually a very simple 32 bit value, and perhaps smaller if displacement vectors are used – of course associated with a simple Huffman table. Even in some of the new online gaming systems movements are restricted to run, walk, crouch etc, and in general the complexity, and hence the data transmission, is considerably low.

When virtual humans are considered [7], retaining up to 186 (excluding face animation) separate degrees of freedom, communication becomes much more complex; especially if each is represented as a 32 bit value, meaning upwards of 100Kbits/second for just body animation, and over 150Kbits/s with face animation included. In order to counter this kind of aggressive network use, several groups have produced a compression standard to reduce this figure considerably. Based on similar techniques found in video compression, virtual human animation was found to be best considered across its temporal space. This means that the connection between two joints in the same body have relatively no relationship, and that a joint is compressed based on its position in the next frame. Depending on the compression type used, streams can be reduced down to a ratio of 5:1 or more, and fits well into the region of a simple modem connection.

**CVE Systems, WWW Revolution**

It is interesting to note that CVE systems have been around way before the World Wide Web was invented; but have not been adopted on anywhere near the same scale. This is possibly because of their complexity and base requirements being much more demanding, or possibly the content being much harder to create – whatever it was it is also an interesting point to see that after the WWW birth, many CVE systems started to adopt the technology in order to allow users an easy access to the technology (circa. 1995).

Most systems use a Java interface to control the additional communication required, display like 3D in the browser window and also to handle the interactions from the user; although some systems are presented as plug-ins for a browser – allowing it to be simply a wrapper for the key technology underneath. Most of these systems use the Hyper Text Transfer Protocol to obtain the actual scene with an additional back channel in place to provide a way to communicate with other connected users.

Almost all the systems studied used little or no other form of network traffic reduction scheme as their main goal was to provide interaction in a local window. The trend for these kinds of systems has not wavered and there is a consistent percentage of systems based on WWW type access.

**CVE Systems, Present Day**

**Computer Game Systems**

It is possibly difficult to write any paper these days without mentioning collaborative games, and especially as they are possibly the most popular CVE systems to date. With home network connections increasing towards some tens of megabits and powerful graphics hardware becoming evermore common, more and more games are switching from single-player to multi-users experiences, where players are hooking up their consoles and computers to the Internet and joining a multitude of other players in a vast array of fantasy worlds, so elaborated it has become an addiction for some. From First-Person Shooters to Role-Playing, and Real-Time Strategy games, the range and scope of interactions are enormous. While at the beginning their implementation was mainly based on P2P, i.e. the main game model and the inputs being located on every computer, or simple Client/Server architectures (where the game module is stored on a Server and downloads locally), online games now tend to adopt server architectures that, by optimizing the required resources, allow standard PCs, with xDSL connections, to act as servers.

Possibly the key to their success is the balance of technology, whilst none provide high levels of interaction with specific devices (e.g. data-gloves) – such as might...
be expected in a CAD session – the immersion is quite high. Firstly, the collaboration level is advanced, many systems joining several hundred thousand participants to the same arena (and just released one system providing connections for up to one million – such as Rekonstruction [8]). The storyline and system allows participants to progress in the story “en mass” without bizarre repercussions or problems. This helps enormously as participants feel they can interact and change things.

Secondly, the graphics are usually stunning and extremely elaborated, using scene segmentation and smooth transitions between different segments (obviously with some hand made design behind the scenes most of the time) provide users with the feeling they are really inhabiting a very large space.

Thirdly, the user can interact easily, although not on a high level, they have access to a range of motions and in some cases can communicate with direct speech. These are limited to a set of specific idioms and gestures in order to preserve network, but thanks to careful planning and preparation most cases are considered.

**Motion Pictures Expert's Group**

The Motion Pictures Expert’s Group (or MPEG as it is commonly known) first met in May 1988 in Ottawa and has held regular meetings four times a year since. It is a well known standards body, first embracing video and audio and later (in 1996) it started to introduce Synthetic Natural Hybrid Coding (or SNHC) which basically involved the mixing of both natural video and audio, with synthetic graphical objects. This has progressed through many versions, revisions, corrigenda and is still an ongoing topic with the new Animation Framework Extension becoming Part 16 (all different modular elements of MPEG are broken down into parts) of the MPEG standardisation process. The main focus of standardisation is that whilst it is not the be all and end all of a technology, it is intended to make use of the best techniques of the time and must pass many rigorous checks before it can become a standard (both verification models and reference software must be provided). Therefore, MPEG can be considered not necessarily to be the state of the art, but the end of an era of struggling researchers attempting to obtain recognition and allow many others to directly use their research, although sometimes wrapped up with patents.

It is not the purpose of this article to introduce and discuss all the aspects of MPEG, there are good reference books on this subject including Pereira et al [9] which can be used for a broad overview and Bourges-Sevenier et al [10] which is more related to graphics, but to focus on one particular aspect: collaborative virtual environments using MPEG.

**Multi-User Technology**

As can be seen from the previous introduction section, MPEG was never intended to be used as a collaborative tool; in fact never with the real intention to be used as a graphics tool. However, due to the fact that the underlying structure of MPEG is so well designed, it was not a problem to add these elements directly into the specification without any major modification of the pre-existing technology. Recently, Multi-User Technology (MUTech) has been introduced to MPEG-4, Part 11, in order to provide some kind of collaborative experience using the MPEG specification.

On the whole it is surprisingly small and compact, consisting mainly of session commands and space structuring. The main reason behind this is that whilst earlier systems, described in the previous sections, detail methods for network control etc, MPEG is very modular and therefore already has many of these components already defined; these include the network layers, multiplexing of complex data and exchange methods, and obviously, graphical elements (both 2D and 3D).

**Sessions and Zones**

The main contributions consist of the introduction of a “Back-Channel”, which allows the normally unidirectional MPEG data flow to go in two directions (although
limited to a specific set of commands), a Multi-User Session, and a Multi-User Zone. These are explained in more detail as follows:

- **Multi-User Session** – This is basically the container for all the objects within the Session. Although there can be more than one session in a scene, they cannot be nested. The session can be considered a multi-user group node.

- **Multi-User Zone** – This is basically a physical space within a session, and is basically used for space structuring. Obviously multiple zones can exist, but also nested zones are also permitted.

- **Back-Channel** – MPEG has been, until recently, quite a passive system whereby a user requests a stream and then the stream is delivered. Dynamic requests were difficult if not impossible. This is why the Back-Channel has been introduced, in order that messages can be dynamically (during a session) transmitted back to the Server so that data can be distributed to other clients (or updated and sent back to the same client). This data generally consists of messages requesting to be joined to a session etc.

**MPEG and Previous Systems**

Whilst MPEG retains concepts for scene segmentation and object sharing, there are many elements that are either not supported or are left up to the system to support. MPEG is not attempting to standardise CVE configurations, it is merely attempting to standardise the connection configuration and all the elements necessary to support that structure. For example, Dead Reckoning is not generally supported; however this is more for the client to decide and the transport mechanisms for these updates are supported. Multicast groups for segmentation are also not present, which is a shame because this is one of the more interesting aspects of multicasting.

All in all though MPEG supports the majority of mechanisms either directly or indirectly, and with its flexible network structure and high-level compression it should not be long before CVE systems are emerging fully supporting the technology.

**Conclusion**

Collaborative Virtual Environments have progressed enormously over the past 13 years or so. The main challenges remain the same with many systems being introduced in order to tackle them. Most of the systems today address two key concerns: either how to provide basic CVE platforms to users in order that they can expand and develop more complex interaction methods (address mainly by the use of middleware, component based architectures and plugin based systems), or how to increase overall usage and make CVE platforms a standard rather than a speciality (through the use of java applications, PC based software, and web interfaces). There are of course others, working on better network integration, the development of multicast technologies, and smoother interaction.

MPEG has introduced its contribution to this area of research and overall it has tackled the problem with a simple design which incorporates many of the basic methods already in use. However, there are certain drawbacks to MPEG that could mean its adoption into the CVE world being delayed. These are outlined as follows:

- Firstly conforming to a standard is complex, and difficult. With proprietary systems the understanding is simple as long as the developer has a good design and also sufficient background knowledge. With a standard, in order to use someone else’s scene or model, especially in compressed form, the interpretation needs to be exact (literally bit for bit). This is very difficult with a standard as large as MPEG and makes it very difficult to obtain an initial working system.

- Secondly, stemming from the first argument, a developer can proceed down two paths (on a modular level if necessary): either the developer uses the MPEG reference software (ignoring the obvious costs in obtaining it) which is generally slow as it is only used to
demonstrate how something is done, rather than something which is optimised, or the developer can write their own code – obviously taking considerable time and experimentation. Both are not easy paths, although possibly it is slightly easier to use the reference software.

- Thirdly, also stemming from the second argument, due to the number of individual components in MPEG, it is only recently with the most powerful computers and dedicated developers that entire systems have been shown working together in real-time. Therefore, by adding multi-user systems together, the number of moving objects and additional data streams will be increased exponentially; hence it is difficult to determine what will be the upper limit of these systems, although it could be hypothesised that it will be less than most of the prototype systems around today.

Having said this even though MPEG, with its strict standardisation policies and cumbersome patents, has these problems it has generally always been the case and still comes out as being one of the most popular standards for video and audio in the world today; and with a little bit of perseverance graphics and collaborative virtual environments will also be dominated by the group in the future – there is no doubt that there are many groups striving for its worldwide adoption.

Acknowledgements

This research has been funded through the European Project ISIS (IST-2001-37253) by the Swiss Federal Office for Education and Science (OFES).

References