SPACE-TIME NAVIGATOR FOR ARCHITECTURAL PROJECTS

SUBMITTED: July 2005
REVISED: January 2006
PUBLISHED: February 2006 at http://itcon.org/2006/1/
EDITOR: C. Anumba

Sébastien Riendeau, Ph.D. student
Faculté de l'aménagement, Université de Montréal
email: s.riendeau@umontreal.ca

SUMMARY: *This paper describes a prototype that assists in the reading of various versions of a 3D model within an architectural design context. Our research is based on the problem that in architecture, the final object is represented but the design process remains hidden. 3D software, like other programs, is built according to a model in which each file relates to the saving of a state at a precise time. We propose bypassing this model by using a software tool which follows the various versions of a project, creating a new set of files that are interconnected and producing a 4D interface to navigate through these files. The interface, like chronophotography, juxtaposes and superimposes the various steps of a project in the same space-time. This makes it possible to have an overall picture of the design process or to highlight the modifications between different versions.*

KEYWORDS: *Architectural project, design process, visualization interface, 4D space, virtual space, project memory.*

1. INTRODUCTION

The aim of our research is to create a prototype to represent the creative process of an architectural project by reading the digital 3D files produced during the design process. Our research is based on the problem that actual architectural software assists us in producing a better solution but does not record the design process that leads us to this solution. The final object is represented, but the design process remains hidden, though many researchers in architectural education have shown the importance to designers of having a good representation of the design process. In the field of knowledge design reuse, many researches have worked to represent this design process. We will undertake a similar process, but we will not represent a high-level knowledge structure that requires indexing work. We will stay at the level of raw data and, specifically, 3D data. By using the concept of time combined with 3D data, we will work with a 4D representation like the one used in the research into 4D CAD, but we will represent the design process, rather than the construction process, and the common use of 4D CAD. The explored tool, called ArchiChronos, is a small program that compares digital 3D files and creates a 4D (3D + Time) space with these files. This technical tool, even though small, becomes interesting because it changes the way we visualise archives of the design process. This work is part of a broader research aimed at developing the means to improve the preservation and communication of the memory of an architectural project during the first phase of the design process.

In the next sections, we will show that the work with computers leads us, in architecture, to produce a huge quantity of data throughout the process of design. We will show how these data could be more exploited to represent the process of design using the computer’s potential. The creation of new tools to exploit computer potential must meet the specific needs of architects, so we will complete this section with a list of the needs on which we base our research.

Faced with the computer’s potential and the needs of architects, we will present the following question: *How to take advantage, with simplicity and dynamism, of the potential of multiple 3D files produced during the creation of a project, and how to document the evolution of ideas during the first phase of the architectural design process?*

We will present some projects that partly answer our research question. To complete the answer, we will add a theoretical research on the concepts of representing objects evolving in a timeline and on the many potentials of computer tools to represent objects that are evolving.

With these theoretical inspirations, we have worked to create a small program that compares digital 3D files and creates a 4D (3D + Time) space with these files. The development methodology is simply a recursive go and return from the program coding to the test with a set of files trying to have a clear and friendly visualization. The set of files used for the test was produced during the design process of a real residential project that we had already designed in 2004-2005. The result of the software will be present in section 6.

2. PRESENTATION OF THE PROBLEM

2.1 We produce a huge quantity of data.

Work with computers leads us to produce a huge quantity of data throughout the process of design. Computer science has created a true flood of information. “The quantity of raw data available is multiplying and accelerating” (Levy, 1997:12). This statement is also true for the process of architectural design. “Yet involving computers will surely increase the complexity and amount of information to be managed: each designer can generate up to 1000 pages a year of conceptual design documents. The original generator of the documents, as well as later viewers, needs help in organizing and navigating through this enormous amount of information” (Baya et al, 1990). At the end of a project, we not only have a plan or a model, but tens of different versions. This phenomenon originates in the very basics of computer science. Since data are stored digitally, it is possible to make an infinite number of perfect copies of a document through an entirely automatic process requiring no human effort. The cost of the copy is often infinitely small, even negligible, and only resides in working machine time and memory support. This changes the way we work. We are not modifying a model to produce a new version, as is done for cardboard models; with digital technology, we make a copy and it's this copy that we modify, but in keeping the original unmodified, we can go back in time. Secondly, data acquisition tools, such as digital cameras, scanners or digital video cameras, make access to a large quantity of data possible. Thirdly, networking also contributes to multiplying sources of information. Within the scope of this project, however, we are only concerned with the multiplication of digital models.

The method of copying files has the advantage of keeping traces of the creative process. Once the final result is obtained, the multiple files (if not already deleted or overwritten) become archives, which are “A depository of cultural materials slowly sedimented through time. This is the typical image of the archive when viewed as final product” (DeLanda, 2003:8). Very rarely used, these archives are accumulated in digital memory.

2.2 The huge quantity of data could be more exploited.

The huge number of data from architects could be more exploited to represent the process of design, thanks to the computer potential. If computer science can lead us to produce a large quantity of data, could it not also assist us to access these data? It is theoretically possible to develop tools that allow us to take advantage of the numerous traces produced during the design process by organizing and presenting them in a way that helps the architect or historian analyze them. The implementation of the application would allow delegating repetitive and mechanical tasks to the computer, allowing the user to focus on the more reflexive and creative ones. In order to do this, there must be a minimum of standardisation in the data to process. Furthermore, the data must be written in an accessible, open format. If data are produced with 3D software and only this software is able to read them, we will then be limited by this software in manipulating the data. In contrast, an OpenDoc relates to a “document-concentric model” (Fertig, 1996:410), where “documents are now the sun around which applications revolve” (Fertig, 1996:410). This desire to establish an open-document format, independent of software, is equally shared by Professors Guité and Parisel in their *Réflexion sur une uniformisation de données pour la description d'objets physiques* (Guité and Parisel, 2000:94).

Facing this mountain of digital documents, teams from the Yale computer science department suggest a review of the file access interface. “Desktop systems are inadequate for the task; new systems must support the management of these objects efficiently and transparently” (Fertig et al, 1996).

On the other hand, Winy Maas of the MVRDV (an international architectural office based in Rotterdam) is in favour of taking advantage of archives. “When you consider archives as only historical matter, then they are inert. But if archives are dealing with, say, accelerating knowledge, then immediately the inertia disappears, and the archive becomes a tool for progress or a tool to transgress knowledge from the existing situation into something new. The word archive – and in that we agree with you – is only interesting in its relation to the production of a future” (Maas, 2003:100).

In addition to this, Foucault suggests another way to avoid being caught in the traditional definition of archiving. “Switching from a conception of the archive as reservoir of information to one incorporating the uses of that information is indeed an improvement” (DeLanda, 2003). Applied to computer science, it could mean that in order to reuse the files in different contexts later, with other software, they must be saved in a standard way and with an open format.

As stated by D. Schön in 1992 and re-asserted by Chupin in 1999, “We suggest … that research should focus on the digital environments that increase the capacity of the designer to catch, store, manipulate, organize and think on what he sees” (Chupin, 1999).

2.3 New tools must meet the specific needs of architects.

The creation of a tool exploiting the computer potential to represent the process of design must meet the specific needs of architects. If an important number of interfaces are being developed for general needs (Fertig et al 1996, Gemmell et al 2004), we think it would be pertinent that architects develop their own interface, allowing management, organization and access to numerous documents based on their specific needs. The following are the specific needs on which we base our research.

2.3.1 The need to represent an architectural object in 3D

Because the object of study in architecture is three-dimensional, it is necessary to be able to visualize an architectural representation in 3D in a dynamic way (Tidafi, 1996). The architect builds 3D digital models in a dynamic environment where he can freely manipulate his model. However, the interactivity, virtual and 3D are flattened before the project output from the computer that created it. The architect’s communication medium with clients and other parties (architects, engineers, designers) is limited to two-dimensional plans, because the other parties do not possess or necessarily understand the specialized software required to manipulate these digital models.

In this context, it seems essential to have visual display software for 3D files independent of the software that has produced the files. This is what happens today with images: not everybody has the image processing software, but everybody has at least one version of image visual display software. Furthermore, the visual display software must allow the user to “walk” inside buildings and not just fly around them, as is the case with some navigators developed to view mechanical 3D drawings (i.e. *Volo View 3 or DWF Viewer 6.5* from Autodesk, or *ACIS 3D Open Viewer 7* from Spatial Technology). If we can walk in 3D in a video game, such as the famous Doom which came on the market in 1993 (Lowood, 2005), and other *first-person view* "game character's view of action" (Encarta, 2005), we should always be able to do it in architectural projects. The architectural walk-through is an important element of a building’s design.

2.3.2 The need to communicate the process

The architect is often asked to present the steps of his work. He must justify his decisions vis-à-vis the customers but also lead his colleagues and other players, such as engineers, to work in a common direction. After a project is finished, some architects are also asked to present their design processes to the public, to other architects or to students through conferences or published articles.

2.3.3 The need to understand the process

Architects need to understand their design process. The understanding of the main line of their design process is primarily essential in order to communicate it. We also agree that architects can progress more in their professional practice if they have tools do a “reflection-in-action and reflection-on-action” by asking themselves questions about the path covered to resolve problems (Schön, 1983).

2.3.4 The need to represent the design process

In an academic context, we express clearly the value of representing the design process because we make a link between the quality of the design process representation and the ability to learn and improve our architectural skills. “The reflections, the experiments, the starting material and even the presentations produced in workshops are very often untraceable or unusable. Consequently, the potential of integration and reuse of numerical information is only very partially exploited and the reflexive process desired within the framework of the trainings is often defective” (Guité, 2005).

The architectural object is designed in a period of time, sometimes concentrated into several hours, at other times extended over several years. To understand this process, we must be able to restore the time dimension. If perspectives can fake 3D, then the fourth dimension, the one of time, is too often forgotten.

Even though all the steps of a design are stored in memory, we can explain this difficulty by the fact that different steps are isolated from one another. Each file relates to a step that contains the state of the file at the moment the document was saved. The memory allocated to rollback is generally destroyed when a file is saved. Furthermore, these files are designed to be read only individually.

It would then be useful to be able to represent the design process and try to group together the steps which have been, up to now, isolated in each of the files.

2.3.5 The need to be able to enrich the projects by the reuse of old solutions

When an architect finishes a project, a solution is completed and presented, but there is an important quantity of interesting findings that have been abandoned for diverse reasons. Because it is often only the final version that is published and archived correctly in a portfolio, the traces and even the memory of the these findings are lost or inaccessible. We often miss the opportunity to construct a greater corporate memory. If we want to be able to reuse old ideas, we need to record not just the ideas, but also their context. In this way, we agree with R. Fruchter and P. Demian that “for knowledge to be reusable, the user should be able to see the context in which this knowledge was originally created and interact with this rich content” (Fruchter and Demian, 2002).

2.3.6 The need to have a fluid creative design process

Architecture is designed through a very creative process. On the one hand, the process can be blocked or slowed by complex manipulations, many of them essential, for the goal of archiving and later reuse, and there is a need for a simple and effective tool to support these steps. “During conceptualization, a designer is functioning as a performer, like a musician. And also like a musician, his instrument must be agile or the performance will be cramped and ideas lost” (Lakin et al, 1989).

3. RESEARCH QUESTIONS

The main research question can be stated as follows: how to take advantage, with simplicity and dynamism, of the potential of multiple 3D files produced during the creation of an architectural project, and how to document the evolution of ideas during the first phase of the architectural design process?

More precisely, how should the interface: (1) enable the user to see the evolution of a model in a time frame and represent this notion of space-time, (2) put many models side-to-side, thus transgressing the reading of unique files, (3) give access to 3D files in a dynamic way, but as simply as in a video game, (4) see the differences in the layers of information that make the digital models, (5) work on open files of universal format independently of the software that created them, and (6) maximize the archiving without harming the design process?

4. RELATED RESEARCH ON THE SUBJECT

4.1 In the field of 4D CAD

In the field of 4D CAD there are interfaces to represent 3D data evolution but not to represent the design process. “Since the early 1990s, there has been a growing interest in four-dimensional computer-aided design (4D CAD) for construction project planning” (Heesom and Mahdjoubi, 2004). Following this large definition, “4D CAD is a concept which combines an object-oriented 3D CAD model with time” (Bergsten, 2001). In architecture there’s three time-lines, there’s the time-line of the design process, the time-line of the construction and the time-line of the building’s life. If we look at the research related to the concept of 4D CAD, we only find researches that represent the second time-line, the one of the construction. This explains the more restrictive definition given by V. R. Kamat ant J. C. Martinez in which 4D CAD is the mix between construction schedules and 3D models that “focuses on communicating what component(s) are built where and when” (Kamat and Martinez, 2002). In that way, “4D CAD systems are used by contractors for visually checking the construction process” (Vries and Harink, 2005), but also for the designers to verify the feasibility of their design concept. In our research we will reuse a similar concept of time-line and the virtual reality tools of the 4D CAD, but with the goal of showing the evolution of the building conception, rather than the building construction.

4.2 In the field of architecture

In the field of architecture there are interfaces to represent the design process but only in 2D. Some interfaces already give an opportunity to students to explore the new digital potential to document their design process. The *Carnet numérique*, realized by M. Guité of Université de Montréal, is a digital environment that directs and accelerates the work of recording the traces of the design process. This environment can be compared to the portfolio or the bulletin board of the architect's "real" world (Guité, 2005). Another example of a project being used as support to communicate and record the process of design is the *Observatoire-laboratoire des technologies numériques* (De Paoli et al, 2002). This one is less directive on each project than the first one but brings the opportunity to compare many projects on an interface 2d. With this kind of interface, our research will be characterized by the use of an interface 4D.

4.3 In the field of knowledge design reuse

In the field of knowledge design reuse there are interfaces to represent the design process but not specifically the 3D data. Many researchers have worked to create tools to reuse knowledge from design projects. R. Fruchter and P. Demian define design knowledge reuse "… as the reuse of previously designed buildings, building subsystems, or building components, as well as the knowledge and expertise ingrained in these previous designs" (Fruchter and Demian, 2002).

A part of the research in the field of reuse knowledge is based on the vision of the design project as a rational problem. The researchers build tools that automate the design process by automated reasoning about previous cases. However, many are opposed to that vision and a growing number of researchers are now thinking that computer systems should only support designer to make design reuse (Demian, 2004:23). For example, in their paper about the project named ARCHIE, E. Domeshek and J. Kolodner "discuss [their] strategic choice to build a design aiding system as opposed to a system that generates designs on its own" (Domeshek & Kolodner, 1993). Many ways are opened in that new approach of reuse knowledge to support designer. "Highly structured representations of design knowledge can be used for *reasoning*... However, these approaches usually require manual pre- or post-processing, structuring and indexing of design knowledge" (Demian, 2004:20). An other way is to use lower structured representations of design knowledge. One example is the time-based approach that normally "requires a low overhead but results in a collection of disparate documents" (Demian, 2004:20).

In our work we chose to stay with the low overhead while working to regroup the disparate documents. The tools that we generally use juggled with knowledge, or knowledge with data. To communicate them to the user they need the use of a query-answer or a tree-diagram interface. In our tools we stay at the level of data and try to represent them as limpidly as possible to help the designer to see the knowledge among the data and, more specifically, among the 3D data.

5. THEORETICAL DEVELOPMENT

Our working procedure is based on the meeting of concepts and technologies. We could even say that the result is only a new assembly of ideas and existing technologies. We will present in this section the different concepts that are the sources of our project.

5.1 Inspiration for the representation of the fourth dimension

The representation of the notion of time is far from obvious. To find inspiration, we went through the history of graphical representation and we found on that topic a very rich period, the invention of chronophotography.

In a second step, we looked at various programs that produce documents which incorporate a notion of time.

5.1.1 Chronophotography

The history of chronophotography begins with the juxtaposition of images and continues with their superimposition.

The first step, in 1878, was the production of juxtaposed images. "E. Muybridge and E.J. Marey have given a first answer to the paradox of representing movement while searching to obtain a multiplication of snapshots. Since 1878, Muybridge had conceived a battery of 24 black rooms allowing him to produce a series of images of a body in movement." (Lemagny and Rouillé, 1986:73). A second step, in 1882, was the superimposition of images on the same film; however, the superimposition of bodies over each other gives a formless result. We notice this problem principally if the subject does not move enough in space.

> Marey has made an important step forward by producing a unique and synthetic image of movement: the "chronophotograph" (1882) makes 10 images per second on a unique and same plaque. The print obtained gives the 'structure' of a movement, but we must notice that is at the price of a certain unstructuration, or disappearance of the body: as if the 'truth of movement' eats the aspect, the shape, and substituted for it something like an aura (Lemagny and Rouillé, 1986:73).

Fig. 1, the sequence of movements of a man vaulting, presents an example of superimposition of images, whereas Fig. 2, woman walking downstairs, presents an example of juxtaposition.

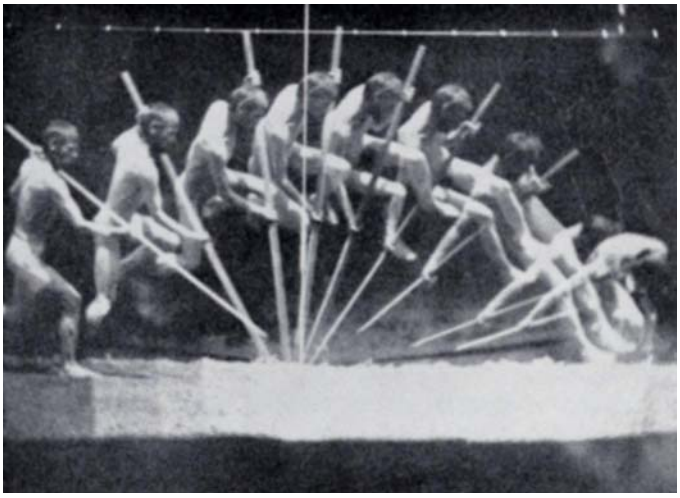

FIG. 1: "Sequence of movements of a man vaulting", Etienne-Jules Marey, Paris, 1889

A scandalous painting for its time, Marcel Duchamp's "*Nu descendant un escalier n°2*" (Nude Descending a Staircase, number 2), produced in 1912, was inspired by the chronophotographs of Marey and Muybridge, more particularly by Muybridge's "*Woman Walking Downstairs*" (late nineteenth century) and "*Ascending and Descending Stairs*" (1884-1885) (Rush, 2000:15). "I wanted to create a static image of movement," said Duchamp (Cabanne and Duchamp, 1987). It is interesting to compare, in Fig. 2 and Fig. 3, these two representations of the same scene. The first has a purely scientific goal, each image is showed individually. The second, with its superimposition, allows us to feel the movement, and to get a feeling of it without necessarily trying to quantify it.

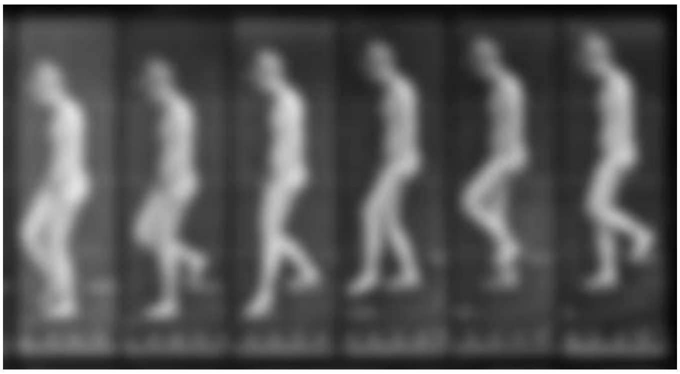

FIG. 2: Part of "Woman Walking Downstairs", Eadweard Muybridge, late nineteenth century

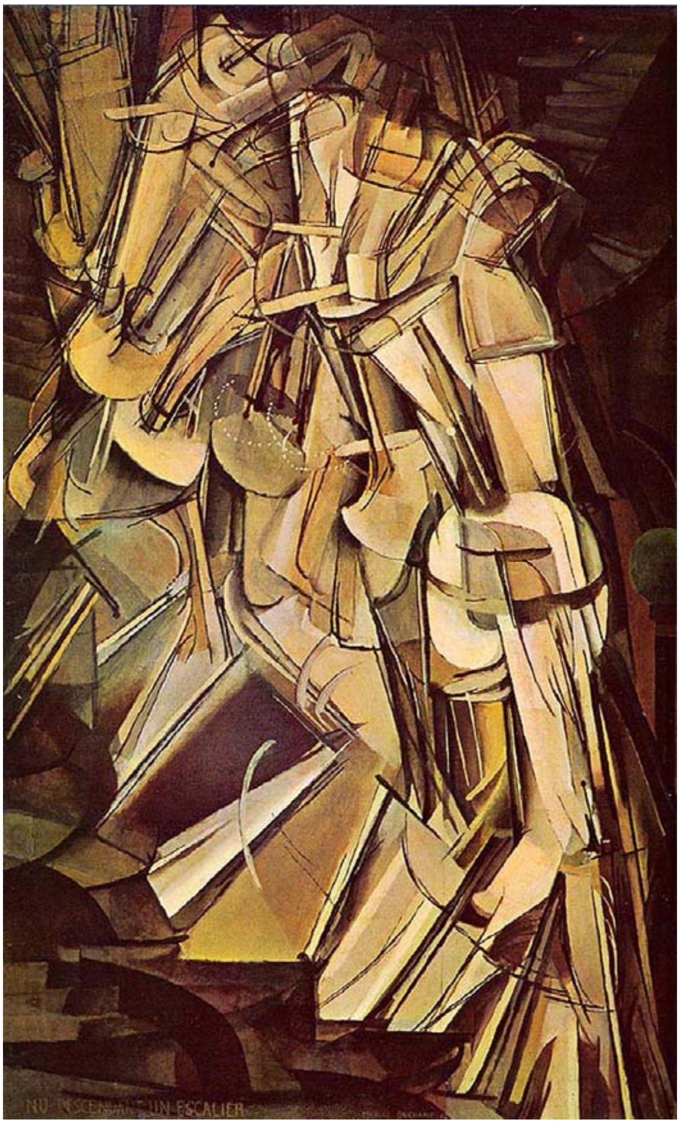

FIG. 3: "Nu descendant un escalier n°2", Marcel Duchamp , 1912, Oil on canvas 146 x 89, Philadelphia Museum of Art, Louise & Walter Arensberg Collection

An understanding of movement requires its decomposition. First, photography allows us to fix subjects that are moving too fast. "Photography reveals – unveils and fixes – the aspect of structures we do not see" (Lemagny and Rouillé, 1986:72). However it is not enough to understand movement : "[Fixing] photographically a flying bird does not teach us anything of the movement of the wings. Thus, we get to the paradox (noticed by photographers such as Albert Londe) that a snapshot image fixes – instead of "unveils" – the structure of the movement" (Lemagny and Rouillé, 1986:73). The understanding only happens with a series of snapshots. This series allows the evolution of a continuous object, too quickly and with no fixed point, to an object decomposed in a series of fixed images which we have the power to advance, send back and stop in time. The time of understanding is now independent of the time of the movement. If the number of images acquired is sufficient, we have archived the whole movement, but not necessarily at the key moments that allow understanding it. The great volume of information is not a direct access to understanding: synthesis is an essential step. Plato, in his dialogue with Phaedra, presents us his cutting-up concept:

> There are two ways for which it would be interesting to study methodically the virtue. … It is first of embracing in one view and bringing back to one idea the scattered notions in order to explain with definition the topic we want to process. … [The second] consists of dividing again the idea of elements following their natural articulations, and making the effort not to cut down anything, as an awkward butcher (Platon, 1964:151).

For example, pages of a book divide it, independently of the content, where chapters, depending on the content, would be the “natural articulation.” A body moving in space also has key moments: a waltz can be reduced to three steps, three images.

In conclusion, it is interesting to be able to present juxtaposition and superimposition of a space-time, at the same time having representations allowing more analysis and a better feeling of the project's evolution. However, because the sampled time in the case of a file backup is often independent of the content, in addition to presenting the stroboscopic sequence, it is judicious to present the key moments. In architecture, a key moment can be, for example, a space-time between two meetings with the client. Architects could annotate this delimiting moment, or the most revealing ones, by adding a pound sign to the names of their files, which would be interpreted and displayed on the time line.

5.1.2 Flash’s Key frame

Many programs work with the time notion, in particular those that work with video editing and animation. We examined the time line of the video animation software, Macromedia Flash. This time line allows us, by moving the cursor, to access a specific moment of the animation. It is also possible to select a complete period of time, which leads to a superimposition of space-time that is achieved by the older frames becoming, stepwise, more transparent. In our ArchiChronos we reuse the concept of navigating dynamically on a time line and the concept of a graduated superimposition of object, but applied to 3D.

5.2 Inspiration for a simultaneous reading of many files

Some programs have shown us how to superimpose different files and highlight their differences. The first one is, of course, Microsoft Word, which has a “Compare Documents” option. However, some small programs are specialised in these kinds of text comparisons and are widely used by computer programmers. “Beyond Compare” by Scooter Software is a good example of one. These programs have each their own colour code to represent the deleted, added and modified elements. In fact, they allow us to avoid the game of “seven mistakes” (a play for children consisting of finding the differences between two images). Architects must sometimes play this game when they communicate all the modifications made since the previous presentation of the project.

5.3 Potential of the visualization of spaces

We have adopted the concept of visualization of interfaces. The aim of these interfaces is not to produce documents, but to read or make visible those already produced. We can consider them as monitors or observatories. In this category are the Web browsers, image visualization software, some music players and Acrobat Reader. Each of them, in its own way, considers the question of information management, search, annotation, preview, and navigation in large documents, and each has characteristics that are specific to the format of the documents visualised.

Some would prefer that our tool allows the different parties to modify the documents. However, we will limit ourselves to the aspect of software dedicated to reading documents. We believe that it is helpful for an architect to use different programs to create a document, rather than the same program already used to observe the evolution of a series of documents. On the other hand, we have the vision of an architect as an orchestra conductor when he dialogues with other parties. He listens to the requests of the different parties, tries to understand the foundations of these requests, and uses his experience to translate them into volumes, materials and spaces, then he considers solutions. If the other parties express their requests directly in architectural forms, they deprive themselves of the architect’s experience.

5.3.1 The limited potential of the computer: its place near to man

We have adopted the vision in which the computer is present to support human thought. We use the power of the computer to display the data, but we leave the task of creating links and meanings among those data to the brain of the architect. We do not believe that the computer can resolve all problems: the computer can resolve mathematical/logical problems but not all problems can be expressed in a mathematical/logical way. This vision diverges from the Galilean vision of the universe. “The Universe […] is written in the language of mathematics” from (Galileo in 1623, quoted by Machamer, 1998:64). Following this statement the computer, as a number master, should easily replace human intelligence. However, if the physical world can, to a large extent, be written in a mathematical way, the universes of ideas, emotions or symbolism are much less accessible to numbers. Indeed, these universes can not be quantified and do not follow the rules of logic. “The computer has a perfect memory (with an unlimited capacity), a high computation speed, a great discipline in executing repetitive tasks and a total insensitiveness to fatigue” (Volle, 2000). However, even though very powerful, a computer is limited to the logical and quantifiable universe. Finally, a computer is very handicapped. “It does not know how to do a synthesis of a set of facts and draw a conclusion from it. It does not know how to take decisions. It has no imagination” (Volle, 2000). The computer should be perceived as what it is, a complement to human intelligence: it is man who is motivated by an objective and who has the spirit of synthesis and creativity. The position of the computer is to serve man in order to allow him to exploit his own aptitudes. The computer must be used to optimise the conditions favouring the emergence of creativity, of a spirit of synthesis, of imagination, of intuition. Thus, in our project, we use the mathematical capacities of a computer to put geometrical shapes in parallel and to represent archived data more clearly in order to allow the architect to have a better basis of thinking, to be better equipped to extract the meaning of data, and to have a better communication tool.

5.4 Potential of proximity with the past

We inspired ourselves from the very important proximity with the past that we have the chance to experience today. Our knowledge of history is phenomenal, compared to what those of other centuries could have known, for whom history was often limited to the oral tradition. Today, we can “see” dinosaurs. Technology allows us to reconstitute the past, making it accessible in a way never done before.

Recent history has become more and more important. We are surprised to see, at a party, a guest who came later than the others, seen in a photo summary of the beginning of the party. Photos are used ubiquitously and access to them is instantaneous. The transition from analogical to electronic allows a phenomenal memory capacity. “From 1956 to 1996, computers’ hard drives have seen their storing capacity be multiplied by 600 and the density of information stored multiplied by 720 000. Adversely, the cost of a megabyte [has been considerably reduced]” (Levy, 1997:39).

Ideally, the past is very near to us, within our reach, small, light, not very cumbersome, discreet and easily accessible. The *Ipod Shuffle* is a good example of ultra-light access to the past, because, yes, recorded music is a trace of a past event, but a trace rich enough to make us experience again the emotions of a moment. The virtual encyclopaedia, available on the Internet, is another very good example of accessibility to the past, making easier the task of trying to understand past events.

Technology can allow us to return to the architect’s studio at the moment the architect created a work and to see him thinking. It is with this in mind that we have built the interface of the present project. It has the objective of opening a door on the past, using a very lightweight tool which is easily accessible and simple to use.

5.5 Potential of the concept of virtual reality

We have recovered and exploited the concept of virtual reality in its philosophical meaning. “In philosophical acceptance, the virtual is only what exists in power and not in action, the field of forces and problems that tend to resolve its actualisation. The virtual is upstream of the effective or formal concretization (the tree is virtually present in the seed)” (Levy, 1997:55). Because the seed has the potential to become one thousand trees, depending on the context in which it grows, the virtual model has the potential at the moment of its actualisation in image to take diverse forms following the viewpoint requested.

Instead of creating a virtual space with fixed volumes, we have looked at creating a space that is twice virtual, in the sense that it does not have a fixed form itself. The interface (in VrmlScript) allows the user to enter choices and combinations of volumes, (juxtaposed, superimposed, coloured, invisible, black and white) and to create a virtual space dependent on these choices.

5.6 Potential of hierarchy of information

We have inspired ourselves by the way architects prioritise their documents. At first glance, an architect’s studio can look disorganized. If we look closely, however, there is an efficient enough structure without it being very rigorous. The drawings hung on the wall are references, those on the desk are probably in development or transition, those on the floor are of lesser importance, and sometimes there are piles of drawings conserved as alternate solutions, eventual archives, that might be destroyed, but only at the end of the project. This hierarchy evolves in time as well: a lost sketch of an idea to follow is brought back to the desk, the drawing on the wall is removed, after finally becoming a bit too old. In computers, however, the files can seem well ordered in a folder but they are always represented by icons of equal size and importance. It thus becomes difficult to tell one from another, and we have no understanding of the relative importance of the documents.

Associated with the concept of key moments or the natural articulation of a project, as presented in the paragraph on chronophotography, we have decided that we want to express these key moments. The addition of a pound sign ("#") to the name of a VRML file is of particular importance. This simple annotation allows it to be easily removed during the process, if the importance of the document diminishes.

6. RESULT: SOFTWARE DESCRIPTION

ArchiChronos version 1.0, the software that we developed, is a space generator. At first, it compares files of an architectural project and, then, it generates a 4D space and a user interface to navigate inside it. We have developed this software by using files produced during the design process of a residential project we already designed in 2004-2005.

6.1 Description of the user interface

Let's first look at what the developed interface allows us to see and how we can navigate with it. In Fig. 4, at the bottom of the main panel, we can see two square brackets that enclose a sequence of points. Each column of points corresponds to a file produced during a project's design process. Each line of points corresponds to different layers of information used in these files. The name of each layer is written to the left of the sequence of points. Behind the interface there is, on the one hand, a collection of juxtaposed models and, on the other hand, in the centre, a collection of superimposed models. By moving the square brackets, we can choose the steps that will be superimposed (inside the square brackets) and the ones that will be juxtaposed. The superimposed steps each have their transparency level, which allows us to distinguish them while having a preview of the transformations. On the side of each layer, a button allows us to select three different states. In the first state, all the volumes are shown in opaque grey, which allows us to represent the geometry at a precise moment of the process in a traditional way. The second state allows us to visualise the modifications between two steps, following a code of colours and transparencies. The third state is the one of invisibility: it allows us to keep visible only the layer we want to study. We can, for instance, choose to follow only the evolution of the structure from one model to the other, or the relation between the structure and the envelope of the building. After having chosen the desired parameters, we push the button "Go" and the 3D models take the desired position and appearance.

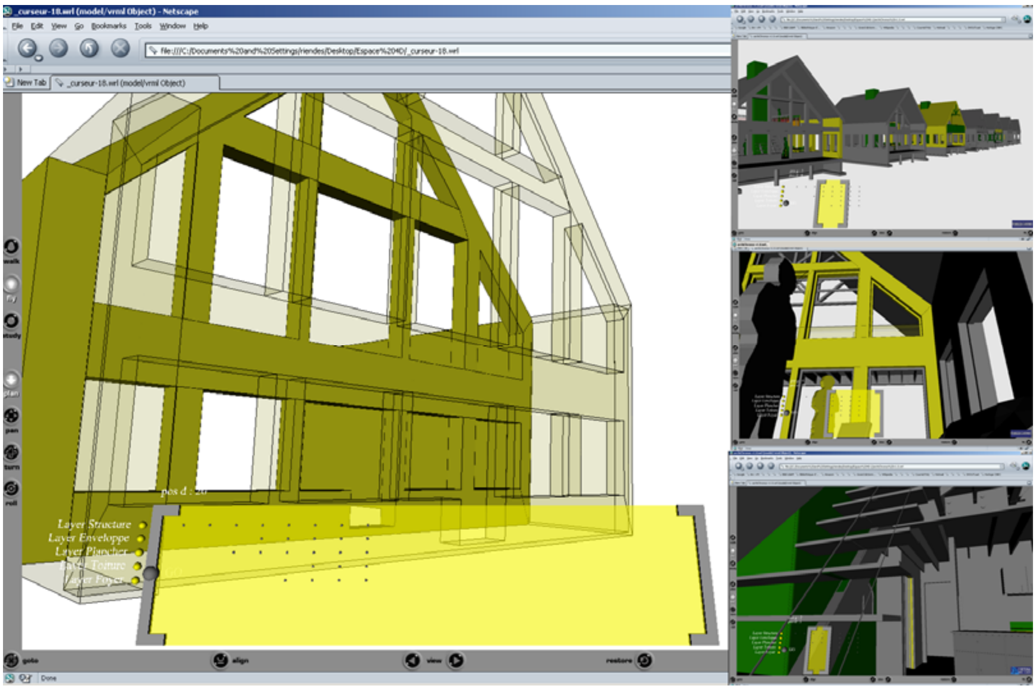

FIG.4: PrintScreen of the 4D interface

Here is the colour code that we have used: the unchanged volumes are in opaque grey, the new volumes in opaque green, the modified volumes in yellow (the old version in semi-transparent and the new one in opaque), and finally the deleted volumes in semi-transparent red. The volumes of the present time are opaque and the ones of the past are semi-transparent. The unchanged objects are in grey, while the modifications are highlighted with colours. The superimposed volumes in the interface are grey, with different transparency levels. The Table 1 illustrates these colour codes.

TABLE 1: Colour codes use in the interface

		The tree options								
		coloured			**black-white**			**superimposed**		
		colour	opacity	outline	colour	opacity	outline	colour	opacity	outline
Type	**unchanged**	grey	100%	-	grey	100%	-	grey	0-100%	yes
	new	green	100%	yes	grey	100%	-	grey	0-100%	yes
	modified (old)	yellow	10%	yes	-	-	-	-	-	-
	modified (new)	yellow	100%	yes	grey	100%		grey	0-100%	yes
	deleted	red	10%	yes	-	-	-	-	-	-

6.2 General description

Let's now look at how this interface was built by looking at the use-case diagrams of Fig. 5. In the upper left, the architect works on specialized 3D software that produces 3D files specific to this software. He must regularly export his model in VRML format. At the moment of visualizing his file, the observer launches the *ArchiChronos Interface generator*. This generator, as a first step, will rewrite the VRML files by comparing the modifications between the files. The information on the evolution of the volumes is added to the new files as well as the keys allowing modification of some of their variables. Furthermore, the new files are fragmented so that each of them only contains one layer of information. The VRML files production is done following the logic proposed by Professors Guité and Parisel, who have elaborated a standardisation system made of separated files carrying "a denomination made of a prefix for the type of project considered followed by a reference number to identify it" (Guité and Parisel, 2000). As a second step, the interface generator produces an interface in function of the files found. This interface, written in VRML and VRMLScript, can be used through a VRML viewer. These VRML viewers are a stand-alone program or more generally a VRML plug-in install on a Web browser.

To see his progression, the user has two new programs to use: the *ArchiChronos Interface Generator*, to create a new set of enriched files and a *VRML viewer*, to see the files. It would have been possible to include the *VRML viewer* (virtual reality visualisation and navigation within 3D worlds) directly into the *ArchiChronos Interface Generator* and have only one program, which would have given a more integrated result. However, it was faster and sufficient, for our experiment, to use an existing VRML viewer.

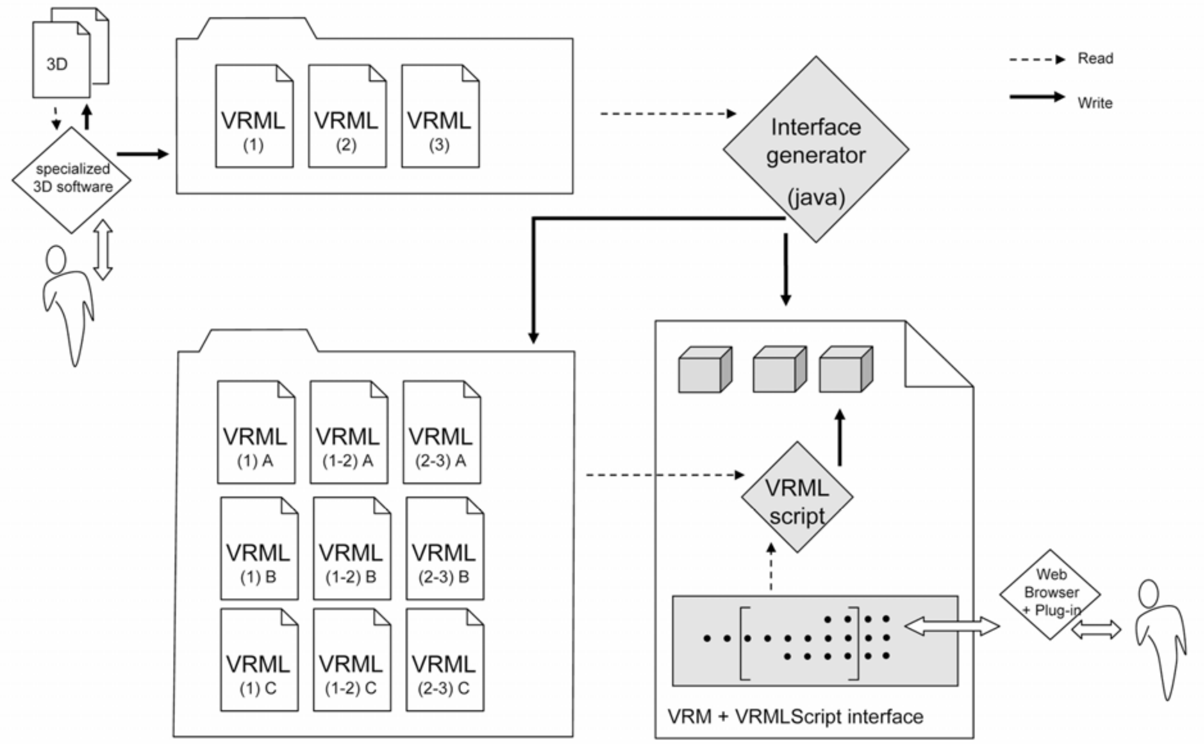

FIG. 5: Use-case diagrams

6.3 What's inside

Without describing each of the functions in detail, the structure of the interface generator (written in the JAVA language) is illustrated in Fig. 6. The *Controller* gives two names of Original VRML files to the *VRMLreader*. The *VRMLreader* parses (converts the VRML nodes in VRMLobject) the content of the two files and creates two vectors of VRMLobjects. *VRMLcompare* compares the content of each vector of VRMLobjects and generates new, enriched vectors of VRMLobjects. Then, the *VRMLwriter* generates the modified VRML files. Afterwards, each file is compared with the next one. The *InterfaceCreator* produces the visual display interface. An example of this repetitive production of new enriched files is illustrated in Fig. 7.

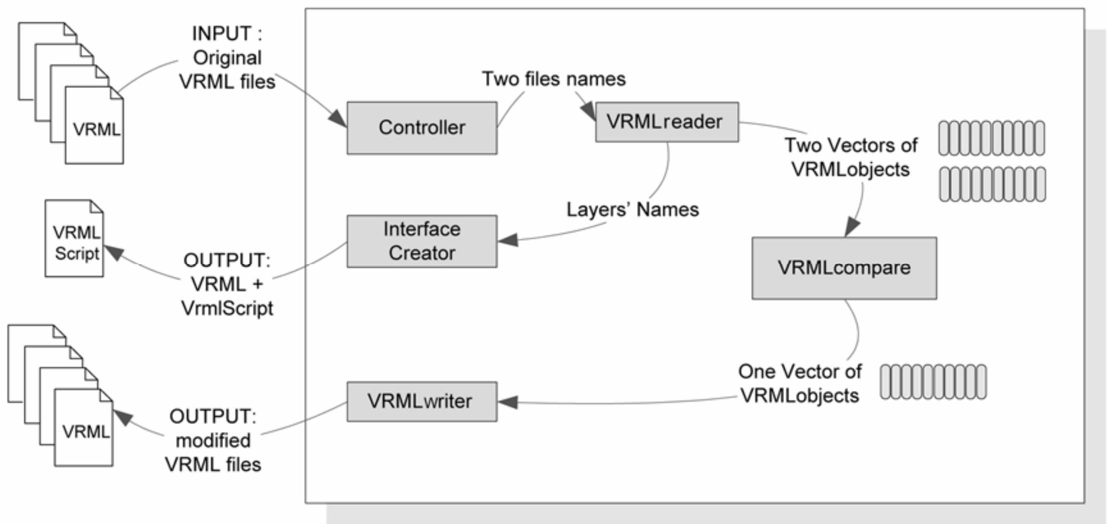

FIG. 6: Structure of the interface generator

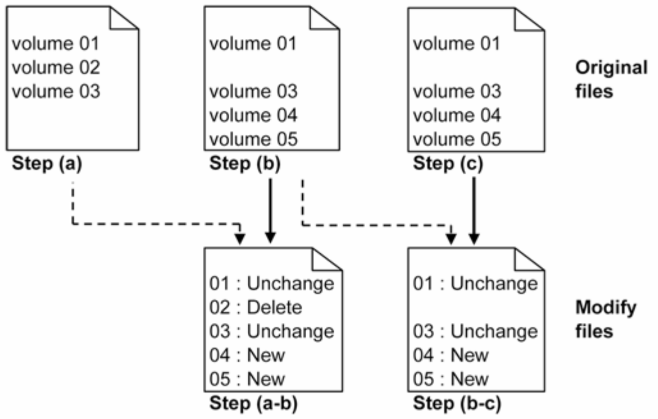

FIG. 7: Example of file comparison

The interior of the VRM + VRMLScript interface is illustrated in Fig. 8. The user intervenes in the different parameters by manipulating VRML volumes in the user interface. Each of these control volumes is associated with a script written in VRMLScript. Once triggered, the principal method gathers the different parameters and generates links, allowing the volumes of buildings to appear.

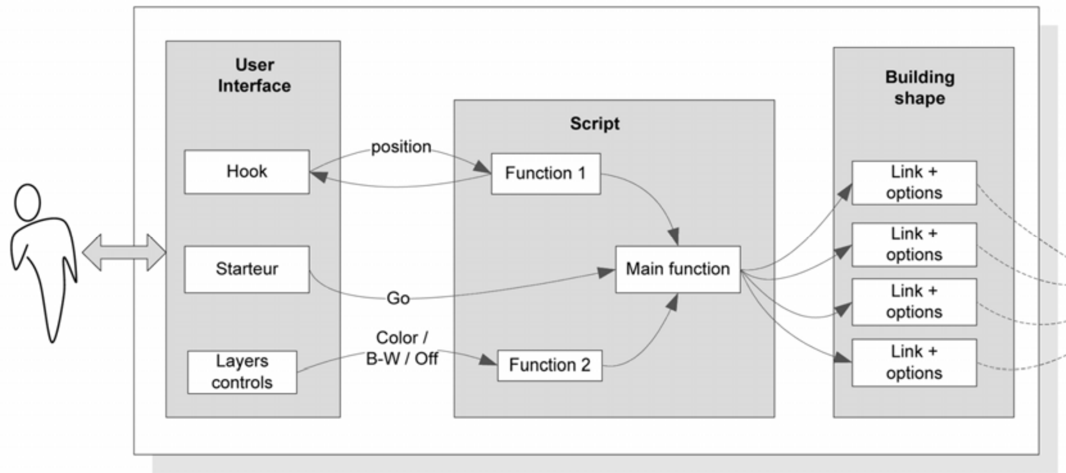

FIG. 8: Structure of the VRM + VRMLScript interface

If the JAVA application further divides the files, the VRML+VRMLScript interface allows us to link together different files and thus show them, at the same time, their juxtaposition and superimposition of different space-times.

6.4 The VRML choice

The choice of using the VRML technology, "Virtual Reality Modelling Language," results from several key arguments. In the first place, we wanted to recuperate the files produced by various 3D programs, and most modelling softwares used by architects allow an export to VRML. The big exception is AutoCAD, but small programs are available which can be used to make this conversion. In the second place, we wanted an open file format. On this topic, VRML files are alphanumeric files of ASCII format OpenInventer 2.0 (Saleh, 1996:94). In the third place, it was important to have the capability to navigate dynamically, freely, and be able to “walk” into the buildings. Once again, VRML allows this option. In the fourth place, VRML is not limited to navigate in static worlds, but allows animated objects and a great deal of interaction in the form of clicking on 3D objects tools to make things happen. Fifthly, the VRML will allow us, in future developments, to include images, animated images and sound files. One of its disadvantages, however, is that its drawing speed is poor compared to the majority of video games, but this was not so important for the context of our experiment.

7. CONCLUSION

The memory of an architectural process survives because of fragmented and scattered traces in the digital archives of the architect's studio. Barely used, these traces are, however, a gold mine that can help us understand, enrich or communicate the design process of the projects from which they are born. In order to exploit these traces, we asked the question: "*How to take advantage, with simplicity and dynamism, of the potential of multiple 3D files produced during the creation of a project, and how to document the evolution of ideas during the first phase of the architectural design process?*" Led by this question, we have found that the superimposition and juxtaposition of 3D objects, combined with the use of a colour code, can simplify and enrich the task of reading the evolution of a project. We find furthermore that by using the potential of the existing technology, such the VRML Viewer, it was easy to quickly create new dynamic interface to better visualize and manipulate 3D objects. By giving a better access to the old data, the ArchiChronos interface allows us to change our conception of archives by making them more accessible for daily use. The computer thus contributes to the improvement of the conditions for the intellectual work of the architect.

8. FUTURE DEVELOPMENTS

Now that we have the first version of our software, we plan to work with an architectural office to evaluate it in a design process. We also plan to improve it in one important aspect: our interface has some limits, as it only processes 3D digital objects, while the creative process is often expressed orally, with sketches, technical drawings, images or even sometimes gestures. A broader range of documents could be used to constitute the memory of a project. An interesting challenge would be to know how to archive the spirit of a project, or the state of mind of the designers at the moment they created the project.

9. ACKNOWLEDGEMENTS

First of all, we want to thank our friend, Julien Mazloum, for his invaluable help in the technical realisation of the software. We thank our parents, friends and employers who support us in their own ways. We also thank our research directors, Giovanni De Paoli and Manon Guité, who are our role models and who give us all the freedom we need to explore new ideas.

10. REFERENCES

Baya V., Cannon D. M., Lakin, F., Baudin, C., & Leifer, L. (1990, March 21). *The electronic design notebook (TM)*. Retrieved January 5, 2006, from http://www-cdr.stanford.edu/GCDK/vmacs/edn.html

Bergsten S. and Knutsson M. (2001). Publish electronically, *Proceedings of IT in construction in Africa 2001*(Eds, G, C. and F, B.) Mpumalunga, South Africa.

Cabanne P. and Duchamp M. (1987). *Dialogues with Marcel Duchamp,* Da Capo Press, New York, N.Y.

Chupin J.-P. (1999). L'alpiniste, le nageur et l'architecte, *Créativité & Informatique* (pp. 55-66). Lyon: Grame. (Translated from the French)

DeLanda M. (2003). In *Information is alive : [art and theory on archiving and retrieving data]*NAi Publishers, Rotterdam.

Demian P. (2004). *CoMem: Design Knowledge Reuse from a Corporate Memory*, Stanford University.

Domeshek E. and Kolodner J. (1993). Using the points of large cases. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing,* 7**,** 87-96.

Encarta (2005). first-person view. In Encarta Dictionary http://encarta.msn.com/dictionary_701706041/first-person_view.html

Fertig S., Freeman E. and Gelernter D. (1996). Publish electronically, *Proceedings of ACM SIGCHI Conference Vancouver,* British Columbia, Canada.

Fruchter R. and Demian P. (2002). Publish electronically, *Proceedings of CIB W78 Conference - Distributing Knowledge in Building* The Aarhus School of Architecture, pp. 93 - 100.

Galileo G. (1623). *Il saggiatore* (The Assayer) Academy of the Lynxes.

Gemmell J., Williams L., Wood K., Bell G. and Lueder, R. (2004) Publish electronically, *Proceedings of First ACM Workshop on Continuous Archival and Retrieval of Personal Experiences (CARPE '04),* pp. 48-55.

Guité M. (2005). Publish electronically, *Actes du colloque CAPIC 2005, Colloque sur l'avancement de la pédagogie et les technologies de l'information et des communications*Université Laval, Québec. (Translated from the French)

Guité M. and Parisel, C. (2000). In *Modélisation architecturale et outils informatiques entre cultures de la représentation et du savoir-faire*ACFAS, Montréal.

Heesom, D. and Mahdjoubi, L. (2004). Trends of 4D CAD applications for construction planning. *Construction Management & Economics,* 22**,** 171-182.

Kamat V. R. and Martinez J. C. (2002). Publish electronically, *Proceedings of 2002 Winter Simulation Conference*Institute of Electrical and Electronics Engineers (IEEE), Piscataway, NJ, pp. 1765-1770.

Lakin F., Wambaug H., Leifer L., Cannon D., and Sivard C. (1989, 1996). *The Electronic Design Notebook performing medium and processing medium*. Retrieved January 5, 2006, from http://www.pgc.com/pgc/home-stuff/papers-archive/edn-abstract.html

Lemagny J.-C. and Rouillé A. (1986). *Histoire de la photographie,* Bordas, Paris. (Translated from the French)

Lévy P. and Conseil de l'Europe (1997). *Cyberculture : rapport au Conseil de l'Europe dans le cadre du projet "Nouvelles technologies, coopération culturelle et communication",* O. Jacob, Paris. (Translated from the French)

Lowood H. E. (2005). Doom. In *Encyclopædia Britannica*. Retrieved December 29, 2005, from Encyclopædia Britannica Premium Service. http://www.britannica.com/eb/article?tocId=9389949

Maas W. (2003). In *Information is alive: [art and theory on archiving and retrieving data]*NAi Publishers, Rotterdam, pp. 100.

Machamer P. (1998). *The Cambridge Companion to Galileo*, Cambridge University Press.

De Paoli G., Riendeau S. and Langevin L. (2002). Publish electronically, *Actes du colloque CÉFES, Portes ouvertes sur l'enseignement à l'UdeM, Des stratégies pédagogiques différenciées*Université de Montréal. (Translated from the French)

Platon and Chambry E. (1964). *Le banquet; Phèdre,* Flammarion, Paris. (Translated from the French)

Rush M. (2000). *Les nouveaux médias dans l'art,* Thames & Hudson, Paris. (Translated from the French)

Saleh I. (1996). *Les langages de développement sur Web: Java, Schockwave, VRML,* Hermès, Paris. (Translated from the French)

Schön D. A. (1983). *The reflective practitioner: how professionals think in action,* Basic Books, New York.

Tidafi T. (1996). *Moyens pour la communication en architecture : proposition de la modélisation d'actions pour la figuration architecturale,* Université de Montréal, Montréal. (Translated from the French)

Volle M. (2000). *La question de l'intelligence des ordinateurs*. Retrieved April 4, 2005, from http://www.volle.com/rennes/intelligence.htm

Vries B. d. and Harink J. (2005). Publish electronically, *Proceedings of 11th International CAAD Futures Conference*(Ed, Brown, B. M. a. A.) Vienna, Austria.