The Problem of Time Scales in Computer Visualization

Mark Burgin[1], Damon Liu[2], and Walter Karplus[2]

[1] Department of Mathematics, University of California, Los Angeles
Los Angeles, CA 90095
mburgin@math.ucla.edu

[2] Computer Science Department, University of California, Los Angeles
Los Angeles, CA 90095
{damon, karplus}@cs.ucla.edu

Abstract. A new approach to temporal aspects of visualization is proposed. Empirical and theoretical studies of visualization processes are presented in a more general context of human-computer interaction (HCI). It makes possible to develop a new model of HCI as a base for visualization. This model implements a methodology of autonomous intelligent agents. Temporal aspects of the model are rooted in the system theory of time. This theoretical background provides for obtaining properties of time coordination. In particular, it is proved that some theoretically advanced models for computation and visualization cannot be realized by physical devices. Applications of the model, its implications for visualization and problems of realization are considered. The new approach does not refute or eliminate previous research in this area, but puts it in a more general and adequate context, systematizing it and providing efficient facilities for the development of computer visualization.

Keywords. Visualization, interaction time, system time, intelligent agent, virtual reality.

1 Introduction

Utilization of computers is based on three main components: computers as the technical component, users as human component, and human-computer interaction (HCI). Visualization is an important part of this interaction. At the very beginning, the main emphasis was on computers. Engineers and computer scientists tried to make computers more powerful, more efficient, and more flexible. At the same time, teaching computer skills and later computer science were organized.

The problem of visualization in HCI was understood much later [18]. In spite of this, research in visualization as a tool for HCI has been spectacularly successful, and has fundamentally changed computing. The main emphasis has been traditionally made on the means and forms of representation. Virtually

V.N. Alexandrov et al. (Eds.): ICCS 2001, LNCS 2074, pp. 728–737, 2001.

all software written today employs visualization interface toolkits and interface builders.

An aspect of visualization that has usually been downplayed until now involves *time.* However, time dependencies and temporal constraints are an important aspect of action, and failure to meet them leads to an important class of human errors; many of the errors associated with safety critical systems have a significant temporal component. For example, as is stressed in [9], the goal of physics-based graphics modeling tools is a compromise between getting visual realism and reasonable computational times. Another example of a situation where time is a critical parameter is given in [8]. The scenario concerns an air traffic controller whose task is to schedule the arrival of aircraft at an airport, while maintaining an adequate separation distance between them. Thus, in many cases time plays a crucial role, and cannot be relegated to a secondary consideration or to be treated implicitly. For example, in most problem-solving systems, time is hidden in a search process, while in formal grammars time is overshadowed by the sequences of transformations. Besides, time is important for business systems. People expect their computers to interact with them quickly; delays chase user away.

Various authors considered the problem of time in HCI from different points of view. However, in each case only a partial perspective has been considered: functioning as interface software, behavior of a user, or perception of a user. Thus, in the method of the state transition diagrams [11], interaction is represented as a net of tasks where time is not given explicitly. Grammatical approach [12] also gives some temporal relations without including time as a separate parameter. In [12], the main concern is user behavior during interactions with computer systems.

Our aim is to elaborate a complete model for computer visualization with an explicit structure of time dependencies and relations. Here we consider a single level interaction model. For example, we take into account temporal aspects of computational processes on the level of data but disregard operations with bits of information as well as time in physical processes in computer chips. Multilevel models are more complicated and will be considered in the future.

The conventional approach assumes that time is a one-dimensional quantity, made up of points, where each point is associated with a value [12]. The points are ordered by their values. The common concepts of later and earlier correspond to larger and smaller values of time, respectively. This view is compatible with the traditional psychological, thermodynamic, and cosmic views of time [13].

Another approach is based on temporal intervals rather than the standard view of mapping time to points on the real number line [1]. Time intervals are primary objects of observation and objects of learning by children.

Our approach to time is essentially different being based on the system theory of time [6]. The main distinctions are: multiplicity of time, non-linearity in a general case and multiplicity of formal representations.

Multiplicity of time means that each system has its own time scale. Only interaction between systems makes explicit various relations between times in each of the systems in question. In addition to this, processes going on in the same system may also develop in different time scales.

Non-linearity of time means that the order between time elements (points or intervals) is not necessarily linear. For example, let us consider a cartoon which relates a story that goes on in 1900 and covers a one year period of time. Then time in this cartoon will be cyclic because each display of this cartoon repeats all events in the same sequence. Another example is given by a complex computer game, like Civilization, which is played on two computers that have no connections. In this case, we cannot say that one event in the first game happened earlier than another event in the second game. Thus, there is no linear time scale for both games.

Multiplicity of formal representations means that different formal structures may be corresponded to time. For example, time is usually represented by a real line (as in physics or biology), system of intervals on the real line [1], system relations (as in temporal and dynamic logics), and natural numbers (as in theory of algorithms and theoretical models of computers: Turing machines, Petri nets, RAM, inductive Turing machines [7], etc.). To explicate time in visualization to a full extent, we need a model for human-computer interaction which represents the time issues in visualization.

2 The M-Model of Human-Computer Interaction

HCI is a process of information transmission and reception, which are called *communication.* Consequently, HCI is a kind of communication. Usually two types of communication are considered: *technical communication,* which is the base of information technology, and *social communication* [2]. HCI, as the foundation on which society is structured, is both technical and social, including both human beings and technical systems. Consequently, it demands a specific approach.

The communication process as a total event has been a subject of many studies. Many models or structural descriptions of a communication event have been suggested to aid the understanding of the general organization of the process. Models provide clues that permit predictions of behavior and thus stimulate further research.

There are *static* and *dynamic* models. Static models represent the system in which the communication goes on. Dynamic models feature its functioning.

The simplest static model consists of three elements: *sender, receiver,* and *channel* (Figure 1a). The simplest dynamic model also consists of three elements: *sender, receiver,* and *message* (Figure 1b).

We use the more detailed model from [5], which incorporates both static and dynamic features of the process. It is presented in Figure 2. Connection here consists of three components: the *communication media* (*channel*), *message,* and *communication means.* Communication means include language for coding and decoding the message, transmission and reception systems, etc.

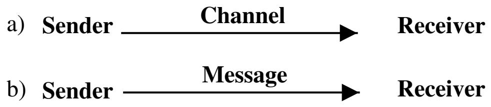

Fig. 1. The simplest models of a communication event.
a) Static model; b) Dynamic model.

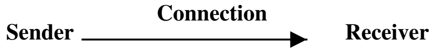

Fig. 2. Synthetic model of communication.

All structures (1a, 1b, 2) are special cases of a more general structure that is called a *named set* or a *fundamental triad* [5]. Besides, these structures are models of a single communication act. They are "elementary particles" of a communication process when the sender and the receiver exchange their roles. A complete model of communication as a whole process is complex and is built from such elementary models. Thus, the cyclic representation of communication is given in Figure 3.

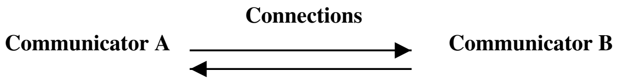

Fig. 3. The cyclic representation of communication process.

At the beginning, HCI was correlated with a general schema of communication. However, the spectacular development of computer interface changed this situation, and HCI became a mediated communication. Actually, in many cases communication in society is mediated by technical devices such as telephones, facsimile machines, satellites, computers, and so on. But these technical devices are only examples of advanced communication channels, while the human-computer communication possesses a mediator of a different type. We call this mediator the *communication* or *interaction* space. Thus, we come up with the following schema (cf. Figure 4) which represents the *mediator model,* or simply, *M-model* of HCI.

The communication space is not simply a channel because both participants of the communication, the user and the computer, are represented in the corresponding communication space by special objects, and these objects interact with each other. We call these objects *communication representatives* of the user and the computer, respectively. One real object or person (a computer, program or user) may have several communication representatives (e.g., avatars of the user). Communication functions are delegated to these representatives. Real parties of the interaction process (users and computer programs) control these representatives, but their behaviors have to be described separately. An efficient tool for such description is given by multi-party grammars which portray, in a formal way, actions of several parties [12].

User
Communication Space
Computer

Fig. 4. The relationship among user, computer, and communication space.

For the majority of software, communication space is the screen of the monitor of a computer. On the screen, as a communication space, a user is represented by a cursor and/or special signs (such as >) denoting commands together with the fields for these commands. A computer (a program) is represented by different icons and/or special signs denoting computer messages to the user together with the fields for these messages. However, current software does not capture the needs and preferences of the end users.

Only in some cases, the communication space is a virtual reality. In the future, VR will be the main type of communication space. The way to achieve high quality HCI is based upon personalization of the user representatives [16] and intellectualization of the computer representatives [19] in the communication space.

3 Multiagents to Enhance Computer Visualization

Visualization, according to [10], can go on different levels. The lowest one is *geometric modeling,* when only geometric forms are reproduced. Then goes the *physics-based modeling,* when the physical rules governing interaction of the visualized systems are taken into account and incorporated in visualization programs. The highest level is the *behavioral modeling.*

Agents may be used on each level but they become essentially efficient and even necessary only on the third level: behavioral modeling. This modeling always includes animation and also has three levels. On the first, *predetermined level* rules visualization and animation of a system **R** but ignores the changes in the environment of this system, in particular, the interaction of **R** with other systems. The next, *stimulus-reaction level* is based on a direct reaction of the visualized system to the current state of its environment. The highest, *cognitive level* presupposes learning of the visualization program. Means for such learning have been developed in artificial intelligence. The interaction of computer graphics and artificial intelligence is now on the merge of explosive growth as a new breed of highly autonomous, quasi-intelligent graphic characters begins to populate the domains of animation production, game development, and multimedia content creation, as well as distributed multi-user virtual worlds, e-commerce, and other Web-enabled activities [10].

In stratified visualization, image is stratified, i.e., separated into several strata, allowing to compute data for visualization of these strata by different devices (processors). The two natural ways of image stratification are localization and multilayer imaging. *Localization* means that the whole picture is divided into several parts and visualization information for each part is processed on a separate device. Another kind of localization is the systemic localization when a separate system, for example, a character in animation, is visualized autonomously. *Multilayer approach* presupposes that we have a dynamic picture that is changing in a different mode for some of its components. For example, in animation the whole picture is treated as composed of some active moving beings (people, animals, etc.) and things (cars, planes, etc.), and the background that changes much slowly. This makes it possible to animate moving things and beings separately from the background. Thus, two strata are utilized. Often, disjoint moving things and different beings are also animated separately, giving birth to many other strata.

Logical means for interpretation and investigation of stratified visualization are provided by logical theories of possible worlds [14], which were formalized by logical varieties [4]. Under certain conditions, this can considerably enhance the efficiency of computer visualization.

Structural means for realization of stratified visualization are provided by autonomous intelligent agents and multiagents. Here multiagent means a multiagent system [15] that acts as a single agent. In nature, a hive of bees or a colony of ants are examples of multiagents. Usually agents are considered as semi-autonomous computer programs that intelligently assist a user with computer utilization. They act on behalf of a user to carry out tasks.

Levels of behavioral modeling correspond to the types of software agents that are used for such modeling. According to [20], we have the following types of autonomous agents: *reflexive, model based, goal based,* and *utility based* agents. This linear typology is developed into a three dimensional classification with the structural, functional, and operational dimensions.

Along the structural dimension, we have three grades: 1) agents only with a stimulus-reaction system (SRS); 2) agents with SRS and a world model (WM); 3) agents with SRS, WM, and self-model (SM).

Along the functional dimension, we have the following grades: 1) action; 2) perception and reaction; 3) perception, reaction, and reasoning; 4) perception, reaction, reasoning, and mission orientation; 5) perception, reaction, reasoning, mission orientation, and evaluation; 6) perception, reasoning, mission orientation, evaluation, and self-transformation.

Operational dimension is determined by algorithmic classes that realize agents. For example, finite automata agents (level 1) are at the beginning of this dimension while Turing machine or RAM agents (level 2) are much higher. Super-recursive agents (level 3) are even higher [7].

In this three dimensional space, reflexive agents have the coordinates (1, 2, 1), model based agents have the coordinates (2, 2, 1) or (2, 3, 2), goal based agents have the coordinates (3, 4, 2) or (3, 4, 3), and utility based agents have the coordinates (3, 5, 2) or (3, 4, 3).

In the context of the M-model of HCI, we have agents of three types: 1) agents for a user (users) or U-agents; 2) agents for the software system or S-agents; 3) agents for the interacting media or M-agents.

As an example, let us consider visualization in a computer game, in which a player competes with different characters (dragons, tigers, bandits, etc.) who try to prevent the player achieving the goal of the game. As a rule, the user is represented on the screen by some character. According to the multiagent technique, this character is visualized by a U-agent. Characters who are competing with the user are visualized by a S-agent or S-multiagent. The background of this competition is realized by a M-agent.

Cognitive technology for agents design [10] makes it possible to create an intelligent virtual reality in the game. The same is true and even more urgent for training programs based on simulation of different characters. As examples, we can take military or police training programs which, to be realistic and thus efficient, have to employ technique of intelligent agents and cognitive visualization that can reflect adequately temporal aspects.

4 Dynamics and Time in Computer Visualization

According to the system theory of time [6], each system induces three categories of time: 1) *Internal* or *system* time; 2) *External* or *environmental* time; 3) *Intermediary* or *connective* time.

Let **R** be some system.

Definition 1. *Internal time of* **R** *is the time in which the whole system* **R** *or some of its subsystem functions.*

Definition 2. *External time for* **R** *is an internal time of its environment as a system.*

Definition 3. *Connective time for* **R** *is the time in which the processes of interaction of* **R** *with its environment goes on.*

Remark 1. According to these definitions, internal time of any system **P** that exists outside **R** is an external time for **R**. It implies that there are, as a rule, many different external times for a system as there are many systems that interact with **R**.

The existence of a variety of times is compatible with a logical approach to system representation that is based on logical varieties instead of calculi [4]. If time is included in the description, the variety of representation consists of dynamic logics (implicit representation of time) or temporal logics (explicit representation of time).

With respect to the general time classification, which is given above, we have three distinct times related to visualization:

1. System time of a user.
2. System time of a computer or of a separate computer program.
3. System time of the interaction space, which is the intermediary time for both the user and computer.

As a result, the interaction space is used for synchronization of the system times of a user and a computer. These times are essentially different. The system time of a user is a psychological time of a human being [6]. It is determined by the processes in the psyche. As interaction with a computer demands some intelligence, conscious processes play a more important role in the user's internal time than unconscious processes. If we compare conscious processes in the brain with information processing of a computer, we see that system time of the computer, which is defined by this processing, is much faster than the psychological time of a human being. That is the reason why computers perform calculations better than most people.

It should be recognized that any computer has not a single time but many, because different programs are functioning mostly in their own time scale. This is especially evident when a computer is utilized in the time-sharing mode.

Each computer has one *gauge time,* which is used for the synchronization of all computations and consequently, of all other internal times in this computer. This is necessary for the organization of digital computations, where time is discrete. It is assumed that analog computations go in continuous time [21], but actually their time is only fuzzy continuous [3].

The gauge time of a computer is the time of its clock, signals of which gauge time in all subsystems and processes in the computer. This is similar to the situation in society: each person has his or her own internal or inner time, but all synchronize their activities by physical time which is measured by clocks and calendars. The interaction space provides a similar possibility. Namely, it is useful to have a model time in the interaction space. The aim is to synchronize all processes in the interaction space. Now the time of computer clocks is used as the model time of the interaction space. However, in the future, a variety of virtual reality implementations will bring up with many different kinds of time. Even now we have some differences. While the physical time is linear (at least, it is assumed so), the time of computer clocks is cyclic. This distinction caused the famous Y2K problem, which troubled so many people at the end of 20th century. The cycle of time in computers is equal to a hundred years. Computer games also have a cyclic time but with much shorter cycles.

An important problem arises from existence of distinct time scales in different systems. This is the problem of time coordination. For example, computer time is much faster than human time. Consequently, if a computer displays on the screen the results of arithmetic calculations as they were produced, the user would not be able to discern these results because human visual perception is much slower than modern computers.

For a system $\mathbf{R}$ to interact properly with another system $\mathbf{Q}$, it is necessary to have a time coordination function $t_{c_{\mathbf{RQ}}}$. This function maps homomorphically time $T_{\mathbf{Q}}$ in $\mathbf{Q}$ into time $T_{\mathbf{R}}$ in $\mathbf{R}$. Here, homomorphically means that $t_{c_{\mathbf{RQ}}}$ preserves a chosen structure in $\mathbf{Q}$. For example, our conventional time has the structure of an order set [13], i.e., if we have two moments of time, then one of them is less (earlier) than the other. Homomorphism means that the relation to be earlier, or equivalently, the dual relation to be later, is preserved. Dynamic properties of systems and their investigation involve topological properties of time. With respect to time coordination functions, it implies continuity, or more generally, fuzzy continuity of such functions. Properties of fuzzy continuous functions are studied in [3]. These properties are important for time coordination.

Here we consider one example that has important practical implications. Many theories of super-recursive algorithms are based on infinite computations [7]. Practical realization of such computations, which took place at the finite interval of the user time, might extend enormously power of computers [22]. However, their realization is a very complicated problem which depends on existence of the time coordination functions of definite type. In particular, we have the following results.

Let $T_{\mathbf{U}}$ and $T_{\mathbf{C}}$ be inner times of user and computer, correspondingly. We suppose that both times are linear, and are subclasses of the real line.

Theorem 1. *If there are only fuzzy continuous time coordination functions* $t_{c_{\mathbf{CU}}} : T_{\mathbf{U}} \to T_{\mathbf{C}}$, *then it is impossible to realize infinite computations.*

A proof of this theorem is based on the following property of fuzzy continuous functions, which is proved in [3].

Theorem 2. *A real function is fuzzy continuous if and only if it is bounded.*

Note that realization of super-recursive algorithms essentially extends the capacity of intelligent agents for visualization.

5 Conclusion

A model for the dynamic representation of human-computer interface developed in this paper is used for explication of the temporal aspects of computer visualization. The results obtained show that there are several time scales and consequently, multiple times in virtual reality. This important feature has to be taken into account in visualization design as well as in software design.

These peculiarities of the interaction space and its time characteristics are utilized in the Virtual Aneurysm system, developed at UCLA [17].

At the same time, the problem of different time scales is urgent not only for visualization but also for HCI design in general. Development of HCI tools and research in this area made clear that this is a critical factor for improving quality and usability of HCI [12]. Consequently, the dynamic model of multi-system interaction proposed in this paper is also aimed at further development of HCI.

References

1. J.F. Allen, "Towards a general theory of action and time," *Artificial Intelligence,* vol. 23, pp. 123–154, 1984.
2. K. Brooks (Ed.), *The Communicative Arts and Sciences of Speech,* Merrill Books, Columbus, Ohio, 1967.
3. M. Burgin, "Neoclassical analysis: fuzzy continuity and convergence," *Fuzzy Sets and Systems,* vol. 70, no. 2, pp. 291–299, 1995.

4. M. Burgin, "Logical tools for inconsistent knowledge systems," *Information: Theory and Applications,* vol. 3, no. 10, pp. 13–19, 1995.
5. M. Burgin, "Fundamental structures of knowledge and information," *Academy for Information Sciences,* Kiev, 1997. (in Russian)
6. M. Burgin, "Time as a factor of science development," *Science and Science of Science,* vol. 5, no. 1, pp. 45–59, 1997.
7. M. Burgin, "Super-recursive algorithms as a tool for high performance computing," *Proc. of the High Performance Computing Symposium,* pp. 224–228, San Diego, California, 1999.
8. B. Fields, P. Wright, and M. Harrison, "Time, tasks and errors," *SIGCHI Bulletin,* vol. 28, no. 2, pp. 53–56, Apr. 1996.
9. N. Foster and D. Metaxas, "Modeling water for computer animation," *ACM Communications,* vol. 43, no. 7, pp. 60–67, July 2000.
10. J. Funge, "Cognitive modeling for games and animation," *ACM Communications,* vol. 43, no. 7, pp. 40–48, July 2000.
11. M. Green, "The University of Alberta user interface management system," *Computer Graphics,* vol. 19, pp. 205–213, 1985.
12. H.R. Hartson and P.D. Gary, "Temporal aspects of tasks in the user action notation," *Human-Computer Interaction,* vol. 7, pp. 1–45, 1992.
13. S.W. Hawking, *A Brief History of Time,* Toronto: Bantam, 1988.
14. J. Hintikka, *Knowledge and Belief,* Cornell University Press, Ithaca, 1962.
15. M.N. Hunks and L.M. Stephens, "Multiagent systems and societies of agents," in *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence,* G. Weiss ed., Cambridge/London, The MIT Press, 1999.
16. J. Kramer, S. Noronha, and J. Vergo, "A user-centered design approach to personalization," *ACM Communications,* vol. 43, no. 8, pp. 45–48, Aug. 2000.
17. D. Liu, M. Burgin, and W. Karplus, "Computer support system for aneurysm treatment," *Proc. of the 13th IEEE Symposium on Computer-Based Medical Systems,* Houston, Texas, pp. 13–18, June 2000.
18. B.A. Meyers, "A brief history of human computer interaction technology," *ACM interactions,* vol. 5, no. 2, pp. 44–54, Mar. 1998.
19. M. Minsky, "Commonsense-based interfaces," *ACM Communications,* vol. 43, no. 8, pp. 67–73, Aug. 2000.
20. S.J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach,* Prentice Hall, New Jersey, 1995.
21. H.T. Siegelmann, *Neural Networks and Analogy Computation,* Birkhauser, Boston, 1999.
22. I. Stewart, "The dynamics of impossible devices," *Nonlinear Science Today,* vol. 1, no. 4, pp. 8-9, 1991.