TECHNICAL REPORT

Department of Computer Science

TIME IN SPECIFICATIONS: WHAT EXACTLY ARE WE TRYING TO SPECIFY?

M.S.Kutar, C.Britton, S.Jones

Report No 329

March 1999

T|RN029

Time In Specifications: What Exactly Are We Trying to Specify?

M.S. Kutar, C. Britton and S. Jones
Department of Computer Science
University of Hertfordshire
College Lane, Hatfield, Herts, UK
AL10 9AB
Tel: 01707 285124 / 284354 / 284370
e-mail: M.S.1.Kutar, C.Britton, S.Jones@herts.ac.uk

Abstract

It is generally accepted that the specification of time can be a difficult goal to achieve. Much of the research in this area has concentrated on new methods which allow for the specification of time. However, little attention has been concentrated on the concept of time in specifications, and that in order to fully address the problem of specifying time we need to have a clearer idea of what we are trying to do. This paper attempts to fill that gap, and aims to discover what we are trying to achieve through the inclusion of time in a specification. It goes on to consider the role of representations of time in the area of human computer interaction and considers what is required of such representations in HCI.

1: Introduction

An important gap in the literature appears when one tries to discover the nature of what we are trying to specify when including time in a specification. Much of the literature concerning time in specifications is specifically addressed to 'real time' systems, which may be broadly defined as those systems where the correctness of the computations not only depends upon the logical correctness of the computation, but also on the time at which the result is produced. The greatest concentration of effort in this area is on the development of formalisms which aim to establish whether timing correctness may be achieved. There is little discussion, however, of what exactly is meant by timing correctness.

Additionally, developments in other areas of computer science, particularly human computer interaction, are causing the challenge of time to be recognised as an important topic, even for those systems which would not necessarily meet our broad definition above. Again, there seems to have been little attention focused on defining the nature of the concept we are attempting to represent, although given that the topic of time is still in its infancy in these areas it is perhaps a little more understandable.

It is hoped that by going 'back to basics' on this problem, and attempting to define what exactly we are trying to represent when including time in a specification, that some light may be shed on how best we can address the complex issue of how best to represent time in a system specification. The rest of this paper follows two strands. Section two considers the concept of time in real time systems, and section three considers the concept of time in interaction.

2: Time in Real Time Systems

2.1 General Introduction

As we have seen above, real time systems can be broadly defined as those in which not only the logical correctness of a computation is considered when evaluating whether a system is meeting requirements, but also the timing correctness of that computation. This may be illustrated by considering a robot which must pick up an object from a conveyor belt. There is only a small window of time in which the robot can correctly perform its task - if it is too late or too early, then it will fail to pick up the object, even though it has performed the action correctly in every other respect. Thus a real time system is one in which an emphasis on real time is placed when defining the systems required behaviour.

Real time systems may be further classified as being either 'soft real time systems' or 'hard real time systems' [Nissanke 1997]. A hard real time system is one in which timing correctness is of critical importance and may not be sacrificed for other gains. In some (often safety critical) systems, timing correctness is of such importance that logical correctness requirements may sometimes be relaxed in order to achieve timing correctness. By contrast, a soft real time system is one where timing correctness is of great importance but is not critical. Consequently, under special circumstances, timing requirements may be relaxed.

A further definition of real time systems is given by Kurki-Suonio [Kurki-Suonio 1994], in which he defines real time systems as those where the desired real time properties cannot hold unless some logically possible computations are excluded, and he considers that this is perhaps the only type of system which should be considered to be truly real-time (This definition initially appears to be analogous to Nissanke's definition of 'hard' real time systems above, but on closer inspection it seems to introduce the idea that the logical correctness is *always* subordinate to timing correctness.) Kurki-Suonio's argument would lead to the conclusion that unless timing correctness is considered to be of greater importance than logical correctness then a system is not a real-time system, which does not appear to fit with the general consensus of opinion. In addition, it must be borne in mind that *all* systems, are to at least a small extent, ones in which time is of relevance, even if it is not of critical importance.

2.2 Abstraction

At its simplest level of abstraction real time may be seen as a simple ordering of events according to their precedence. The precedence relation may be either implicit or explicit. For example, in the case of inputs and out puts it would be implicit that an output will follow an input. In more complex situations the temporal precedence is explicit, although this does not imply that it must be explicitly stated.
Other forms of abstraction contain a more explicit notions of real time. The choice of abstractions of real time is quite wide, as evidenced from the many forms of temporal and real time temporal logics which have been developed (for example [Ostroff 1989], [Mok 1989], [Koymans 1989]). Three major forms of abstraction are founded on the following:
- an explicit notion of clock time.
- an implicit notion of real time through time intervals, relations between them and through durations.
- a qualitative notion of real time through modalities of time, as expressed by terms such as 'some time' and 'always'

[Nissanke 1997]

Everyday concepts of time with which we deal in every part of our lives, are, of course, in themselves abstractions, and indeed all of the forms of abstraction shown above are ones which we generally have no difficulty reasoning with in most situations. It seems strange therefore, that the inclusion of time in specifications causes sufficient difficulty that unless it is of sufficient importance to be considered critical to system correctness, it is usually ignored. This leads to the conclusion that the source of the problem may be less to do with the application of time in computer science, than the general lack of understanding which we have of the concept of time. Augustine's lament, "What then is time? If no one asks me, I know what it is. But if I want to explain it to someone, I do not know it." still holds true. Time is a concept of such familiarity that we can use it without serious thought, yet it is such an abstract concept that no satisfactory description has yet been found. Perhaps, we will struggle to apply it until such time as we gain a more concrete understanding of the concept itself.

2.3 Timing Correctness

Timing correctness, or timeliness is obviously highly dependent on the timing requirements of the application in hand. There are, however several global temporal properties of systems which will have a bearing on timing correctness. [Kurki-Suonio

1994] [Nissanke 1997]. Qualitative studies of real time systems concentrate on temporal properties such as liveness, fairness, eventuality and safety.

2.3.1 Liveness

Informally, this may be described as 'something good eventually happens'. This could be for example, the termination of a sequential computation. It is a temporal property which guarantees the eventual occurrence of individual events.

2.3.2 Fairness

Forbids the continual omission of an operation. It is a theoretical restriction on nondeterminism, which guarantees the occurrence of an event sufficiently frequently. What is an acceptable frequency would clearly need to be defined for each system.

2.3.3 Eventuality

This is simply that an event will eventually occur, and precludes the inclusion of events which will never take place.

2.3.4 Safety

Informally, 'something bad never happens'. This does not require the occurrence of events, merely the non-occurrence of undesired ones. In reality it may be taken to include events which have a very low probability of occurrence.

All of these temporal properties may be reasoned about within a logical framework based on temporal logic and concurrency. Underlying the notion of concurrency are the notions of maximal parallelism and interleaving models. Maximal parallelism is the assumption that events occur as soon as they are enabled, without constraints such as lack of resources. This allows for the simultaneous occurrence of events, which may therefore be only partially ordered. Interleaving breaks down this notion, requiring total ordering of events, and therefore removing the possibility of their simultaneous occurrence.

2.4 Time in Real Time Systems - Conclusions

We have seen that a real time system is one in which timing correctness is of at least equal importance to logical correctness, and have considered the qualitative notions of liveness, fairness, eventuality and safety. The major forms of abstraction have been outlined, and consequently we have some general notions of what qualities a real time system must have. Thus we may assume that a notation for the specification of real time systems should tie in with one of the major forms of abstraction, and also allow us to reason about the qualitative notions shown in section 2.3. However, more specifically quantitative requirements which might arise are not discussed (either here or in the literature) and so we have no concrete examples of what the timing requirements of a system might be, we know only that a timely response is required. Essentially we require that the system responds to some external event within a given time. It would seem reasonable that two forms of abstraction, an explicit notion of clock time and an implicit notion of real time through time intervals, relations between them and through durations, are the most important, as these allow for more specific requirements to be expressed than an abstraction based upon modalities of time.

3 Time in 'non real time' systems

3.1 Introduction

The growth in interactive and multimedia systems, has led to the concept of time becoming increasingly important in systems which are not considered to be truly real time. Interaction between the user and system has temporal properties, and these temporal properties of interaction have a great deal of relevance to usability issues. For example the systems potential for supporting different user strategies and the systems demand on user memory are significantly affected by timing, particularly system or user delay. Temporal properties of interaction are particularly noticeable in multimedia applications, distributed systems, and use of the world wide web. It should be borne in mind that whilst temporal properties have a bearing on usability issues for all applications to some extent, there are a number of applications, such as word processors and spreadsheets, where temporal properties are quite low down on the ladder of factors relating to usability.

In many interactive system development projects, the need for user understanding of representations is addressed by building a skeleton prototype to offer options which can then be refined in line with the user's requirements. A prototype facilitates communication between developers and clients or users, allows users to relate what they are shown to their own experience of tasks, and enables them to give meaningful feedback on the design ideas which are embodied in the prototype. There are, however, certain cases where the prototyping approach may not be feasible or not, on its own, adequate. For large complex systems of any kind the development of complete system prototypes is unlikely to be cost-effective, and for safety critical systems, development based on a prototype would be unlikely to gain the client's confidence. Even in development situations where prototyping appears to be desirable, problems can arise relating to the complexities of version control, or the amount of time that is needed for a client to evaluate the prototype [Britton & Doake 1996]. Where prototypes cannot be used, or are not enough on their own, one of the main alternatives is currently to use one or more representations constructed using languages designed for specifying software. At the requirements validation point in development, developers will have constructed representations of the client's understanding of their problem and any requirements they have for the system to be developed. These representations are now presented to the clients and users for feedback on whether or not their intentions have been correctly represented. This feedback can only be useful and meaningful if clients and users have a good understanding of the representations. An important point to consider therefore is the nature of notations which are used to represent time in interaction. Most of the formalisms which have been developed to represent time are not only textual, but also quite mathematical in nature. It is accepted that graphical representations can assist with understanding, [Britton & Jones 1998] particularly for inexperienced readers, and so it may be that the notion of time in interaction presents new challenges, namely the development of more graphical methods of representing time, or the graphical representation of existing notations.

3.2 Temporal Challenges of Interaction

Johnson [Johnson 1995] identifies a number of challenges presented by interaction. Two of these have particular relevance for distributed systems:

- how may we provide users with an adequate representation of the flow of information form remote sites? This is a challenge which has been taken on board in the design of web browsers such as Netscape, which give a user some indication of the rate of progress. However, it is less apparent in other areas.
- How can users be provided with sufficient information about other users and systems so that they can make accurate predictions about the changing state of interaction over time?

Both of these issues, if they are to be resolved, must be considered early on in the systems design, and incorporated into the requirements documentation. As with many usability issues however, they tend to be overlooked in favour of what are considered to be more performance related requirements. An additional problem in this area is the reluctance of developers to incorporate temporal requirements which they see as non-essential. There is evidence that outside the area of hard real time systems consideration of performance and response issues is lacking. [Lubars 1993]

A second area in which temporal challenges of interaction have been identified is that of multimedia systems. [Johnson 1995] Technological limitations, and particularly problems with speed of retrieval from remote sites, cause multimedia bottlenecks. This can have a considerable effect on the presentation of real-time media such as audio and video output. Real-time multimedia modalities are obviously tightly interwoven, but Johnson suggests that as well as considering technological solutions (increasing processor speeds, shortening transfer times etc.) we should consider ways in which the user may exercise more power over the bottleneck, by choosing for example, to sacrifice some video quality, in order to gain enhanced sound quality, or vice versa. Consequently the user could indicate a preference for a particular perceptual modality. Thus two further challenges he identifies are:

- how can designers represent the changing priorities that might be assigned to different modalities during interaction?
- how can these priorities be adequately related to the changing demands of particular user tasks?

3.2.1 Abstractions of Time in Interaction

The granularity of time in interaction is quite varied. It can include, for example, the consideration of the time which it will take the user to complete a particular task, as well as consideration of the time period in which a second click of the mouse will constitute a double click. As we have seen above, there are a number of different abstractions of time which may be used in temporal representations. Clock time is probably of less relevance in interaction given that the environment is so varied, which suggests that representations which use interval time may be more effective.

However, these may be seen to lack the precision needed at later stages in the development process. Thus an ideal solution would probably be a representation which allow more detail to be included as design progresses, or which may easily be translated to a more precise real time notation as required. Another choice which must be made is that between (more deterministic) linear time models, and (more nondeterministic) branching time models. The very nature of interaction suggests that the branching time models may be more useful, particularly in the early stages of design. However, whilst it is clear that these choices must be made, there is a lack of criteria which may be used to assist with the decision.

3.3 Two Timing

One difficulty with existing design models of timed interactive behaviours [Mezzanotte 1995] [Palanque 1995] is their lack of reference to either user strategy, or theories of human behaviour. If this difficulty could be overcome, Parker [Parker 1997] argues, a higher order model of users temporal perceptions and behaviours might be validated.

Dix [Dix 1987] has suggested that there may be no relation between the users appreciation of time and the actual execution times of internal machine events. He argues that clock-driven timings of system events should be separated from attention-driven timings of user perception. However, there has been little follow up to this idea and consequently, the notion of user perception seems to have been largely ignored. This appears to be an important idea which, if taken up, may provide dividends in the form of better usability design. It is unclear whether methods seen as suitable for the specification of real time systems will also be suitable in this context. Indeed there is so far insufficient information to judge whether abstractions and qualitative frameworks which are in use in the field of real time systems might be equally applicable in the field of interaction. This certainly warrants further research.

4 Conclusions

The main body of work on the specification of time has, understandably come from the world of real time systems. However, in this area the concentration is on new methods rather than the consideration of the fundamental temporal requirements which are to be represented. This may be because the abstractions and the qualitative notion of what constitutes timing correctness are sufficiently well founded. However it cannot be discounted that developments in the fundamentals may assist with solutions to the problems inherent in representing time.

An emerging research area is into the inclusion of time in representations of interaction. The ideas here are still very much in their infancy - indeed we are only now beginning to see consideration of this area. There appears to have been little consideration of what abstractions might be suitable for use in this area, nor any real definition of what needs to be achieved. Future work might include the consideration of the fundamentals in this area as well as whether representations of time in the area of real time systems are suitable for temporal representation of interaction.

References

Britton, C. & Doake, J. (1996). Software system development: A gentle introduction. McGraw-Hill.

Britton, C. and Jones, S. (1998) The Untrained Eye: How Languages for Software Specification Support Understanding in Readers Who are New to Them. To appear in *The Journal of Human Computer Interaction 1999*

Dix, A.J. (1987) The Myth of the infinitely Fast Machine. In *People and Computers III* Diaper and Winder (Eds.) CUP

Johnson, C.W. (1995) The Challenge of Time. In The Design, Specification and Verification of Interactive Systems. Palanque, P. and Bastide, R. (Eds) Springer-Verlag 1995

Koymans, R. (1989) Specifying Real-Time Properties with Metric Temporal Logic. Research Report, Philips Research Laboratories, Eindhoven, October 1989

Kurki-Suonio, R., (1994) Real-Time: Further Misconceptions (or Half-Truths). In *IEEE Computer* June 1994, pp 71-76

Lubars, Potts, C. and Richter. (1993) A Review of the State of Practice in Requirements Modelling. In *Proceedings IEEE Int Symposium on Requirements Engineering, RE93, pp2-14.* IEEE Press 1993

Mezzanotte, M., Paterno, F. (1995)Including Time in the Notion of the Interactor. In *Papers From a Workshop on Temporal Aspects of Usability.* Johnson, C. and Gray, P. (Eds.) University of Glasgow, Dept of Computer Science, GIST Technical Report G-95-1.

Mok, A.K., (1989) Programming and Scheduling in Real Time Logic. In *Real Time Systems, Proceedings of the Joint University of Newcastle Upon Tyne / International Computers Limited Seminar* (Newcastle, September 1989)

Nissanke, N., (1997) Realtime Systems. Prentice Hall

Ostroff, J.S., (1989) Mechanising the Verification of Real-Time Discrete Systems. *Microprocessing and Microprogramming* 27, 1989, pp 649-656

Palanque, P. and Bastide, R. (1995) Time Modelling in Petri Nets for the design of Interactive Systems. In *Papers From a Workshop on Temporal Aspects of Usability.* Johnson, C. and Gray, P. (Eds.) University of Glasgow, Dept of Computer Science, GIST Technical Report G-95-1.

Parker, H., Roast, C.R. and Siddiqi. J.I. (1997) Towards a framework for investigating temporal properties in interaction. SIGCHI Bulletin, 29(1), January 1997. Available on-line at:
http://homepages.shu.ac.uk/%7Ecmscrr/PAPERS/ucre96.pdf.