TECHNICAL REPORT

Dept of Computer Science

On The Representation of Time In Interaction

Maria Kutar, Carol Britton and Sara Jones

Technical Report No 326

Feb 1999

On The Representation Of Time In Interaction

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

The growth in interactive, distributed and multimedia systems and particularly the explosion in use of the World Wide Web means that the concept of time is being recognised as of increasing importance in systems which are not considered to be truly real time in the traditional sense. We survey the literature relating to the role of time in interaction and consider how representations of time in interaction might encourage more effective interface design, if temporal properties can be incorporated into formal system specifications. Abstractions of time and the formal methods currently used to represent time in the realm of real time systems are discussed. The potential use of formal methods in representing temporal aspects of interaction is considered, although the fact that the temporal challenges of interaction are not yet fully understood hinders us in stating with any certainty that specific methods may be effectively applied in interaction.

1 Introduction

The growth in interactive, distributed and multimedia systems, particularly the explosion in use of the World Wide Web, means that the concept of time is becoming increasingly important in systems which are not considered to be truly real time in the traditional sense. This paper surveys the literature relating to the role of time in interaction and considers how representations of time in interaction might encourage more effective interface design if temporal properties can be incorporated into formal system specifications. We discuss the formal methods currently used to represent and specify time in the realm of real time systems, and consider how these techniques may be applied to new paradigms in interaction. The paper is structured as follows: section two discusses the role of time in interaction, section three, abstractions of time. Section four surveys formal methods used in the specification of real time systems, section five considers the potential application of formal methods, to incorporate temporal properties of interaction into system specification, and section six presents our conclusions.

2 Time and Interaction

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 system's potential for supporting different user strategies and the system's 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. A small pilot study [Byrne & Picking 97] surveyed and analysed the issues which Web users regard to be of central importance. The results indicated that users consider temporal issues to be intrinsic to usability.

The granularity of time in interaction is quite varied. It can include, for example, the consideration of the time period in which a second click of the mouse will constitute a double click, as well as consideration of the time which it will take the user to complete a particular task. In addition to this, a different type of temporal quality, pace, has been argued to have a (possibly more important) bearing on the users' experience of interaction [Dix 92]. During interaction, channels of communication are used not at a constant rate but intermittently. Consequently, bandwidth, which assumes continuous transmission, is not necessarily an appropriate measure of communication. Pace, the measure of the rate at which individual communications occur through a channel is proposed as a primary property. This is not so as to dismiss other properties such as granularity, nature of the medium (auditory/visual etc.) but it is an important issue with regard to interaction which is both measurable and quantifiable and may therefore provide a useful way of relating the pace, or potential pace of the channels to the tasks the user must carry out. For example, when analysing the rate at which two individuals exchange emails, the pace of their interaction could be defined as the rate at which individual mail messages are produced. This would give us a somewhat different insight into the temporal properties of that relationship than if we were to consider, for example bandwidth which assumes continuos transmission and could therefore tell us only about the average transmission of messages over a given period.

Related to the notion of pace is the question of what time-scales humans can comfortably work to. It has been suggested [Dix 96a] that our experience of whether timings are good or bad are affected by a combination of psycho-motor abilities and external stimuli. Rhythms are easier to deal with than occasional delays and it would seem that prediction is easier if we are dealing with regular time intervals. This means that regularity may be more relevant to the users experience than absolute response time intervals. Thus a slower but consistent interface may be more effective than a generally fast but inconsistent one.

If the pace of interaction is too slow the user's execution/evaluation loop is broken [Dix 94a]. The user generally expects feedback, informing him of what has happened, but with a very slow pace of interaction it may be that nothing has yet happened. Thus, once a response finally arrives, the user must recall the appropriate context. An alternative scenario is that a response fails to arrive at all, requiring the user to both recognise that it has failed to arrive and take appropriate action. Feedback is often delayed in open and cooperative systems. This obviously causes pace to slow and the user must therefore devise strategies to overcome the difficulties outlined. In order to provide alternatives these strategies must be identified and considered during interface design. Effective interface design must not only identify current user solutions for dealing with such problems, in order that new support strategies do not interfere with those (possibly subconscious) strategies currently in use, but also introduce new support systems to assist the user in overcoming these problems.

Long-term interaction, where the rate of turnaround of individual messages may take hours, days or weeks, poses problems which are different from those found in higher pace interaction [Dix et al 98]. The user may have difficulty in recalling the context of a delayed response (action-effect gap), may forget to act themselves if they cannot react instantly to a request (stimulus-response gap) or, if an expected external response is not forthcoming, the whole interactive process may break down (missing stimulus). A recurrent pattern of activity is identified - request, receipt, response, release (the four R's) This poses system design issues of how to assist the user with coping strategies, particularly when automating a functioning paper-based system.

Dix [Dix 87] has suggested that there may be no relation between the user's appreciation of time and the actual execution times of internal machine events. A formal analysis and description of typical, steady, system behaviour is used to relate the steady state functionality, which is described in most system specifications, to the actual temporal behaviour described by the user. The temporal behaviour of the interface is viewed separately from the actual temporal behaviour of system computation, a technique referred to as 'two-timing'. The notion that increased computational power is all that is required to improve temporal qualities of interaction is questioned and indeed in a further paper [Dix94b] it is noted that hardware insufficiencies have largely been overcome by advances in technology only to for us to be faced with similar temporal difficulties as a result of the increase in networking and its associated delays. The fact that, in the user's view, the hardware hurdle has been overcome simply to be replaced by a network hurdle causing similar usability problems, emphasises the point that the user experience should be separated from hardware issues.

Another way in which the interface may be affected by temporal issues is the area of status-status mappings [Dix 96b]. Status is defined as those things in an interface which have a constantly available value. Examples include screen contents or mouse position. Status-status mappings are constraints or functional dependencies between different status within a system, such as the relationship between a document and the screen image of it within a word processor package, or the way in which an icon being dragged across the screen follows the mouse movement. It is argued that these mappings can be identified during the specification stage of interactive systems and thus highlight potential temporal problems in the interface. Without both infinitely fast computation and infinitely fast communication, constraints between status in the interface are bound to be violated on some occasions. This will occur because different kinds of event and agent mediate changes in status and so constraints tend to be relaxed as development progresses. The violation may simply take the form of a slight lag between the source of a change and its display, or may at times lead to inconsistency between parts of the interface. No easy solutions are offered, but it is hoped that by allowing the early identification of such potential anomalies, designers may have greater control over the external behaviour of the system and thus can attempt to avoid the problems identified. There are two potential solutions to the problems which may be presented by status-status mappings. The first is technological, namely that faster and faster computation will eventually mean that any anomalies are not apparent to the user. The second is a design-oriented solution, namely to design the system in such a way that the inevitable delays are of no consequence to the user.

Johnson [Johnson 95] identifies a number of temporal 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 from 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 system development, 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 et al 93]

A second area in which temporal challenges of interaction have been identified is that of multimedia systems [Johnson 95]. 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?

In can be seen, therefore that temporal issues affect the interactive process in a number of different ways. Temporal properties such as pace and rhythm may have a greater bearing on the users interactive experience than simply speed of response and therefore it can be seen that temporal challenges in interaction may be quite different to temporal challenges in other areas of computer science. There appears to be an underlying theme however, that these issues must be identified at an early stage in the design process if solutions are to be provided to the problems that they can cause. It would seem sensible therefore to look to existing formal methods which allow time to be specified, and where possible, use these methods to specify temporal aspects of the interface.

The experience of the past decade whereby hardware insufficiencies have largely been overcome by advances in technology only to for us to be faced with similar temporal difficulties as a result of the increase in networking and its associated delays indicates that it would be wise to consider temporal properties of interaction separately from internal system performance.

3 Abstractions of time

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 outputs it would be implicit that an output will follow an input. In more complex situations the temporal precedence may be made explicit, although this does not imply that it must be explicitly stated. Other forms of abstraction contain a more explicit notion 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 89], [Mok 89], [Koymans 89]). 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 97)

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.

4 Formal Methods and the Specification of Time

A number of formalisms have been developed in order to incorporate time in systems specification. This section briefly summarises the available methods as presented by Ostroff [Ostroff 92]. These methods have generally been developed for use in the development of real time systems. Real time systems may 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 system's required behaviour. Notations for the specification of time have been developed with the aim of allowing developers to prove that timing correctness can be achieved, and therefore the emphasis is on formalisms with well defined semantics.

4.1 Structured Methods and Graphical Languages

Structured Methods [Hatley & Pirbhai 88] [Ward & Mellor 85] give a structured set of system requirements, incorporating various views such as data flow, control flow, a requirements dictionary. Timing requirements are represented in a table of response times which lists incoming and outgoing events alongside their expected repetition rates and response times. This means that timing requirements are seen somewhat separately from other requirements. Other problem areas include the lack of formal semantics, lack of support for formal verification and an inability to model nondeterministic behaviour.

Statecharts [Harel 87] are seen as an improvement to structured methods. The normal state transition diagram is enhanced with hierarchical and compositional features. A graphical tool, Statemate [Harel et al 90] is formally based and provides automated simulation for execution of the model.

Object-Oriented Methods such as Real-Time-Object-Oriented-Modelling (ROOM) [Selic et al 94]provide an object-oriented variation of the statecharts formalism for distributed real-time systems. The Unified Modeling Language (UML), and modeling constructs originally developed for (ROOM), have been combined into UML for Real-Time. [Lyons 98]

Petri Net theory [Peterson 81] deals with concurrency, nondeterminism and causal connections between events. Properties such as liveness, safety, freedom from deadlock and boundedness can be checked. (For a full discussion of these properties see [Kurki-Suonio 94]).There exist a number of timed extensions to Petri Nets such as Timed Petri Nets [Ramchandani 74]which allow real time to be incorporated into the model.

4.2 Logics and Algebras

These provide the most abstract approach to system specification and analysis. Examples include the timed transitions models / real-time temporal logic framework (TTM/RTTL) [Ostroff 85] and timed Communicating Sequential Processes [Reed & Roscoe 91]. They generally contain a number of elements: a high level formal specification language to be used to specify required behaviour, a proof system to establish the correctness of the system relative to the specification and in some cases a set of heuristics and guidelines for use with larger systems. Temporal logics and process algebras may both be endowed with discrete or dense time semantics but the two vary in their syntactic specification styles.

Temporal semantics may be interval (based on intervals of time which may be used to represent finite chunks of system behaviour), or point (whereby temporal behaviour is described with respect to a defined reference point in time). Point semantics may be further classified as either linear, where each moment has only a single possible future corresponding to the history of the development of the system, or branching, where time has a tree-like structure which allows time to split into alternative courses which represent different choices made by a system. Real-time temporal logics allow quantitative properties such as periodicity, delays and real-time response to be expressed. For a fuller discussion and detailed examples of the different temporal logics and process algebras the reader is directed to [Ostroff 92]

Temporal logic is useful for the description of 'whole system' properties such as safety, fairness, liveness and real-time response, but has a relatively unstructured style in comparison with process algebras. As yet there does not appear to be a language which combines the best elements of each approach.

5 Temporal Representations and Interaction

We have seen in section two above that temporal issues provide a number of challenges in interaction. These challenges have a somewhat different emphasis to those in real-time systems. In real-time the aim has generally been to prove both timing and system correctness at an early stage in the design process. In interaction, however this is likely to be of lesser importance, particularly if we were to use the notion of 'two-timing' whereby temporal issues of interaction are considered separately from performance related temporal issues. In addition, the temporal properties which are of relevance in interaction may be quite different to those found in other areas. Absolute response time may be less important than consistency and

pace for example and thus the requirements for the coverage of a notation may differ also to those in, for example real time computing. In some areas, it may be possible to provide assistance to the user merely through the provision of sufficient temporal information, allowing the user to make informed decisions about his most appropriate form of action when presented with delays.
There appear to be two main areas in which the representation of time and the use of formal methods may be of relevance in interaction. Initially it may be that temporal modelling of interaction may allow us to gain a greater understanding of how temporal issues affect the interactive process. For example it may be possible to develop a temporal model which encompasses the whole interactive process, from very small to very large scale which would allow us to consider the combination of time and interaction as a whole rather than piecemeal. Secondly, it may be possible to improve the quality of interaction by considering temporal issues at the specification stage of design, in order that the design fully addresses the difficulties they may cause. This is of particular relevance in systems where delays arise in areas which may be out of the control of the designer. Thus, although the designer may be unable to prevent the cause of the difficulties he may be able to provide the user with more temporal information or coping strategies.

This presents a challenge to designers to both recognise these potential problems and to produce interfaces which either eliminate the problems or provide the user with coping strategies. In the case of delays which result from networking difficulties that are obviously out of the control of the interface designer, it would be sensible to provide the user with as much information as possible which would allow him to decide on the most appropriate course of action. For example the web browser Netscape provides a small amount of information regarding the percentage of data so far retrieved, but it does not give any indication or estimation of how long it will take to retrieve and display the full page. If these issues are not considered early in the design phase then it is unlikely that they will be satisfactorily resolved.

It would seem reasonable therefore that the inclusion of temporal information relating to interaction at the specification stage may assist designers in overcoming temporal problems. However, in order to do this, the method used to produce the specification must be capable of representing time in an appropriate manner. The consideration of temporal issues and interaction is still in its infancy, to the extent that there has been no real investigation of the abstractions of time which are most relevant to interaction. In order to include time in specifications of the interface, the methods used must be able to express the abstractions of time which are most appropriate in interaction. As was seen in section three above, there are a number of different abstractions of time which may be used in temporal representations. Clock time may be of less relevance in interaction given that the environment is so varied, which suggests that representations which use interval time may be more effective. In addition interval based notations may be more suited to the representation of temporal qualities such as pace. However, notations which use interval based semantics may be seen to lack the precision needed at later stages in the development process. Thus an ideal solution would possibly be a representation which allows 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 remains a lack of criteria which may be used to assist with the decision. We have seen in section four above that a wide variety of formalisms are available to represent and reason about time. As these have mainly been developed in response to the requirements of real time computing the emphasis has been on the representation of response time and the ability to prove timing correctness of the system. This does not mean that these methods will be unsuited to the demands of interaction, but until we have a clearer idea of what notations used to represent time in interaction must do it is difficult to state with any certainty that suitable methods have yet been developed. The rich variety of temporal logics and process algebras suggest that it may well be possible to meet the challenges presented by interaction but clearly further research is needed. It may be that in order to find methods which encompass the temporal paradigms in interaction, we need to look to other disciplines where time is specified. For example, Product/Process Modelling [Britton et al 98] is used in the field of quality assurance systems and incorporates temporal properties based on a well defined semantics.

Interaction is an area where user validation of requirements is particularly important. One difficulty posed by the inclusion of time into a formal specification is that temporal representations are often logically or mathematically based, leading to difficulties in understanding by the user. 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. With large complex systems of any kind, for example, the development of complete system prototypes is unlikely to be cost-effective. 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 96].

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. It would seem reasonable therefore, that methods which allow representation of temporal properties could be used in these situations, allowing temporal requirements to be validated as well as functional requirements. An important point to consider therefore is the nature of methods 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 logical or mathematical in nature. It is accepted that graphical representations can assist with understanding, [Britton & Jones 98] 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 such as that suggested for CSP [Kutar 98].

6 Conclusions

Temporal properties of interaction are being recognised as having a significant impact on the quality of interaction. However this is very much an emerging research area and thus the temporal challenges of interaction are only beginning to be defined. It seems that the way in which temporal issues affect interaction may be viewed separately form performance related issues and that we may in fact find them to be quite different to the temporal problems found in other areas, with consistency and pace being more relevant than absolute response times. This means that when seeking to represent temporal aspects of interaction the methods used may need to possess quite different properties than are expected in other areas. Both the abstractions of time and the use of the representation may differ, but the wide variety of notations currently available in the field of real time computing provide a rich seam from which we may pick. The most pressing research challenge is, however, to define more clearly what we are seeking to represent in interaction. Only once the goals are more clearly defined will it be possible to see whether existing formal methods are adequate.

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*

Britton, C., Kaposi, A.A. and Myers, M. (1998) Models with Measures: "Project Control for 2000 and Beyond" In Kusters, R., Cowderoy, A., Heemstra, F. and Trienekens, J. (Eds.) Proceedings of the combined 9th European Software Control and Metrics Conference and the 5th conference for the European Network of Clubs for Reliability and Safety of Software, Shaker Publishing 1998

Byrne, A. and Picking, R. (1997) Is Time Out to be the Big Issue? Presented at *Time and the Web, Staffordshire University 19 June 1997.* Available at: http://www.soc.staffs.ac.uk/seminars/web97/papers/picking.html

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

Dix, A.J. (1992) Pace and Interaction. In *People and Computers VII.* Monk, Diaper and Harrison (Eds) CUP

Dix, A.J. (1994a). Que Sera Sera - The Problem of Future Perfect in Open and Cooperative Systems. In *People and Computers IX.* Draper and Weir (Eds) CUP

Dix, A.J. (1994b) Seven Years On, The Myth Continues. University of Huddersfield Research Report RR9405

Dix, A.J. (1996a) Natural Time. Position Paper for *CHI 96 Basic Research Symposium, April 1996, Vancouver, BC.* Available at: http://www.soc.staffs.ac.uk/~cmtajd/papers/natural/natural.html

Dix, A.J. (1996b) Temporal Aspects of Usability. Delays and temporal Incoherence Due to Mediated Status-Status Mappings. In *SIGCHI Bulletin, Vol 28 No 3, April 1996*
Available at: http://www1.acm.org:82/sigs/sigchi/bulletin/1996.2/Alan-Dix.html

Dix, A.J., Ramduny, D. and Wilkinson, J. (1998) Interaction in the Large. In *Interacting With Computers, Special Issue on Temporal Aspects of Usability* (in press)

Harel, D. (1987) Statecharts: A Visual Formalism Fort Complex Systems. In *Sci. Comp. Progr.* 8, 231-274 (1987)

Harel, D., Lachover, H., Naamad, A., Pneuli, A., Polti, M., Sherman, R. and Trachtenbroot. Statemate: A Working Environment for the Development of Complex

Reactive Systems. In *IEEE Transactions on Software Engineering* 16 pp 403-414 (1990)

Hatley, D.J. and Pirbhai. (1988) Strategies for Real-time System Specification. Dorset House Publishing Co, New York 1988

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

Kutar, M. (1998) A Graphical Representation for Communicating Sequential Processes. In *Proceedings of Workshop on Formal Aspects of Human Computer Interaction*. Sheffield Hallam University, Sheffield, UK. 5-6 September 1998

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, pp 2-14.* IEEE Press 1993

Lyons, A. (1998) UML for Real-Time Overview. Available at http://www.objectime.com/otl/technical/umlrt.html

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. (1985) A Temporal Logic Approach to Real Time Control. In *Proceedings of the 24th Annual Symposium on Decision and Control 1985.*

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

Ostroff, J.S. (1992) Formal Methods for the Specification and Design of Real-Time Safety Critical Systems. In *Journal of Systems and Software 1992 18 pp 33-60*

Peterson, J.L. (1981) Petri Net Theory and the Modelling of Systems. Prentice-Hall 1981

Reed, G.M. and Roscoe, A.W. (1991) Timed CSP: Theory and Practice. In de Bakker, J.W., Huizing, C., de Roever, W.P., Rozenberg, G. (Eds) Real-Time: Theory in Practice. LNCS 600, Springer-Verlag 1991

Ramchandani, C. (1974) Analysis of Asynchronous Concurrent Systems by Timed Petri Nets, Technical Report MAC TR 120 Massachusetts Institute of Technology 1974

Selic, B., Gullekson, G. and Ward, P.T. (1994) Real Tome Object Oriented Modelling. Wiley 1994

Ward, P. and Mellor, (1985) S. Structural Development for Real Time Systems. Yourdon Press, New York 1985