The Two Real-Time Solitudes: computerized control and telecommunications

Paul Freedman Daniel Gaudreau Raouf Boutaba
Ahmed Mehaoua

Centre de recherche informatique de Montréal (CRIM)
1801 avenue McGill College, bureau 800
Montréal (Québec) CANADA H3A 2N4

Abstract

While the two solitudes of our title, computerized control and telecommunications, are both concerned with computerized solutions to real world problems, we suggest that they are really addressing different needs and are therefore naturally preoccupied by slightly different concerns. Here we present a taxonomy of key elements intended to make such differences explicit in order to promote more mutual understanding and more collaborative work where concerns are shared. In particular, we suggest that one key difference lies in the nature of the timing requirements associated with reactivity, and whether they are considered as extensions to logical correctness or performance.

1 Introduction

While the two solitudes of our title, computerized control and telecommunications, are both concerned with computerized solutions to real world problems, we suggest that they are really addressing different needs and are therefore naturally preoccupied by slightly different concerns. And because this is poorly understood, practitioners in both camps sometimes appear to be needlessly castigating each other about real-time *truths* e.g. [10]. Other misunderstandings are apparent in the recent (March/April 1996) thread in the Usenet newsgroup *comp.realtime* about the 'twilight zone' of real-time.

Here we present a taxonomy of key elements intended to to make such differences *explicit* in order to promote more mutual understanding and more collaborative work where concerns are shared.

In particular, we note that modern telecommunications networks, especially over the last two or three years, are best characterised by *two* real-time contexts: (i) the service provided by the network (e.g. message passing throughput) and (ii) the automated (closed-loop) network management. As we shall see, the differences between computerized control and telecommunications are most apparent in terms of (i), while there is increasing common ground in terms of (ii). We acknowledge that in order to make our comparison clearer, certain differences may appear to be overstated.

2 A taxonomy of differences

2.1 The importance of physics

Computerized control systems are fundamentally designed to impose some kind of desired behavior upon *physical* phenomena, i.e. the operating 'environment' is physical in the sense that physical objects must be made to perform in a required way e.g. making a robot arm follow a specified trajectory, mixing chemicals in specified proportions.

By subjecting the physics of such phenomena (e.g. kinematics and dynamics of motion) to mathematical modelling and analysis, we may obtain the specification of a feedback controller whose computerized implementation typically takes the form of interactions with the physical environment (called "process" or "plant" by control system engineers); here the operator acts in a supervisory capacity, to oversee the behavior of the control. More precisely, the embedded computer is called upon to generate outputs, i.e. actuator commands and operator displays, as a function of input from the operator (e.g. operating mode, control setpoint) and changing environmental conditions (e.g. sensor data), based on a feedback control law. It is this combination of sensing, numerical calculations, and actuation which is typically called "closed-loop" (feedback) control. More generally, the operating environment exhibits subparts with different dynamics and as a result, the feedback controller typically consists of multiple control loops executing at different frequencies, e.g. computations associated with feedback paths are performed more frequently than those associated with feed-forward paths (used to update the values of parameters in plant models).

In contrast, the operating environment of telecommunications systems is largely 'artificial', i.e. non-physical, composed of artificial objects, i.e. messages containing sound, video, and data. Stated differently, telecommunication systems deal primarily with logical phenomena e.g. sending a message. (Of course, when concerns about reliability are present, computerized control and telecommunications systems must both also deal with "physical" equipment failure, buffer overflow.)

0-8186-7614-0/96 $5.00 © 1996 IEEE

2.2 The nature of temporal requirements

Since computerized control systems must deal with physical phenomena, there is a particular need to predict and analyze system behavior, and especially temporal behavior, as early as possible in software development [5]. In particular, many software-related problems are often traced to problems with data 'timeliness', e.g. data arrives too early, data arrives too late, new data overwrites 'old' still waiting to be processed [11].

Indeed, the control theoretic analyses behind the definition of a feedback control law for computerized control, are typically performed within the context of *sampled data* systems whereby true continuous time phenomena are modelled in terms of discrete time variables which take on values at precise time instants. As a result, the performance of the embedded control system, as measured by the difference between desired and actual environmental behavior, is typically associated, in part, with the accuracy with which this sampled data context can be actualized [14]. In contrast, it's less clear how to measure the improvement in the 'usability' of video-conferencing as the transmission delay is decreased. More generally, 'real-time' in a telecommunications context is often measured with respect to human perception (sight, sound), i.e. is the video sequence smooth enough, does the service provide a dial-tone soon enough, etc.

Note the distinction here between timing requirements of the kind "if event E occurs, then the system must react with a maximum delay of D time units" versus "the sensor must be read every T time units in order to ensure that the sampled data assumptions made as part of the design of the (discrete time) feedback controller will be respected".

In telecommunications systems, a kind of closed-loop control is also present for both operating and managing the network. The control function is commonly ensured by hard-coded real-time algorithms with a response time on the order of microseconds, while management functions are often initiated by operators where automated management tools are not provided [1] and are thus inherently orders of magnitude slower.

More generally, the closed-loop control associated with both computerized control and telecommunications systems typically requires that state information be collected, often at different time scales and for operating modes. The collection of such state information may be performed in a time-driven (periodic) way e.g. every 10msec, or every 128 messages, or in an interrupt-driven way e.g. in response to the detection of an alarm.

Observe that timing requirements at the system level of the kind "once the operator pushes the button, all of the doors in all of the subway cars must open within 5 seconds" describe bounds placed on the responsiveness of the computerized system in order to make its behavior more predictable. *Such timing requirements associated with reactivity are considered as extensions to logical correctness and not performance.* Indeed, such timing requirements are never phrased in terms of 'fast enough' or in probabilistic ways e.g. a dial-tone must be provided 90% of the time within 100msec[1].

As for the adjectives 'hard' vs. 'soft', they simply describe the *consequences* of not respecting a temporal constraint and not the nature of the temporal constraint itself. However, it is true that many temporal constraints associated with computerized control systems are hard in the sense that human life might be put at risk; this is rarely the case with telecommunications systems. Still, whether constraints are described as hard or soft, there may be large financial penalties incurred.

2.3 The importance of temporal determinism

In order to address the need to predict temporal behavior, the *design* of computerized control systems tend to promote temporal *determinism*. More specifically, such determinism typically takes the form of interactions with the operating environment defined in a *periodic* way, i.e. sensors are read and commands are sent to actuators in a periodic way, where the period is largely determined by the dynamics of the physical phenomena to be controlled in the operating environment. In contrast, sporadic operator intervention to change operating modes, for example, is also often cast in a periodic light by polling for operator input in a periodic way. However, the reactivity associated with detecting and processing alarms, for example, is typically assured by associating (prioritized) interrupts with such events, which are then trapped by the underlying real-time processing system.

As a result, the information processing associated with computerized control systems is primarily time-driven, whereas that of telecommunications systems is primarily interrupt-driven, i.e. responding to the arrival of a new message (to be routed).

(Of course, at the operating system level, both become signals and in this way may be mixed, with varying priorities.)

In contrast, current evolutionary trends in computing to support telecommunications illustrated by the increasing importance of distributed multimedia applications [7] and by the emergence of interactive user interfaces based on digital audio-video, require that new generation computer communication networks also offer time-bounded real-time services.

In particular, the main requirements to be met by a communication infrastructure in order to efficiently support real-time applications are bounded channel access delay, bounded message delay and bounded delay jitter e.g. an upper bound on the value of the delay associated with the transmission of a byte (on the order of microseconds) or of a message (on the order of milliseconds). Additional requirements are typically related to the communications mechanism such as the percentage of message loss, the message trans-

[1] A 'strong probability' that an emergency brake system will be activated once an alarm condition is detected, within 100msec, is obviously not a good enough timing requirement if a train system must be certified as being safe for passengers.

mission rate, the deadline miss percentage, the effective channel utilization, and the scalability of these mechanisms.

Such constraints are therefore intended to address network resource optimisation and guarantees about performance, in terms of information flow [2].

Thus, the new challenges faced by the telecommunications community are closely akin to those faced by computerized control community, i.e. the integration of temporal specifications into data and control flows. Indeed, the development of new generation networks require the prediction, as early as possible, of the dynamic behavior of lower service layers in order to guarantee deterministic performance to upper-laying applications[2].

2.4 The nature of multi-tasking

In practice, the computing architectures associated with both computerized control and telecommunications systems often take the form of communicating computing tasks[3]. In particular, the computing architecture of a computerized control system typically 'mirrors' the physical elements present, i.e. where there is a sensor, there is a sensing software element (module). Here multi-tasking means cooperation among multiple threads of execution (reading a sensor, performing a calculation, driving an actuator, etc.). Note too that inter-task communication (as opposed to inter-task synchronisation) is often asynchronous, since the various threads are often designed to respect different temporal constraints.

In contrast, multi-tasking in a telecommunications context often means passing control from task to task, with inter-task communication taking the form of (synchronising) remote procedure calls. In addition, the definition of the real-time tasks typically requires additional guidance in the form of structuring criteria in order to optimally partition conceptual units (e.g. objects) as part of the design process [5]. Stated differently, "in [computerized] control systems, a more symmetrical viewpoint is often useful in which active objects exchange messages rather than yielding control via [remote procedure] calls" (p. 9) [12].

As for runtime concerns, priorities are often assigned according to the messages in transit, rather than to the tasks themselves, as is typically the case in computerized control.

[2]That's why, some media access protocols as the Fiber Distributed Data interface (FDDI) for LAN environment, and the Synchronous Optical Network (SONET) protocol related to WAN area, have been specifically designed to provide not only excellent transmission performance (e.g. several hundred Mbytes/second) but also bounded delays (e.g. every 125 microseconds, a new byte is processed).

[3]We shall use the term 'task' to refer to a minimal unit of computing concurrency; in practice, a task either becomes a 'process' with a distinct address space or a 'thread' sharing an address space within a process with other threads.

2.5 Dealing with concerns about reliability

Concerns about *reliability* are typically reflected in computerized control systems by carefully engineered combinations of software and hardware redundancy, so that control behavior may 'gracefully' degrade. Since computing architectures for computerized control systems are almost always statically defined, task priorities are also statically defined, making early schedulability analysis possible e.g. [9].

In contrast, reliability in telecommunications is, *in addition*, reflected as adaptability or re-configurability or more generally, as fault tolerance, whereby some subsystems are engineered to be able to assume the functions of other subsystems. The former property refers to the ability of the communication system to continue to operate effectively in the presence of network traffic variations and temporary network overloading. The latter property refers to the protocol ability to survive communication channels failures such as those that can be caused by a noisy channel. As a result, computing architectures may change giving rise to the creation (and destruction) of new nodes (or new objects). This, in turn, means that scheduling policies are typically dynamic, i.e. task priorities are also subject to change.

2.6 The importance of third party products

Computerized control systems typically call upon third party products for computing infrastructure: hardware elements e.g. backplane buses such as STD-32 and VME; networking elements for physically distributed subsystems e.g. IEEE token bus, ArcNet, CAN; real-time operating systems e.g. POSIX.4 'conformant' products such as VxWorks, QNX, and LynxOs. As a result, timing issues are addressed in terms of software development and the careful selection of infrastructure elements (guided by corporate constraints).

In contrast, the development of new generation systems in the telecommunications industry is typically a combined hardware/software effort since the economies of scale make it possible to study subtle hardware/software trade-offs and VLSI fabrication. Nonetheless, there is some evidence to suggest that this too is changing as third party products such as real-time operating systems and microprocessor elements gain increasing favor.

2.7 New generation tool support for system/software development

In order to respond to the increasingly sophisticated demands of clients in terms of safety, reliability, and performance, there is increasing interest in both the computerized control and telecommunications industries in new generation (CASE) tool support for system/software development.

At the same time, 'computer-based symbolic specification tools' [8] intended to facilitate detailed software design have continued to evolve to address more phases of the software development process. In addition, new generation tools make possible more than just rapid prototyping and simulation.

Our recent examination of the state-of-the-art [3] has suggested that the most comprehensive of the new generation tools, i.e. those integrated software development environments which provide the greatest amount of support and automation (code generation) capabilities, were born in the telecommunications world. New products make it possible to translate an executable model of hierarchically organised software elements defined in an object-oriented way into a real-time executable tailored for a particular third party real-time operating system.

However, our experience to date [4] with one such product, the ObjecTime Toolset from ObjecTime designed to support the ROOM method [13], suggest that those formalisms developed by the telecommunications world such as Message Sequence Charts for describing protocols, are not always useful e.g. describing the checking of parameter values within a complicated control/status message packet [6]. In addition, the association of priorities with messages instead of tasks is testimony to the primarily event-driven nature of most telecommunications applications.

3 Conclusions

In this paper, we have tried to demonstrate the extent to which both telecommunication systems and computerized control systems are both "real-time" and yet different.

In particular, since most computerized control systems are physically distributed e.g. a train consisting of many cars, a manufacturing facility consisting of many integrated workcells, the way in which data and control information (in the form of messages) circulates (speed, reliability, etc.) is of critical importance. Such real-time concerns about networking are of equal concern in telecommunications, where lower layers must provide increasingly time-bounded services to upper layers.

The new federating vision of computerized control systems and underlying telecommunication infrastructures is confirmed by the organizational evolution of current real-time control systems for which the transmission system becomes an unavoidable and key component e.g. new generation air traffic control featuring partially automated dialogue between the control tower and the aircraft [6].

We believe that the integration of these two complementary fields at different levels is irreversible, and hope that this position paper has helped to clarify some of the important issues.

The authors gratefully acknowledge the financial support of Bombardier's Transportation Equipment Group and the Natural Sciences and Engineering Research Council of Canada.

References

[1] R. Boutaba and A. Mehaoua. Applying the policy concept to the management of atm networks. In *IEEE Systems Management Workshop*, Toronto, Ontario, June 1996.

[2] D. Ferrari. Guaranteeing performance for real-time communication in wide-area networks. Technical report, Departement of Electrical Engineering and Computer Science, University of California – Berkeley, 1993.

[3] P. Freedman, J.-M. Goutal, Y. LeBorgne, and D. Gaudreau. Étude sur des techniques, des méthodes et outils de développement logiciel de systemes temps-réel. Technical Report CRIM-95/12-40, Centre de recherche informatique de Montréal, December 1995.

[4] D. Gaudreau and P. Freedman. Temporal analysis and object-oriented real-time software development: a case study with room/objectime. In *IEEE Real-Time Technologies and Applications Symposium*, Brookline, Massachusetts, June 10-12 1996.

[5] H. Gomaa. *Software Design Methods for Concurrent and Real-Time Systems.* SEI Series in Software Engineering. Addison-Wesley, 1993.

[6] F. Goudenove and L. Doldi. Use of SDL to specify airbus future navigation systems. In *7th SDL Forum*, Oslo, Norway, September 1996.

[7] N. Hizalla, B. Falchuk, and A. Karmouch. A temporal model for interactive multimedia scenarios. *IEEE Multimedia Magazine*, Fall 1995.

[8] D.M. Johnson. The system engineer and the software crisis. *ACM Sigsoft*, 21(2), March 1996.

[9] M. Klein, T. Ralya, B. Polak, and R. Obenza. *A Practitioner's Handbook for Real-Time Analysis.* The Software Engineering Institute Real-Time Handbook Series. Kluwer Academic, 1993.

[10] G. Le Lann. Scheduling in critical real-time systems: a manifesto. In H. Langmaack, W.-P. de Roever, and J. Vytopil, editors, *Formal Techniques in Real-Time and Fault-Tolerant Systems*, pages 511–528. Springer-Verlag, 1994. Proceedings of the 3rd International Symposium, published as Lecture Notes in Computer Science no. 863.

[11] R. Lutz. Targeting saftey-related errors during software requirements analysis. In *1st ACM SIGSOFT Symposium on the Foundations of Software Engineering*, pages 99–106, Los Angeles, California, December 7-10 1993.

[12] J. Rumbaugh. A state of mind: modelling behavior. *Journal of Object-Oriented Programming*, pages 6–12, July-August 1996.

[13] B. Selic, G. Gullekson, and P. Ward. *Real-Time Object-Oriented Modeling.* Wiley, 1994.

[14] D. Simon, E. Castillo, and P. Freedman. On the validation of robotics control systems. part ii: Analysis of real-time closed-loop control tasks. Technical Report 2720, INRIA/Sophia-Antipolis, November 1995.