Constructing High Assurance Asynchronous Real-Time Distributed Systems: A Proof-Based System Engineering Approach

Binoy Ravindran
Bradley Dept. of ECE
Virginia Tech, Blacksburg
VA 24061, USA
E-mail: binoy@vt.edu

Gérard Le Lann
INRIA
Domaine de Voluceau, B.P. 105
78153 Le Chesnay Cedex, France
E-mail: Gerard.Le_Lann@inria.fr

Peng Li
Bradley Dept. of ECE
Virginia Tech, Blacksburg
VA 24061, USA
E-mail: peli2@vt.edu

1. Introduction

Asynchronous real-time distributed systems emerging in many domains are distinguished by the significant run-time uncertainties that are inherent in their application environment, system resource states, and failure occurrences [7]. Consequently, upper bounds on timing variables in such systems such as duration of computational and communication steps—manifestations of application workloads and execution environment characteristics—are not known to exist at design time with sufficient accuracy. Furthermore, many of the emerging asynchronous real-time distributed systems are also safety-critical [10, 8]. Therefore, end-users of such systems require guaranteed assurance on the delivery of desired system properties, particularly safety. This defines a *certification requirement.*

Asynchronous real-time distributed systems thus raise fundamental issues: "How to build timely systems that operate in the presence of uncertain timeliness? Furthermore, how to certify that such systems will deliver properties including timeliness and safety?" In this paper, we discuss an approach for constructing certifiable asynchronous real-time distributed systems.

2. Computational Models

As defined in [9], computational models range from pure synchronous to pure asynchronous. Pure synchrony means that duration of every computational and communication steps have upper bounds that are known at design time, whereas pure asynchrony means that no such upper bounds are known to exist.

Asynchronous computational models have the well known advantage that properties such as safety and liveness can be established even when the "adversary" embodied in design assumptions are violated. Examples include asynchronous consensus algorithms. However, the "curse" of such models is that many problems of interest do not have known deterministic algorithms due to impossibility results [2]. To circumvent this, researchers have augmented the pure asynchrony model with additional semantics, including *timed* semantics and *time-free* semantics. In the pure asynchrony model augmented with timed semantics, called the partially synchronous model, some system modules have pure synchrony semantics and others have arbitrary semantics including pure asynchrony semantics. Pure asynchrony models augmented with time-free semantics are simply called asynchronous models [5]. Examples include unreliable failure detectors [1].

Researchers have also defined the notion of "weakest asynchronous" and "weakest partially synchronous" models [5]. The weakest model is a model that is necessary and sufficient for implementing some given time-free semantics. Thus, a given problem is solved using the weakest partially synchronous model if and only if some minimal set of modules in the solution match pure synchrony assumptions and every other match pure asynchrony assumptions.

3. Timeliness Optimality Using Benefit Accrual Predicates

Majority of the timeliness properties that are currently used for real-time distributed systems focus on deadline timing constraints. With deadlines, it is difficult to express timing constraints that for example, include non-contiguous, optimal and sub-optimal completion time intervals. Furthermore, with deadlines, it is difficult to specify timeliness optimality that for example, include lower bounds for system-wide activity completions at optimal and sub-optimal times, in ac-

1530-2059/02 $17.00 © 2002 IEEE

Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering (HASE'02)
1530-2059/02 $17.00 © 2002 IEEE

cordance with their functional importance. Such timeliness optimality assume significance in the event that the actual operating conditions become "stronger" than what was assumed at design-time. During such conditions, the desired lower bound may be achieved by completing as many important activities as possible at their optimal times and less at their sub-optimal times.

Jensen's benefit functions and benefit accrual predicates [6] allow the specification of such timing constraints and timeliness optimality criteria, respectively.

4. Our Recent Research

The use of benefit accrual predicates for asynchronous real-time distributed systems is the direction of our recent research. In [4], we presented on-line resource allocation algorithms called RBA* and OBA, that seek to maximize aggregate timeliness benefit of asynchronous real-time distributed systems. Furthermore, decentralized RBA* and OBA—RBA* and OBA are centralized—were presented in [3]. RBA* and OBA were limited to "step" benefit functions. This was overcome in [11], where the BPA algorithm that allows arbitrary but unimodal, benefit functions was presented.

5. A System Engineering Approach

We believe that certifiable solutions for constructing asynchronous real-time distributed systems can be designed using the proof-based system engineering framework discussed in [8] by considering the weakest possible models. Models include all assumptions concerning future system operating conditions such as (1) computational models, where the asynchronous model dominates all others such as partially synchronous and pure synchronous models, (2) external event arrival model, where multimodal arrival model dominates all others such as unimodal arrival and aperiodic models, and (3) failure model, where byzantine model dominates all others such as crash and omission models [8]. The domination of one model over another is due to the "strength" of the "adversary."

Thus, we propose to consider the *weakest asynchronous computational model* i.e., the pure asynchronous model that is augmented with time-free semantics such as Chandra and Toueg failure detectors [1]. Furthermore, we propose to specify timeliness optimality using *benefit accrual predicates* such as a user-desired lower bound on system-wide, accrued timeliness benefit. With such computational models and timeliness properties, an architectural and algorithmic solution to a given application problem is designed. Since timeliness optimality is specified using benefit accrual predicates, algorithmic solutions such as [4, 3, 11] can be leveraged in the design of solutions.

Once a system solution is designed, safety and liveness are first provably correctly established for the solution. Timeliness properties are later established by constructing feasibility conditions for the solution that are proven to be necessary, and if tractable, sufficient as well. Such conditions are constructed as non-valued predicates that will embody all possible scenarios that can be deployed by the "adversaries" considered in the design models. The quantification of the feasibility conditions will produce the specification of a system solution that can be certified to exhibit the desired timeliness, safety, and liveness properties.

Ongoing efforts include designing system solutions in this paradigm for problems from the defense domain.

References

[1] T. D. Chandra and S. Toueg. Unreliable failure detectors for reliable distributed systems. *JACM*, 43(2):225–267, 1996.

[2] M. J. Fischer, N. A. Lynch, and M. S. Paterson. Impossibility of distributed consensus with one faulty process. *JACM*, 32(2):374–382, April 1985.

[3] T. Hegazy and B. Ravindran. On decentralized proactive resource allocation in asynchronous real-time distributed systems. In *Proc. of The IEEE/IEICE Symp. on High Assurance Systems Engineering*, Oct. 2002.

[4] T. Hegazy and B. Ravindran. Using application benefit for proactive resource allocation in asynchronous real-time distributed system. *IEEE Trans. on Computers*, 51(8):945–962, Aug. 2002.

[5] J.-F. Hermant and G. L. Lann. Fast asynchronous uniform consensus in real-time distributed systems. *IEEE Trans. on Computers*, 51(8):931–944, Aug. 2002.

[6] E. D. Jensen. Asynchronous decentralized real-time computer systems. In *Real-Time Computing*, Proc. of the NATO ASI. Springer Verlag, Oct. 1992.

[7] E. D. Jensen and B. Ravindran. Guest editor's introduction to special section on asynchronous real-time distributed systems. *IEEE Trans. on Computers*, 51(8):881–882, Aug. 2002.

[8] G. L. Lann. Proof-based system engineering and embedded systems. In *LNCS*, volume 1494, pages 208–248. Springer-Verlag, Oct. 1998.

[9] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996.

[10] M. W. Masters. Challenge problem: System certification for real-time systems that employ dynamic resource management. `http://wpdrts.cs.ohiou.edu/challenge_prob.html`, WPDRTS, 2003.

[11] J. Wang and B. Ravindran. Bpa: A fast packet scheduling algorithm for real-time switched ethernet networks. In *Proc. of The IEEE Int'l. Conf. On Parallel Processing*, Aug. 2002.

Proceedings of the 7th IEEE International Symposium on High Assurance Systems Engineering (HASE'02)
1530-2059/02 $17.00 © 2002 **IEEE**