Dynamic Scheduling of Object Invocations in Distributed Object Oriented Real-Time Systems

Jørgensen, Bo Nørregaard; Joosen, Wouter

Published in:
Lecture Notes in Computer Science

Publication date:
1998

Document version
Publisher's PDF, also known as Version of record

Citation for pulished version (APA):
Jørgensen, B. N., & Joosen, W. (1998). Dynamic Scheduling of Object Invocations in Distributed Object Oriented Real-Time Systems. In Lecture Notes in Computer Science: ECOOP'98 Workshop Reader on Object-Oriented Technology (Vol. 1543, pp. 503 - 507). Springer.

Dynamic Scheduling of Object Invocations in Distributed Object Oriented Real-Time Systems

Bo N. Jørgensen, Wouter Joosen

The Maersk Mc-Kinney Moller Institute for Production Technology,
Odense University, DK-5230 Odense M, Denmark.
e-mail: {bnj, wouter}@mip.ou.dk

Abstract. This paper describes some of the issues that we investigate in order to develop distributed object computing middleware for application domains where timely cooperation and coordination between objects are crucial for guaranteeing correct system behavior. In particular, issues on admission control, resource reservation, and dynamic scheduling of invocations are discussed.

1. Introduction

Our research addresses the development of an ORB framework[1] for application domains where timely cooperation and coordination between objects are crucial for guaranteeing correct system behavior. One such area is Intelligent Manufacturing Systems (IMS).

IMS address the needs of today's rapidly changing consumer driven market, where demands move towards limited production runs of many different products or different models of the same product. It is well-known that conventional manufacturing systems are not suited to handle this task, as it is either technical impossible or too expensive and time consuming to reconfigure the manufacturing system to produce a small batch of a different model. Producing *one-of-a-kind* products is out of the question. Thus, the idea of a flexible automated manufacturing system that can be inexpensively modified is very appealing.

The inflexibility of present manufacturing systems is due to product specific control software and static production equipment settings. This limits both the generality and reusability of the production system, despite the fact that its components are themselves highly flexible. For instance, robots and machine tools can be reused in a variety of production settings. The main reason for the present situation is industrial demands for fast project-implementation-run cycles to minimized production cost. On a short time scale, product specific control software and static production equipment settings can minimize the amount of work necessary

[1] An ORB framework is here defined as the set of software components that compose the architecture of distributed object computing middleware.

S. Demeyer and J. Bosch (Eds.): ECOOP'98 Workshop Reader, LNCS 1543, pp. 503-506, 1998.

for implementing a manufacturing system. However the reduction in work effort is gained at the cost of a significant larger amount of work required for adapting the system to new production equipment and products. The result is increased long-term production cost.

New manufacturing paradigms and manufacturing systems modeling techniques have been proposed to meet the challenges of IMS. Some of these new paradigms and modeling techniques are holonic and agent-oriented manufacturing [1], and object-oriented resource modeling [2]. Common to these approaches are that they model products, production equipment, and manufacturing processes as autonomous, cooperating, and self-organizing entities. Thus relations between entities can emerge and disappear during production. This is for instance the case when products enter and leave a production facility. Relations between production facilities may also change as they form coalitions for providing higher-level production facilities. Two single function robots may cooperate to provide a multifunctional robot.

The environment of IMS is typically a big melting pot of different hardware platforms, operating systems, network technologies, protocol stacks, applications written in different languages, and so on. Frequent upgrades by adding new equipment or software, in order to boost performance or increase functionality enforce this situation. This stresses the need for an infrastructure that can provide interoperability of components in a distributed heterogeneous environment.

Distributed object computing technologies such as *Common Object Request Broker* (CORBA [3]), *Distributed Component Object Model* (DCOM [4]), and JAVA *Remote Method Invocation* (RMI [5]) provide solutions for many of the problems related with distributed heterogeneous manufacturing environments [6].

Distributed object technology facilitates the collaboration of objects by providing mechanisms for objects to transparently make requests to – and receive responses from – objects located locally or remotely. The mechanisms hide the communication with, activation of, and the storing of server objects from clients. Thus this allows new manufacturing paradigms to be implemented in a distributed environment without regard to low-level details such as server location, network protocols, parameter marshalling, etc. However, distributed object technology can not yet guarantee timely cooperation and coordination between objects. This is caused by the lack of mechanisms for guaranteeing end-to-end predictability with respect to timing requirements between objects across networks. Research in this area is a key issue for successful application of distributed object technology in IMS.

2. Extending the ORB framework with real-time capabilities

We are currently investigating how the invocation system of an ORB framework can be extended to guarantee end-to-end predictability with respect to timing requirements. Guaranteeing end-to-end predictability in a dynamically changing distributed real-time system is a challenging task [7]. To the best of our knowledge, there is currently no comprehensive and systematic approach to the problems at hand. Work within the Object Management Group's special interest group on real-time has

resulted in a RFP[2] [8] that identifies a number of requirements for supporting real-time systems that operate in deterministic environments. A deterministic environment allows the designer to calculate all resource requirements during system design and thereby enables him to generate a static schedule that satisfies all deadlines. The RFP requires all submissions to apply fixed priority scheduling for effectuating this schedule. That is, all invocations are statically assigned a priority at design-time, which control their execution at run-time.

Since the resource requirements in an IMS can change dynamically, it is not possible to generate a schedule for execution of invocations at design time instead the schedule has to be dynamically generated at run-time. This implies that the ORB framework must dynamically schedule invocations based on their deadlines and their resource requirements (i.e. CPU, memory, and network bandwidth).

In order to provide end-to-end predictability in a system based on dynamic scheduling we must prevent temporal overloading. Temporal overloading occur when the invocations' resource requirements exceed the system's resource capacity. It is crucial to prevent temporal overloading since it can cause invocations to miss their deadlines. Admission control and resource reservation [9] can prevent temporal overloading by preventing acceptance of more invocations than the system's resources can accommodate. Admission control ensures that all accepted invocations meet their deadlines by rejecting invocations that attempt to overload the system and resource reservation guarantees the availability of resources for executing accepted invocations.

In the simple case where admission control only considers CPU usage, an admission control policy can be based on schedulability conditions from classical scheduling techniques, such as *rate-monotonic* and *earliest-deadline-first* scheduling [10]. Rate-monotonic scheduling has been applied to deterministic real-time systems in [11]. The general case, which also needs to consider memory and network bandwidth usage, requires models for memory and network bandwidth consumption of object invocations. Consequently, a resource scheduling framework that includes CPU, memory, and network bandwidth is a requirement for extending the invocation system of an ORB framework such that it can guarantee end-to-end predictability for object invocations based on admission control and resource reservation. We are currently investigating the basic requirements for this framework.

3. Summary

In this paper we have provided an overview of a complex real-time application, and discussed how admission control and resource reservation can be applied to guarantee end-to-end predictability with respect to timing requirements in an ORB framework based on dynamic scheduling of object invocations.

---

[2] RFP – Request For Proposal – is used by OMG to request proposals for technology standardization.

References

1. S. Bussman, "*An agent-oriented architecture for holonic manufacturing control*", Proceedings of the First International Workshop on Intelligent Manufacturing Systems, 1998.
2. M. Fabian, B. Lennartson, P. Gullander, S-A. Andréasson, A. Adlemo, "*A Generic System Architecture for Flexible Production*", Proceedings of the First International Workshop on Intelligent Manufacturing Systems, 1998.
3. Object Management Group, "*The Common Object Request Broker: Architecture and Specification*", 2.2 ed., Feb. 1998.
4. D. Box, "*Essential COM*", Addison-Wesley, Reading, MA, 1997.
5. A. Wollrath, R. Riggs, and J. Waldo, "*A Distributed Object Model for the Java System*", USENIX Computing Systems, vol. 9, November/December 1996.
6. Hyoung Joong Kim, Byung Wook Choi, "*Intelligent Agents for Holonic Manufacturing System over CORBA*", Proceedings of the First International Workshop on Intelligent Manufacturing Systems, 1998.
7. John A. Stankovic, Krithi Ramamritham, "*What is Predictability for Real-Time Systems?*", Real-Time Systems, vol. 2, pp. 247-254, 1990.
8. Object Management Group, "*Realtime CORBA 1.0: Request for proposal*", OMG Document: orbos/97-09-31.
9. Clifford W. Mercer, Stefan Savage, and Hideyuki Tokuda, "*Processor Capacity Reserves: Operating System Support for Multimedia Applications*", In Proceedings of the IEEE International Conference on Multimedia Computing and Systems, May 1994.
10. C. Liu and J. Layland, "*Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment*", JACM, vol. 20, pp. 46-61, January 1973.
11. D. C. Schmidt, D. L. Levine, and S. Mungee, "*The Design and Performance of Real-Time Object Request Brokers*", Computer Communications, vol. 21, pp. 294--324, Apr. 1998.