

Computer Science 490y Thesis Proposal:
Event and Notification Enhancements to GDMO and OSIMIS

Michael J. Katchabaw
Student number: 123456789
Department of Computer Science
University of Western Ontario
London, Ontario, Canada

E-mail: katchab@csd.uwo.ca

September 20, 1996

The supervisors for this thesis are Dr. James W. Hong and Dr. Michael A. Bauer.

Signature of supervisors:

1 Problem Statement

Most typical management systems are comprised of two main components, namely agents to interact with the resources being managed, and managers, which are responsible for sending various management requests to agents. Communication in this kind of management framework usually takes the form of requests from managers to agents to managed resources (and subsequent replies), and notifications about the occurrence of some event pertaining to a managed resource, which are sent from agents to alert managers in the form of event reports. Managed resources, also known as managed objects, can be defined using several possible specifications, one of which is GDMO (Guidelines for the Definition of Managed Objects). While GDMO provides great flexibility for specifying management requests, GDMO currently allows only the specification of trivial notifications. For more powerful (and useful) events and notifications, GDMO's functionality must be extended.

Current research into the management of distributed systems and applications in the Computer Science Department at Western makes use of OSIMIS, a development environment based on the OSI management model. This environment uses GDMO for the specification of its managed objects. Through experience working with this system, it has become evident that the trivial notification specification capabilities provided by GDMO are not sufficient for practical applications. Furthermore, the generic managers and agents provided by OSIMIS are not well-equipped to process events and notifications.

Consequently, it is proposed that this thesis investigate what is needed to overcome these shortcomings of GDMO and OSIMIS, and how this is best to be accomplished. With that established, the appropriate extensions and modifications to GDMO and OSIMIS would be made, thereby solving the problem at hand.

2 Proposed Solution

The solution to the above problem can be broken down into three main parts:

1. Extending GDMO. To provide more powerful and flexible event and notification handling, the functionality of GDMO will have to be extended. The main extension to GDMO would be to allow the specification types of notifications and the circumstances that would warrant such an event being triggered. Previously, it was possible only to specify that an event existed, and the triggering of the event had to be manually embedded in the code of the agent by the programmer. This method lacks flexibility and requires more extensive knowledge of the workings of the agent and the corresponding managed object(s) than is necessary. The proposed solution would correct this.
2. Modifying the OSIMIS Agents. Because of the methods used by GDMO to handle notifications, the triggering of events currently must be embedded into the code for an agent's managed objects by the programmer. By allowing the specification of notifications and the events that result in their sending in GDMO, this functionality is not required to be programmed by hand. Instead, it could be generated automatically by the GDMO compiler using its specification.
3. Adding Functionality to OSIMIS Managers. Currently, the manager applications for OSIMIS have little or no support for handling event reports generated by agents. They merely send requests to agents and wait for the appropriate replies, and do not listen for event reports from agents. OSIMIS provides some sample code and a simple application that may provide some event report handling support, but this is completely separate from the rest of the manager applications. As a result, it would be beneficial to add the necessary functionality to managers so that they can support the receiving and response to event reports from agents. (This would include adding this functionality to the **DAM Manager**, an OSIMIS-based distributed applications manager developed at Western.)

3 Steps Required For Solution

The proposed problem can be solved by solving each of the parts described in the previous section. The first part that must be solved is the extensions to GDMO. Without these extensions, modifications cannot be made to OSIMIS agents. The next part to be solved would be the modifications to the agent. These changes can be tested using the existing simple and separate OSIMIS tool described above. The last part to be solved would be the additions of functionality to the OSIMIS managers. This is done last because we could then best make use of our solutions to the previous parts of the problem.

Below, some steps to the solution of each part is given:

1. Extending GDMO.
 - (a) Gain a more complete understanding of the current limited GDMO specification of notifications.
 - (b) Determine exactly what extensions are required to GDMO.
 - (c) Define specifications for these extensions to be made to GDMO.
 - (d) Implement these extensions by modifying the OSIMIS GDMO compiler.
 - (e) Test the extensions made.
2. Modifying the OSIMIS Agents.
 - (a) Determine where the code should be generated automatically for the handling of notifications in agents.
 - (b) Determine what code should be generated there. (Most likely this will be a generalization of the code that must be currently manually embedded in agents for notifications.)
 - (c) Using sample managed objects (like UNIX managed objects), replace the previous notification method with the improved code.
 - (d) Test these modifications.
 - (e) Upon successful testing, these modifications should be implemented in all managed objects and agents that make use of notifications.
3. Adding Functionality to OSIMIS Managers.
 - (a) Examine in detail the existing simple event report detection code provided with OSIMIS.
 - (b) Determine what else is required in addition to this code to provide the required event report handling in managers.
 - (c) Add this to the standard OSIMIS manager application (**cmismanager**).
 - (d) Test **cmismanager** to determine if the additions work.
 - (e) Upon success, these additions should be made to the **DAM Manager**, and other OSIMIS-based managers. Testing would again follow.

4 Time Table

The following is a list of milestones and completion dates for the remainder of the project:

- September 30, 1994 - Submit this time table and status report.
- October 31, 1994 - Completion of information gathering phase. This phase includes the following steps:

- Forward a copy of the project proposal to George Pavlou at University College London. It is hoped that he may be able to provide additional insight and information that might be useful.
- Consult the bibliography server (Litserver) and personal libraries of CORDS members for related papers and material.
- Read through this related material and extract relevant information to the project.
- Consult current GDMO documentation and OSIMIS documentation to likewise gather more information.
- Inspect the source code and scripts for the OSIMIS GDMO compiler and OSIMIS agents and managers to get a firm understanding of the system.
- November 18, 1994 - Completion of the requirements analysis phase. This phase includes the following steps:
 - Determine what extensions are needed to GDMO.
 - Determine what enhancements are needed to the OSIMIS GDMO compiler.
 - Determine what enhancements and modifications are needed to the OSIMIS agents and managers.
 - Based on the above, write a requirements document for the project.
- November 25, 1994 - Submit the requirements document for evaluation.
- December 5, 1994 - Submit a revised time-table.
- January 15, 1995 - Completion of the design phase. This includes the following steps:
 - Define a specification for the extensions needed to GDMO.
 - Design the enhancements that are needed to the OSIMIS GDMO compiler.
 - Design the enhancements and modifications that are needed to the OSIMIS agents and managers.
 - Based on the above, write a design document for the project.
- January 29, 1995 - Submit the design document for evaluation.
- January 29, 1995 - Submit a revised time-table.
- March 1, 1995 - Completion of the implementation phase.
- March 15, 1995 - Completion of the testing phase.
- March 22, 1995 - Submit the implementation report for evaluation.
- March 22, 1995 - Submit the testing report for evaluation.
- March 22, 1995 - Submit a revised design document for evaluation.
- March 29, 1995 - Complete preparation for presentation.
- Sometime after March 29, 1994 - Give presentation.