The Design, Implementation, and Performance of the URI Static Scheduling and Real-Time Binding Services For TAO

Kevin Bryan, Matthew Murphy, Priyanka Gupta,Yu Liu, Lisa DiPippo, Victor Fay-Wolfe
Department of Computer Science
University of Rhode Island
{bryank,murphym,guptap,dipippo,wolfe}@cs.uri.edu

1 Introduction

This TAO Workshop presentation will discuss the design, implementation, and performance of the University of Rhode Island (URI) Real-Time Distributed Object Computing group's Static Scheduling Service, and Real-Time Binding Service.

The Static Scheduling Service is an implementation of the Real-Time CORBA 1.0 (RTC1) Scheduling Service standard. The Real-Time Binding Service is an implementation of the standard CORBA Trader Service. We assume a TAO Workshop audience that is familiar with TAO and CORBA, but not necessarily with RTC1 and the details of the RTC1 Scheduling Service. The workshop presentation will provide a brief background on the RTC1 features that impact the Static Scheduling and Real-Time Binding Services, as described in Section 2. The presentation will use as an illustrative example our integration of the Static Scheduling Service and Real-Time Binding Service into the Unmanned Arial Vehicle (UAV) Open Experimental Platform (OEP) that has been produced by BBN, as described in Section 3. We will show how both the Static Scheduling Service and the Real-Time Binding Service work, including internal UML diagrams, and sample configuration files that are at the heart of their implementations, as described in Sections 4 and 5. We expect that by the TAO Workshop date these services will have been committed to the TAO Repository; it will be these committed services that we describe. We will conclude with performance results for the services in the UAV application using common metrics derived on the DARPA PCES program, as described in Section 6. If time permits, we will show how these static services lead to our current work on Dynamic Distributed Scheduling and Binding Services, as described in Section 7.

2 Background

Figure 1 illustrates how the URI Static Scheduling Service is used. The software designer uses an external tool to model real-time aspects of the application. In our initial implementation, the RapidRMA tool from Tri-Pacific Software is used. The tool performs real-time analysis that both provides feedback to the designer and determines the scheduling parameters, such as priorities, priority ceilings, and assignment of tasks to processors, used in the analysis. The Static Scheduling Service and the Real-Time Binding (Trader) Service each use a configuration file generated by the tool. The Scheduling Service uses the scheduling parameters in the file to set the real-time capabilities provided by a RTC1 ORB (CORBA priorities, priority mapping, priority lanes, Real-Time POA policies, etc). The Real-Time Binding Service uses the task-to-node assignments to bind CORBA clients to CORBA servers that have been analyzed by the tool as being capable of meeting the client timing constraints.

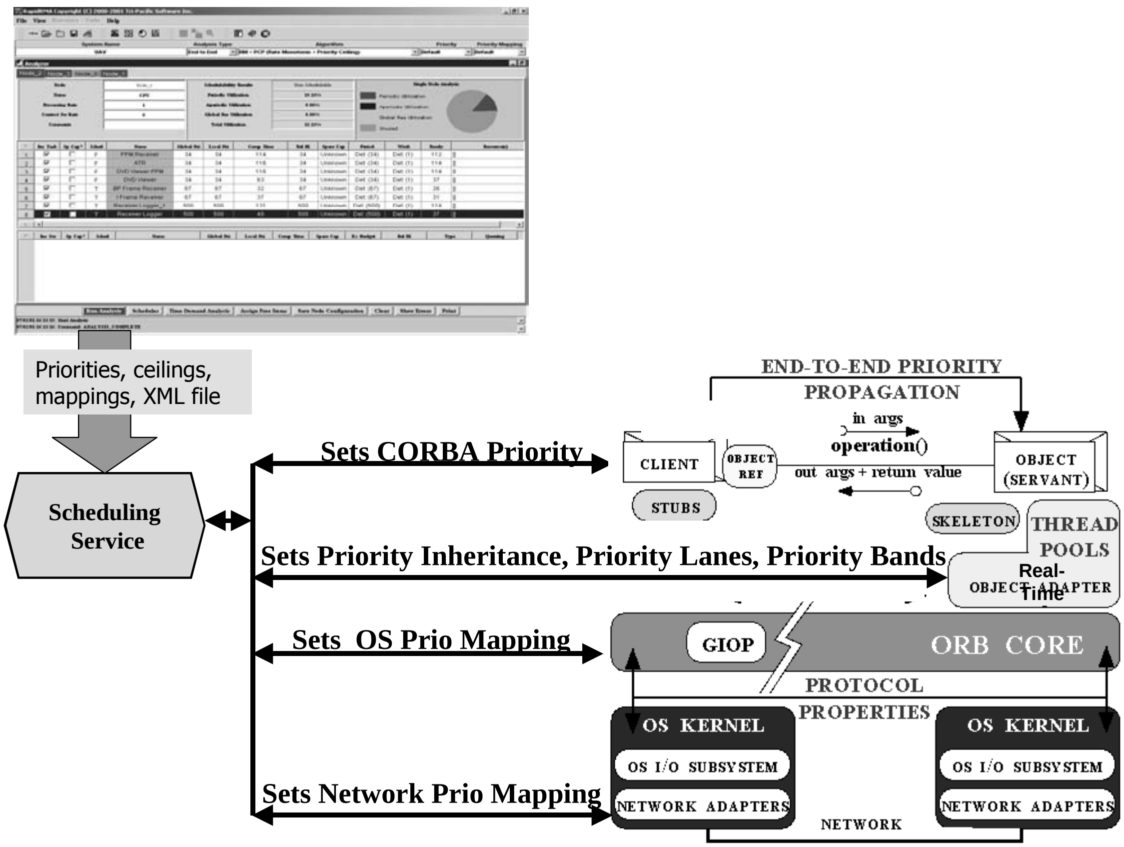

Figure 1: Static Scheduling Service Sets RT CORBA 1 ORB

In the Background of the section of the TAO Workshop presentation will provide a brief tutorial that includes the RTC1 Scheduling Service IDL and semantics, the RTC1 features that the Scheduling Service uses (CORBA priorities, priority mapping, Real-Time POA), and how the CORBA Trader Service works. We will also provide brief background on RapidRMA and how it is used to model real-time applications.

3 Example Application

After the Background section of the workshop presentation, we will introduce the UAV OEP example. Figure 2 shows a UAV sending a video stream to a surface ship. The surface ship has a Distributor node that sends the video, or a modified version of it, to destinations such as human commanders, and automatic target recognition (ATR) software. This version of the UAV is static in that the video stream is established with no dynamic components (e.g. other UAVs or other sources of resource contention).

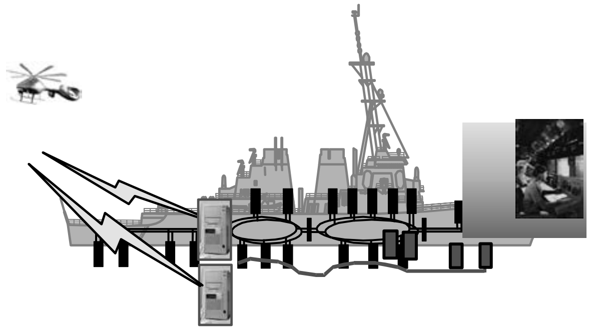

Figure 2: Example Application: UAV Streaming Video To A Surface Ship Distributor

We will show how we took the existing static UAV and inserted calls to our Static Scheduling Service and Real-Time Binding Service. The calls were inserted using an Aspect-Oriented Programming technology called QuO from BBN. The description will involve showing our application-level determination of the real-time parameters in the UAV, such as periods, delays, and resource requirements.

After describing the application, we will present one of the many RapidRMA models that we have of the UAV. We have modeled the static UAV application in every scenario that corresponds to the 19 Experiments described later in Section 6 (these are the common experiments used in the DARPA PCES program). We will show one of these models as an example of how the tool is used to model a real application. As shown in Figure 1, this modeling is the precursor to using the Static Scheduling Services and Real-Time Binding Service.

4 Static Scheduling Service

Following the motivation of the UAV example and how the Static Scheduling Service is used in it, we will show how the URI Static Scheduling Service works. This section of the workshop presentation will include UML diagrams and how each of the IDL features of the standard are implemented. Also included will be an examination of the configuration file produced by RapidRMA and how the Static Scheduling Service uses it.

5 Real-Time Binding Service

Similarly, this section of the workshop presentation will include UML diagrams and how the Real-Time Binding Service uses the existing TAO Trader Service with a real-time internal binding algorithm (based on bindings determined by the external tool using Rate Monotonic utilization worst fit). Also included will be an examination of the configuration file produced by RapidRMA, and how the Real-Time Binding Service uses it.

6 Performance Results

We will present the results of measuring the effectiveness of the two URI services in the UAV OEP. The results will use 19 Experiments that are the basis of a common set of experiments used by the organizations working on the DARPA PCES UAV project. The Experiments measure the latency of "important" video frames and "unimportant" video frame in the presence of different loads on various nodes and network points in the system. It also measures the latency of aperiodic control signals that are returned to the UAV when a target is recognized, which evaluates the technology's capability of handling non-periodic processing and communication. The results of using the Services in the UAV will be measured against a baseline UAV implementation that does not use the Services. Also included will be overhead measurements that use of the URI Static Scheduling Service and URI Real-Time Binding Service incur.

All experiments will be performed in the University of Utah Emulab, a shared experimental testbed accessible via the Internet. This has the benefit that the experimental testbed itself can be made available to members of the TAO community that wish to replicate the experiments which we report.

7 Current Work

If time permits, we will show how the Static Scheduling Service is leading to our development of a Dynamic Scheduling Service that set scheduling parameters for local Real-Time CORBA 2.0 (RTC2) ORB Schedulers, such as Kokyu. We will also show how we are injecting dynamic scheduling algorithms into the Real-Time Binding Service. Some of this will have been presented at the OMG workshop that precedes the TAO workshop, so for many in the audience it will not be necessary to re-hash it, but showing it again in the context of the details of the Static Scheduling Service of this presentation would be valuable if time permits.

8 Conclusion

This TAO workshop presentation will provide an "under the hood" look at the URI Static Scheduling Service and Real-Time Binding Services, both of which should have been committed to the TAO Repository by the date of the Workshop. We will provide background for the audience and use the example of the UAV to illustrate how the Services work. The under-the-hood look will involve UML of the services and an examination of the configuration files that they use. The performance results will show the effectiveness of the Services and their overhead cost.