Unified Object Bus

Services

Manuel Roman (mroman1@cs.uiuc.edu)
Software Research Group
University of Illinois at Urbana-Champaign

Table of Contents

1. Manipulating Component Containers

Component containers provide the execution environment to the components. They provide the address space where components are instantiated and host a set of default components that export minimum required functionality to manage components. One of these components is the Component Manager. This component exports a set of tools to manipulate components (create, delete, attach, etc.) and it is attached to a hook in the domain configurator called "ComponentManager". Components running inside the component container can get a reference to the component manager and request component operations. However, the component manager is a base component and therefore cannot be accessed from outside the component container. To overcome this problem, the unified object bus uses two different components to allow external access to the component manager: CORBA Component Manager Exporter and Telnet Component Manager Exporter.

1.1 CORBA Component Manager Exporter

The CORBA Component Manager Exporter exports the functionality of the Component Manager outside the component container by means of a CORBA interface. This CORBA component allows remote manipulation of components.

When a CORBA Component Manager Exporter is created, it automatically registers itself under the name CORBA/DistributedCM/0, in the component container's naming context (see section 4 in the UOB document for more information about the name hierarchy). This standard name provides the bootstrapping mechanism to retrieve the CORBA Component Manager Exporter.

Table 1 contains the IDL interface of the CORBA Component Manager Exporter.

```
#include "CORBAComponent/CORBAComponent.idl"

interface CORBAComponentManagerExporter: CORBAComponent
{
	typedef sequence<string> stringList;

	exception NotFound {};
	exception WrongHookPath {};

	void shutdownContainer();

	//--------------Operations on Hooks---------------------//
	void createHook (in string UCR, in string hookName)
		raises(NotFound, WrongHookPath);
	void deleteHook (in string UCR, in string hookName)
		raises(NotFound, WrongHookPath);

	//--------------Operations on Components----------------//
	void createComponent (in string compFactName, in string params, out string UCR)
		raises(NotFound);
	void hookComponent (in string UCR, in string hookName)
		raises(NotFound, WrongHookPath);
	void destroyComponent (in string UCR)
		raises(NotFound);
	string getHookedComponent(in string UCR, in string hookName)
		raises(NotFound, WrongHookPath);
	string getComponentInfo (in string UCR)
		raises(NotFound);

	//---------------Information retrieval operations-----------------------------//
	stringList listDomainComponents ();
	stringList listHooks (in string UCR, in string hookPath)
		raises(NotFound);
};
```

Table 1. CORBA Component Manager Exporter Interface

ShutdownContainer finishes the component container. Therefore, all the components that it contains are destroyed.

The **createHook** method creates a new hook. The hook name can be compound (separating the different names by '/'). The hook name is relative to the component provided as the first parameter. If the component is NULL or "/", then the hook name is assumed to start at the domain configurator. All the names in the hook name must exist, except the last one, which is the one to be created.

The **deleteHook** method deletes the hook specified. The rules applied to the component and the hook name parameters are the same as the ones described for the createHook**.**

CreateComponent creates a new component of the type specified by "compFactName" (component factory name, check section 3 in the UOB document). The second parameter is a string that contains the parameters that will be provided to the component. Finally, the third parameter is an "out" parameter and contains the UCR assigned to the component.

HookComponent attaches the component identified by the first parameter (UCR) to the hook (hookName) provided as a parameter. The hook name can be compound and is relative to the domain configurator.

DestroyComponent destroys the component specified by UCR.

GetHookedComponent returns the UCR of the component attached to the specified hook name.a pointer to the configurator attached to the *hookName*, relative to the component provided (UCR). If this configurator is NULL or "/", then the DomainConfigurator is assumed as the starting point.

GetComponentInfo returns a string with information provided by the specified component (UCR).

The **listDomainComponents** method returns a sequence of strings with the UCRs of all the components contained in the component container.

ListHooks returns a sequence of strings with the names of the hooks existing in the provided *hookPath*, which is relative to the specified component (UCR). If UCR is NULL of "/", then the Domain Configurator is assumed.

1.2 Telnet Component Manager Exporter

This is a base component that exports the functionality of the Component Manager through a simple socket. Once the component is instantiated, it listens in a pre-established port for requests. Clients can connect to this component using a socket, and therefore it is possible to interact with it using a telnet connection. The interface it exports is the same as the one exported by the Component Manager; it is simply a wrapper. There is an additional command called "help", which returns a list of all commands available.

2. Browsing the Unified Object Bus

The unified object bus hosts large collections of components and runs in several machines. It is important to be able to browse the state of the bus from a centralized place and in an intuitive way.

2.1 HTTP UOB Browser

This component exports information about the unified object bus by generating HTML pages dynamically. The HTTP UOB Browser retrieves the information from the naming service and generates the HTML pages according to the links selected by the user.

Once this component is instantiated, it is possible to use any Web browser to display the HTML pages. By default this component listens at port 20000. When a web browser connects, it is sent a home page with a list of all the hosts that participate in the unified object bus. When users select one of the machines they obtain a list of all the component containers running in that particular machine. By selecting one of the component containers, it is possible to visualize all the components running in that particular component container. Again, it is possible to select any of those components, and the result is a list of all the hooks created in this particular component. When one of these hooks is selected, the user obtains either the component or the component configurator attached to it with a list of all its hooks.

This component makes it easy for system administrators to check the status of the unified object bus.