Building a Dynamically Reconfigurable minimumCORBA Platform with Components, Connectors and Language-Level Support

Jim Dowling and Vinny Cahill

Distributed Systems Group
Department of Computer Science
Trinity College Dublin
{jim.dowling, vinny.cahill}@cs.tcd.ie

Introduction

Middleware and distributed object request brokers (ORBs) are characterised by the need to support a range of end-user applications and execution environments, each with different requirements, simultaneously and on behalf of different users. The dynamic adaptation of a middleware system provides a means of tailoring the behaviour of the middleware to the needs of the current mix of applications, users and the execution environment.
It is clear from previous work that creating adaptable object-based systems requires the developer to provide some mechanism for constraining dependencies [2] and communication [1] between objects in the face of potential adaptations.
A combination of component-based and reflection approaches to building adaptable ORBs offers the possibility of explicit dependency and communication management between the different subsystems (components) in an ORB. Reflection could allow the automatic generation of code to implement meta-level component connectors that are independent of the base-type of the components.

Connectors and Contracts in the Component Model

We are investigating building a component model on top of the Iguana reflective programming model [4] that provides components with meta-level connectors and contracts for both inter-component communication and explicit dependency management.

Our component model proposes using connectors [6], implemented as meta-level objects using Iguana, to manage inter-component communication policies as well as management of inter-component communication during a component reconfiguration phase.
Connectors encapsulate important aspects of inter-component communication including the:

- policy used to select the recipient of messages
- particular inter-process communication mechanism
- location and identity of the sending/receiving component.

We also propose the use of contracts and callbacks to manage the scope of a component reconfiguration. In our minimumCORBA project, functional contracts are used to specify the static relationships between the different subsystems that make up a minimal ORB. Other types of contracts between components, such as quality of service and extended services contracts, are implemented and managed using meta-level connectors. For example, if the adaptation of a component modifies the component's set of available services or the quality of those services, then any dependent components whose contracts were broken have to re-negotiate their broken contract(s) with the adapted component. Contract renegotiation uses a callback mechanism to inform all affected components of the updated contracts, allowing affected components to also adapt themselves so as to be optimised with respect to the new contracts. A contract renegotiation can, in certain situations, trigger an end-to-end reconfiguration of the middleware system.

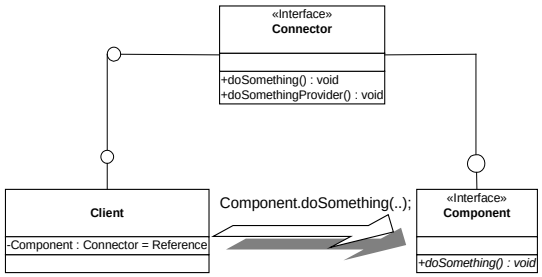

Figure 1a Connectors as Objects

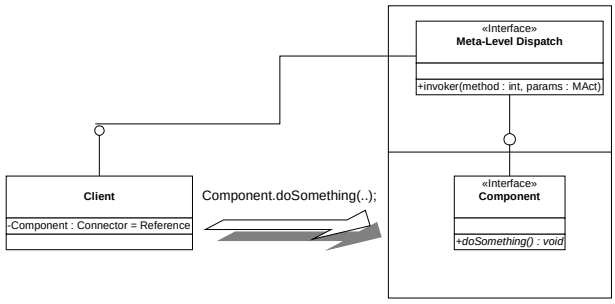

Figure1b Connectors as Meta-level Objects

minimumCORBA

The designer of a component framework for building adaptable ORBs has the conflicting goals of supporting fine-grained adaptation by defining more interfaces and specifying a minimal ORB architecture by defining fewer interfaces. The approach taken by the minimumCORBA project is to capture the static relationships between the different subsystems in minimumCORBA [5] using a component definition language as functional contracts between components. The reconfigurable, or dynamic, relationships between the different subsystems are captured using non-functional (meta-level) contracts and connectors. Connectors can also be used to implement features like component interface evolution, component migration and fault detection.

Language-Level Support

Iguana is a language-independent reflective programming model with support for dynamic adaptation of system software as one of its main goals. A number of small-scale case studies intended to evaluate the use of Iguana as a means of building adaptable system software have already been undertaken. Early results suggest that there may indeed be significant benefits from this approach [3].

An open area, in terms of language-level support for building adaptable systems, is how to abstract out the static and dynamic aspects of a (middleware) architecture at the design phase using language-level constructs. We see Iguana as providing the language level support for component-based programming. Architecture definition languages and component definition languages have already made some progress in this area.

References

[1] Gordon Blair et Al., The Design of a Resource-Aware Reflective Middleware Architecture, Reflection '99, Saint-Malo, France, July 1999.

[2] Roy Campbell and Fabio Con, Supporting Automatic Configuration of Component Based Distributed Systems, 5th USENIX Conference on Object-Oriented Technologies and Systems (COOTS'99).

[3] Jim Dowling, Tilman Schaefer, Vinny Cahill, Peter Haraszti, and Barry Redmond, "Using Reflection to Support Dynamic Adaptation of System Software: A Case Study Driven Evaluation", In Proceedings of the OOPSLA '99 Workshop on Object-Oriented Reflection and Software Engineering.

[4] Brendan Gowing, A Reflective Programming Model and Language for Dynamically Modifying Compiled Software, PhD Thesis, Dept. of Computer Science, Trinity College Dublin, 1997.

[5] Object Management Group –minimumCORBA, OMG TC Document Formal/ 98-08-04, August 1998.

[6] Peyman Oreizy et. Al, An Architecture-Based Approach to Self-Adaptive Software, IEEE Intelligent Systems, May/June 1999.