Specifying Heterogeneous Distributed Components*

Fei Cao[1], Barrett R. Bryant[1], Rajeev R. Raje[2], Mikhail Auguston[3], Andrew M. Olson[2], Carol C. Burt[1]

[1]Department of Computer and Information Sciences
University of Alabama at Birmingham
{caof, bryant, cburt}@cis.uab.edu

[2]Department of Computer and Information Science
Indiana University Purdue University at Indianapolis
{rraje, aolson}@cs.iupui.edu

[3]Computer Science Department
New Mexico State University
mikau@cs.nmsu.edu

Recent years have seen a tremendous interest in component-based software development. By assembling components, software can be produced both efficiently and cost-effectively. The distributed computing systems, which are and will be widely adopted both in the civilian and military sectors, require the development of robust, effective software based on heterogeneous components. Through public interface and private implementation, the independently developed components are expected to be integrated together to realize distributed computing systems, which should satisfy not only the functional requirements such as the nature of computation, but also the nonfunctional requirements like Quality of Service(QoS) [1]. However, the component-based software development for large-scale, de-centralized, robust systems is still in an infancy stage.

In [2] , a framework is proposed for assembling software systems from distributed heterogeneous components, which is based on the notions of a meta-component model called the Unified Meta Model (UMM) [3], a generative domain model [4], and specification of appropriate QoS parameters [1]. In this model, a special component head-hunter is adopted to search for distributed, heterogeneous components and to register their functionality and attempt matching between client and server components. We have implemented a prototype level example in the Voyager [5] environment realizing the mobile search of components as is illustrated in Figure 1. In this figure, there are several processes: (1) An external client component sends a search request with its query information concerning service attributes, etc. (2) The Servlet parses the request parameters and then looks up the component in the repository. (3) The query result is returned to the Servlet. If a matching component is already available, the Servlet returns the handle of that component to the client component, following (7) in Figure 1 . Otherwise in stop (4), the Servlet triggers a search process on the Agent Launcher. (5) The Agent Launcher will retrieve the URLs of remote Component-Info-Retrieval (CIR) objects from the Federated Directory Server; those URLs are registered leveraging the Voyager ORB's federated distributed naming service by remote objects. (6.1-6.4) The Agent Launcher then sends out mobile agents searching for targeted components through CIR objects and the mobile agents return matching components to the Component Repository for further inquiry by external client components. Once a matching component is found, a dialog between client/server components will be performed.

In order to provide enough information in UMM for the headhunter to search for components , which in turn will contribute to the automatic generation of glue and wrapper code after the matching components are found, we need a proper specification of components in a distributed environment, which further requires specification analysis and matching. We propose to add to the Component Registry (as shown in Figure 1) the specification of components to form a library of specifications. At the same time, we will establish the semantic foundation for the specification matching and evaluating choices among multiple matching candidates. The matching between query specification and library specification will be incorporated with an automated approach to glue and wrapper generation.

* This research is supported by the U. S. Office of Naval Research under the award number N00014-01-1-0746.

References:

[1] Brahnmath, Girish J., Raje, Rajeev R., Olson, Andrew M., Auguston, Mikhail, Bryant, Barrett R., and Burt, Carol C., "A Quality of Service Catalog for Software Components", *to appear in Proceedings of (SE)^2 2002, the Southeastern Software Engineering Conference*, April 9-11, 2002,Huntsville, Alabama.

[2] Raje, R., Auguston, M., Bryant, B., Olson, A., Burt, C., "A Unified Approach for the Integration of Distributed Heterogeneous Components", *Proc. Monterey Workshop on Engineering Automation for Software Intensive Systems*, 2001, p.109-119.

[3] Raje, Rajeev R., "UMM: Unified Meta-object Model for Open Distributed Systems", *Proc. ICA3PP 2000, 4th IEEE Int. Conf. Algorithms and Architecture for Parallel Processing*, 2000.

[4] Czarnecki, K., Eisenecker, U.W., *Generative Programming: Methods, Tools, and Applications*, Addison-Wesley, 2000.

[5] Voyager, *http://www.objectspace.com*.

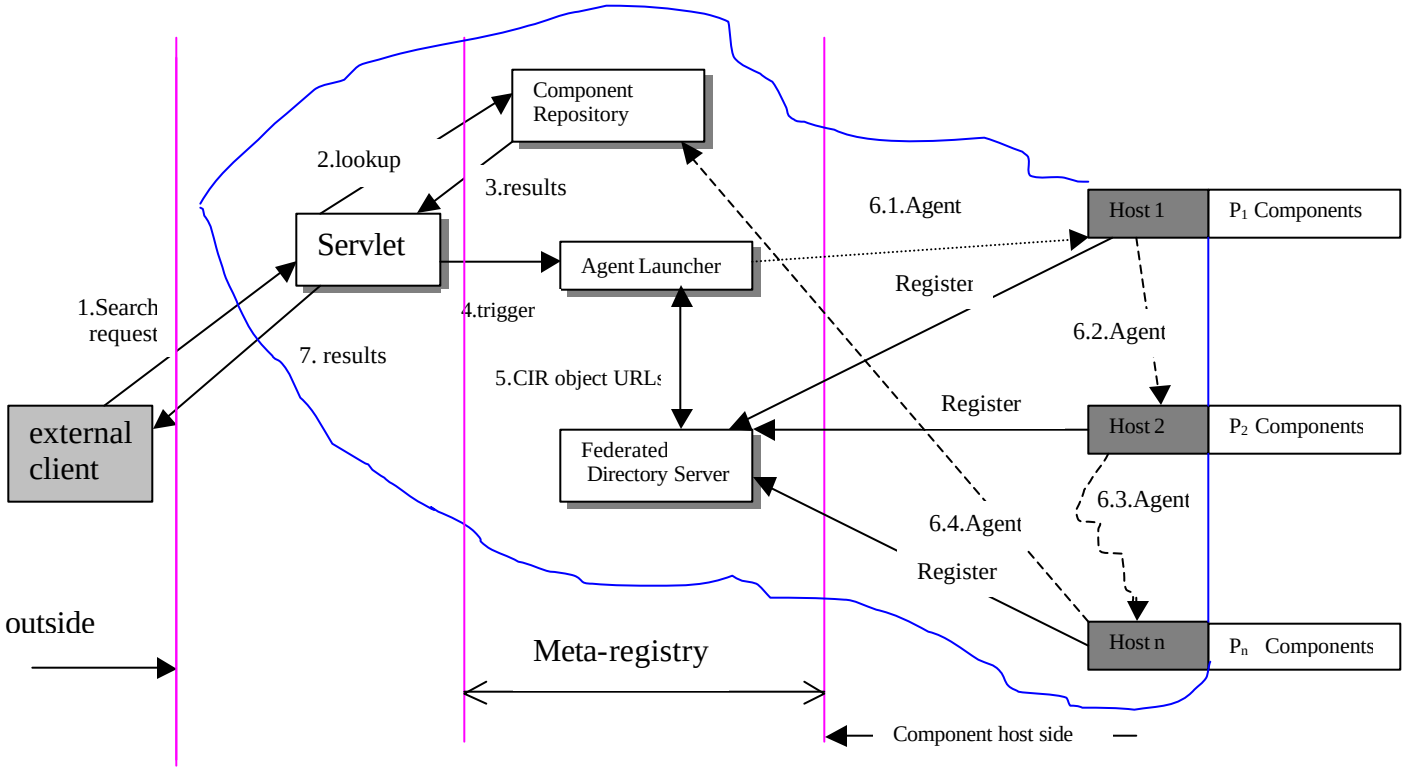

Figure 1: Architecture of Searching Component with Voyager Agent (CIR: Component-Info-Retrieval)