Dynamic Integration of Peer-to-Peer Services into a CORBA compliant Middleware

Rüdiger Kapitza[1], Udo Bartlang[2], Holger Schmidt[3], and Franz J. Hauck[3]

[1] Dept. of Comp. Sciences, Informatik 4, University of Erlangen-Nürnberg, Germany
rrkapitz@cs.fau.de
[2] Siemens AG, Corporate Technology, Munich, Germany
udo.bartlang.ext@siemens.com
[3] Distributed Systems Laboratory, University of Ulm, Germany
{holger.schmidt,franz.hauck}@uni-ulm.de

Abstract. Peer-to-Peer computing has evolved over the last few years and is applied to a rising number of applications. Following this development we present a decentralised approach to dynamically select, load and integrate locally not available peer-to-peer based services into a CORBA-compliant middleware. This is achieved by extending and improving the mechanisms for dynamic service integration of JXTA an open peer-to-peer infrastructure. At object level we build on the fragmented object model provided by the AspectIX middleware to seamlessly integrate and use peer-to-peer services instead of common CORBA client/server-based implementations.

Common object-oriented middleware infrastructures provide mature support for client/server programming but usually miss facilities to easily and seamlessly support other interaction paradigms. We present a flexible approach to integrate peer-to-peer-based services into a CORBA-compliant middleware at object level based on our AspectIX middleware. This offers the possibility to replace client/sever interaction by peer-to-peer communication without any client-side modification.

The integration of peer-to-peer services at object level is achieved by a modularisation of the object reference (IOR) handling using a *generic reference manager* with *portable profile managers* [2]. These profile managers encapsulate all tasks related to reference handling, i.e., reference creation, reference marshalling and unmarshalling, external representation of references as strings, and type casting of representatives of remote objects. Currently, AspectIX provides profile managers for standard CORBA and additionally offers support for the fragmented object model [4] and other non-CORBA middleware platforms, like Jini or Java RMI. In this work, we implemented a profile manager for integrating the JXTA middleware [1] and its peer-to-peer services based on the support for the fragmented object model.

A fragmented object might be composed of several fragments and could be distributed over multiple machines. While a fragmented object offers a standard object interface to the outside, it can exhibit an arbitrary architecture inside. The fragmented object model provides maximal flexibility but might result in a higher development effort especially for recurring demands like fault-tolerance or a peer-to-peer-based internal communication. The integration of JXTA to support peer-to-peer interaction eliminates this drawback as it comprises a fully-fledged middleware.

However, the JXTA middleware cannot be compared to a standard object-based middleware supporting the client/server paradigm. Instead, it misses the concept of an object reference and an application-directed interface, but offers lower level network-directed concepts like a service description specifying a network protocol and an abstract communication endpoint to connect other services supporting the same protocol. Thus, a JXTA IOR profile contains a metadata description called *module specification advertisement* describing a JXTA service and its supported protocol. By combining the fragmented object model and the JXTA service concept we bridge the gap between a standard object-based client/server middleware and the JXTA peer-to-peer infrastructure.

When coping with a fragmented object, the binding usually requires the dynamic loading of code, as it is not feasible to install and load all code modules at every node of the system. The reason is that these would only be used by some nodes and these may even not be known in advance. Therefore, we recently proposed a dynamic loading service that enables the dynamic loading of platform-specific code on demand [3]. In contrary to that work, we propose a generic and decentralised peer-to-peer-based lookup, selection and loading process. This allows multiple parties to independently provide implementations for a certain object. The current prototype extends existing concepts of the JXTA platform to dynamically select and load code, based on advertisements and extends those to provide a truly platform-independent support for the dynamic loading of platform-specific code. Our architecture is composed by three components: A *decentralised implementation repository* represented by a JXTA peer group hosting metadata about all available objects and their implementations. A *code provider*, which is run by every entity offering code, that is responsible for publishing implementation-related advertisements via the implementation repository and sharing the code. Finally, there is the *dynamic loader*, that builds the core of our prototype. It is invoked by the JXTA profile manager during binding time of a JXTA-based fragmented object to select and load an implementation.

Even though our current prototype enables the dynamic integration of peer-to-peer-based services by a precise selection of platform-specific code, we currently assume that a concrete implementation is self-contained. We intent to investigate solutions to support implementations that reference other implementations enabling more modularity.

References

1. Li Gong. JXTA: A network programming environment. *IEEE Internet Computing*, 5(3):88–95, 2001.
2. Franz J. Hauck, Rüdiger Kapitza, Hans P. Reiser, and Andreas I. Schmied. A flexible and extensible object middleware: CORBA and beyond. In *Proc. of the Fifth Int. Workshop on Software Engineering and Middleware*. ACM Digital Library, 2005.
3. R. Kapitza and F.J. Hauck. DLS: a CORBA service for dynamic loading of code. In *Proc. of the OTM Confederated Int. Conf.*, Sicily, Italy, 2003.
4. Mesaac Makpangou, Yvon Gourhant, Jean-Pierre Le Narzul, and Marc Shapiro. Fragmented objects for distributed abstractions. In T. L. Casavant and M. Singhal, editors, *Readings in distributed computing systems*, pages 170–186. IEEE Computer Society Press, 1994.