

# Metabroker: A Generic Broker for Electronic Commerce

Steve Caughey, David Ingham & Paul Watson

Department of Computing Science, Newcastle University,  
Newcastle upon Tyne, NE1 7RU, UK.

{s.j.caughey | dave.ingham | paul.watson}@ncl.ac.uk

## Abstract

From a commercial perspective, the Web has promised much more than it has delivered and the dream of world-wide Internet business enterprises has yet to become a reality. Electronic commerce currently resembles a vast, sprawling bazaar in which visitors must wander through countless market stalls. For most potential customers, this is frustrating and time-consuming.

In traditional commerce, middlemen, or brokers, make it easier for customers to find, compare and buy because they aggregate goods and services from a variety of sources and display them in a way which is helpful to customers. In the electronic marketplace of the near future, there are likely to be large numbers and varieties of brokers. We believe that the widespread and rapid deployment of brokering services depends upon the availability of building blocks which are sufficiently generic that they can be tailored to produce a wide range of specific brokers. We call this collection of building blocks a **generic broker**. If such a generic broker framework is not available then specialised brokers will be implemented in an ad-hoc manner with a consequent wastage of effort, while their construction will be beyond the resources and skills of many companies who would otherwise benefit from them.

In this paper we present the design of the *Metabroker* system, a generic framework for the construction of specialist electronic brokers. Our design is based upon the integration of distributed object, metadata and object database technologies.

**Keywords:** electronic commerce; brokering; metadata; distributed objects

## Metabroker Design

Metabroker comes in two parts, a Generic Broker Module and a Specific Broker Module. The *Generic Broker Module* (GBM) provides a distributed object framework and a set of commonly useful services. This module also contains a run-time library of the common data types (and protocols) encountered on the Internet. All objects within the GBM offer a common set of useful interfaces and are either instances of the data types offered by the GBM or by the Specific Broker Module.

The external entities of concern to a broker are the clients who wish to make use of the broker and the resources which the broker deems of interest to clients. Within the Metabroker, proxy objects (or just *proxies*) are used to represent and maintain information relating to clients and resources. In addition to these external entities the Metabroker will also have its own internal entities. These will include services offered by the broker, services used internally, and entities used to record complex relationships between entities within the broker. All of these entities are also structured as objects, which we collectively call *relationship objects*. Objects can register their interest in specific events with an *Event Service*, or can inform the service of events they can generate and of objects to be informed when this occurs. An *Agent Service* allows clients and resource providers to create agents which can respond on their behalf to the occurrence of specific events.

That part of the broker which is specific to some particular information domain is envisaged as a plug-in Specific Broker Module (SBM). This configures the GBM, populating it with the necessary objects and then monitoring its behaviour. It is capable of on-line reconfiguration where this is deemed necessary. Once configured the Metabroker can go live, listening for and responding to clients.

In order to validate some of the concepts presented in this paper, we have implemented several prototype systems. One of these prototypes was a simple toolkit for building brokers that utilise forms-based Web services. An example broker, known as the *Informed Traveller*, was built which provides a journey planning service encompassing both rail and air travel. The aim of another prototype, *QObjects*, was to demonstrate how content metadata could be structured and queried in a generic manner. The experience gained in implementing our prototypes has convinced us of the viability of the Metabroker project. We aim to utilise components from these prototypes together with our previous experience in distributed object services [Parrington95, Caughey93, Ingham95] to implement the Metabroker architecture.

## References

[Caughey93] S. J. Caughey et al., "SHADOWS: A Flexible Support System for Objects in a Distributed System," Proceedings of the 3rd IEEE International Workshop on Object Orientation in Operating Systems (IWOOS), Ashville, North Carolina, USA, December 1993.

<URL:<http://arjuna.ncl.ac.uk/group/papers/p028.ps>>

[Ingham95] D. B. Ingham, M. C. Little, S. J. Caughey, and S. K. Shrivastava, "W3Objects: Bringing Object-Oriented Technology To The Web," The Web Journal, 1(1), pp. 89-105, Proceedings of the 4th International World Wide Web Conference, Boston, USA, December 1995.

<URL:<http://www.w3.org/pub/Conferences/WWW4/Papers2/141/>> OR

<URL:<http://w3objects.ncl.ac.uk/pubs/bootw/>>

[Parrington95] G. D. Parrington et al., *The Design and Implementation of Arjuna*, USENIX Computing Systems Journal, Vol. 8, No. 3, Summer 1995, pp. 253-306.

<URL:<http://arjuna.ncl.ac.uk/group/papers/p048.ps>>