GAIA: A Generic Architecture and Applicable Platform for Information Brokerage

Authors:

*Hands J., **Darzentas J., **Darzentas J. S., **Spyrou T., **Koutsabasis P., ***Smith R.,

* Fretwell-Downing Data Systems Ltd., Brincliffe House, 861 Ecclesall Road, Sheffield S11 7AE, UK., Tel +44 114 281 6000 Fax +44 114 281 6000, email: jhands@fdgroup.co.uk

** Research Laboratory of Samos, Department of Mathematics, University of the Aegean, GR 83200 Karlovassi, Samos, Greece, email: {idarz, jennyd, tsp} kgp@aegean.gr

*** KYROS, 55 Evripidou Ave., Kallithea 176 74 Athens, Greece, Tel. +30-1-9309908 Fax: +30-1-9309908, email: rsmith@compulink.gr

Abstract

The massive growth and understanding of Internet network and technology is leading to a notable shift in electronic commerce research. After studying various service infrastructure issues (for example: product cataloguing, electronic payment, delivery of goods), the question is now how these services can be offered by one provider-broker as one set of uniform services that involves a large number of suppliers. The ACTS (AC221) project GAIA (Generic Architecture for Information Availability) designed a brokerage service and software architecture that integrates the variety of services that are required in the electronic commerce world, in a manner that is generic and applicable to diverse application domains. This paper focuses on the GAIA functional architecture and debates the technical approach taken with regard to the aims of the brokerage service. The evaluation of the GAIA system is also presented from the perspective of the customer, supplier and system designer.

1. Introduction

The prevailing research focus in the area of electronic commerce has primarily targeted solutions related to service infrastructure issues, the application of real –life commerce tasks and procedures to the electronic world. This has been accomplished by designing or selecting suitable technologies and tools. Whilst there is ongoing research for many of these infrastructure issues (for example: product cataloguing, payment, delivery of goods), the massive growth of the Internet is shifting electronic commerce research towards the integrated provision of these services. The primary vehicle for this integration is the implementation of brokers which represent a variety of service providers / product suppliers in various application domains.

As the Internet technology has proven suitable for suppliers to provide their products electronically, there has been an explosive growth of on-line stores merchandising a variety of products and services. However such services offer only a subset of the required functionality available from traditional commerce outlets. Additionally customers have great difficulties in using these systems. These difficulties result from a number of issues, examples include the selection of the most appropriate service, the differences in the look and functionality of the user interface and the availability of products in a multiple of different electronic formats available from on-line stores.

Information brokerage systems have the goal of providing an integrated set of electronic commerce services, representing a large number of suppliers, in a uniform manner. Brokerage systems are defined by architectures, technological platforms and accompanying business models that attempt to address the needs of stakeholders (customers, information providers, others) involved in the electronic information service chain. Therefore a brokerage system allows:

- information providers to access the global marketplace by addressing various issues such as the offering of integrated services, the promoting of the providers services, the solving of technical issues (interoperability), etc;
- customers to access a uniform environment, with many information providers (a large market in one step), with an integrated set of services;
- other stakeholders (infrastructure providers, banks, etc.) to meet their particular needs;

The paper is based on the work that has been carried out in the context of a European Union ACTS (AC221) project GAIA [1, 3]. The GAIA project has developed a sector and supplier independent Generic Architecture for Information Availability based on brokers, to support multilateral information trading. Additionally the developed architecture has been demonstrated in three application domains: music, technical components and publishing. The paper presents the functional and software architecture of GAIA and debates the suitability of the technical approach taken with the aims of the brokerage service. It also discusses the evaluation results from the perspective of the customer, supplier and system designer.

The following section overviews the objectives of the GAIA project as they derive from current electronic commerce and specifically brokerage systems research. Section 3 presents the GAIA reference model and functional architecture, highlighting the basic actors, roles, functions and components of the GAIA system. Section 4 presents the technical approach taken by the project and debates its suitability versus the aims of the brokerage service. Section 5 presents the key results of the GAIA evaluation from the perspective of the customer, supplier and system designer. Section 6 presents the conclusions.

2. GAIA Objectives for Brokerage

The primary achievement of the GAIA project was the development of a sector and supplier independent Generic Architecture for Information Availability to support multilateral information trading. The resulting GAIA architecture facilitates location independence and the delivery of information, content & digital services through a scaleable brokerage model broadly applicable to distributed information supply chains and networks. Thus the design of the GAIA architecture was targeted towards the following aims:

- To be generic, covering a wide range of required services and applicable to a diverse range of domains;
- To be scaleable and applicable to the distributed on-line nature of today's electronic commerce supply chains;
- To allow implementations that are commercially feasible;
- To ensure fair competition by allowing both large and small companies to enter the brokerage environment, allowing quality and completeness of service to be market differentiators and promoting cultural diversity.

The implementation of the GAIA platform, apart from complying with the aims of the architectural design, must:

- offer usable ways for the customer to access the range of services
- handle the heterogeneity of the systems deployed by the actors involved in the supply chain and the heterogeneous networks linking them with the minimum cost;
- be massively scaleable through distribution.
- ensure the security of user transactions;

The basic outcome of GAIA was the definition of the information brokerage architecture and the development of a generic toolkit for the provision of brokerage services. The GAIA architecture is independent of the design of either the user interface or the underlying database schemata of the brokerage system, being based on a functional specification of the brokerage platform. This offers the flexibility to the developers of a specific solution to customise both user interfaces and underlying database schemata to their own requirements.

3. The GAIA Reference Model

GAIA has modelled the brokerage environment in the application domain of electronic commerce from a number of perspectives, differing in the level of abstraction. At the most abstract level, the GAIA Reference Model provides a common basis for the description and specification of brokerage systems. The GAIA Reference Model is defined in terms of the Roles, Actions, Events and Entities involved in electronic brokerage. The GAIA Functional Architecture defines the functional elements of the GAIA system, embodying the GAIA Reference Model. The Functional Architecture specifies the roles and relationships between the GAIA Services which instantiate the Actions and supporting Events of the Reference Model.

The GAIA reference model is defined by the entities (actors) of customer, broker and supplier, and the actions of search, locate, order and deliver. These very generic entities and actions have been found to be a good basis to discuss and develop a working demonstrator broker infrastructure, with brokers and suppliers spread around Europe.

Whilst brokerage involves various phases of business operation [3], including awareness creation etc., it is the transaction phase that is of most concern to GAIA. As defined in (3), during the final Transaction stage of the brokerage workflow and decision process, transactions occur as a series of co-ordinated conversations between broker, purchaser and supplier roles. The actions, which the GAIA architecture caters for during these transactions, are:

- *Search*, where the customer describes the nature of the product, and the broker returns the identity of products matching this description
- *Locate*, where the customer obtains from the broker the location and terms of supply for a required item
- *Order*, where the customer requests supply of the required item from the given location
- *Delivery*, where the broker causes the item to be sent to the customer

Around these main actions are supporting actions, of which the most important are authentication, tariffing, and connection to payment systems.

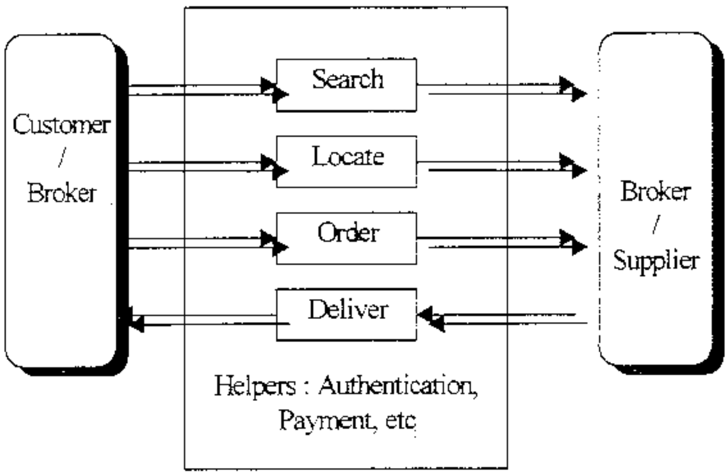

Figure 1 : The GAIA reference model: Generic Actors, Roles and Functions

Central to the reference model is the entity of the GAIA broker, and the roles this entity can undertake. The GAIA broker, acts as both a locator and supplier of information or services to customers, i.e. users who seek services and information online, and likewise a distribution mechanism for suppliers wishing to promote goods and services. The broker either supplies the information or service to the customer itself (in which case it is acting in the supplier role), or else (playing a customer role itself) it sources the information or service from another broker or supplier. Thus, the supply chain may take a number of different forms:

- *a minimal chain*, where the customer and the broker are the ends of the chain, and there are no intervening links. In this case, the broker plays the role of supplier to the customer.
- *a three-piece chain*, where the broker deals with the customer and the supplier, but not with any other broker.
- *a longer chain*, with one or more inter-broker operations.

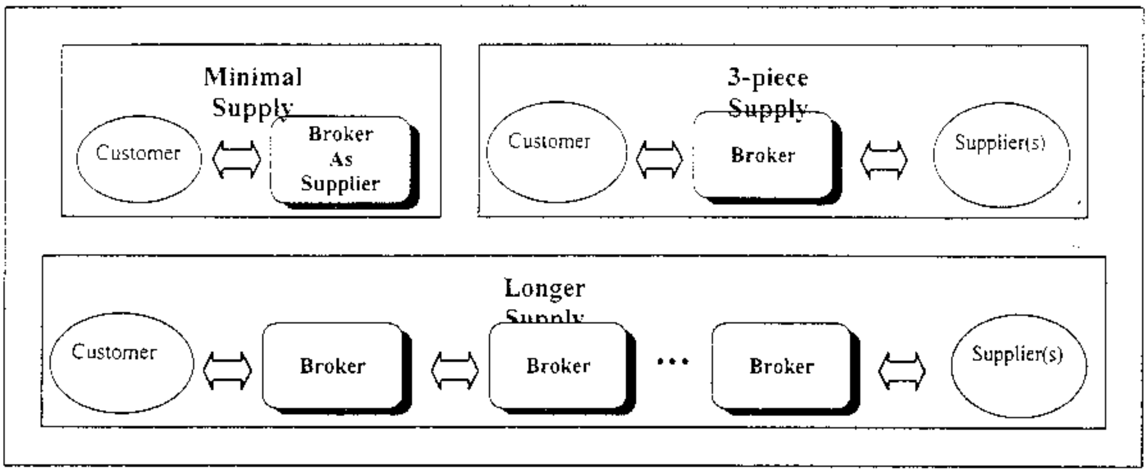

Figure 2 : The GAIA reference model: Broker Roles and Corresponding Supply Chains.

Information about online services or information (generically called products) is maintained by brokers, so that they know where to locate the products required by the customer. The information maintained by a broker may be arbitrarily specialised, with special brokers for particular domains or geographical regions. Brokers may be federated, thereby extending the domain or geographical coverage of individual brokers.

It is obvious that a customer that wishes to interact with the GAIA brokerage system, needs not to be aware of this complex underlying software and network infrastructure. The next section briefly presents the GAIA functional architecture, which identifies the basic functions of the GAIA system. These basic GAIA functions need to be represented by the GAIA user interface.

4. The GAIA Functional Architecture

The GAIA functional architecture decomposes the overall functionality of the brokerage system into a number of components, and describes the roles and relationships of the components, and the manner in which they interoperate. The key elements of the GAIA functional architecture are the *GAIA kernel, Functional Unit Managers (FUM), Functional Units (FU),* and *abstract primitives (AP).*

4.1. Functional Units

The brokerage system provides a number of services to its users and each of these services can be provided by a number of different candidate technologies. For example, in the publishing domain Z39.50 technology is the most prevalent technology used to search for a product. Within the GAIA architecture the required brokerage operations are described in terms of abstract primitives, which can be mapped to the protocol requests of whatever technology is selected to support the function. A mapping component, called a Functional Unit (FU), is defined for each candidate technology, and converts calls to abstract primitives into protocol instructions. The FU acts as an interface between its particular technology and the rest of the brokerage system. Functional Units are defined for each candidate technology that can be used to fulfil a particular functional need of the brokerage system. A Functional Unit accepts abstract primitive invocations, and maps them to calls to the particular technology to which it is dedicated. The results of the calls to the technology are translated into the corresponding abstract primitives and returned by the FU.

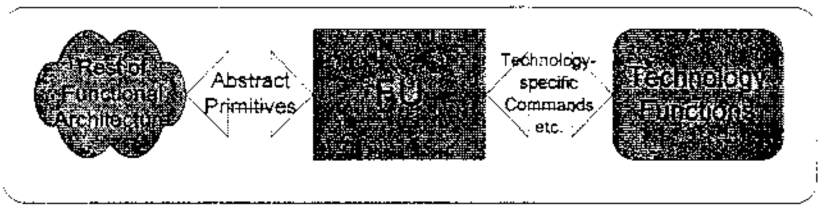

Figure 3. GAIA Functional Unit

4.2. Functional Unit Managers

As noted above, a number of different candidate technologies can be used to fulfil a particular functional requirement of the brokerage system. However, the operations that need to be carried out remain the same; regardless of the selected technologies, the functional requirements do not change. As a number of different Functional Units can exist which fulfil the same functional requirement of the brokerage system, in order to select the most appropriate FU (and technology), the brokerage system needs to know which is most useful at any particular time. This is the responsibility of the Functional Unit manager, or FUM. Thus each function of the brokerage system is represented to the GAIA functional architecture by a single FUM, which is invoked in terms of abstract primitives by the Broker kernel. This FUM selects the most appropriate of the candidate technologies, and calls the corresponding FU.

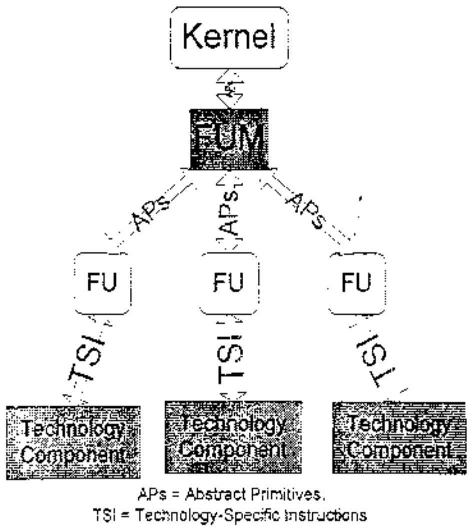

Figure 4. Communication between FUM and FU

4.3. The Broker Kernel and Abstract Primitives.

The kernel of the brokerage system acts as a buffer between the FUMs, and as a bus for the transmission of abstract primitives between FUMs. It also acts as a repository of local information and as a shared data store for FUMs. All calls to abstract primitives are executed via the kernel, which exports all the abstract primitives imported by the various FUMs, and imports all those exported by the FUMs.

Communication between the GAIA kernel and FUMs, and between FUMs and Functional Units, is carried out in terms of abstract primitives. These abstract primitives correspond to a particular operation which a FUM will carry out on behalf of the kernel, or which the FUM expects the kernel to carry out. Each FUM imports a set of abstract primitives, representing those services which the FUM expects to receive from some other part of the system. The services that the FUM is prepared to provide to other elements of the brokerage system are presented in the form of exported abstract primitives. All abstract primitives are imported from, and exported to, the kernel. The kernel acts as a bus for abstract primitives. Abstract primitives are also used in communication between the FUM and its FUs. The FU exports abstract primitives to the FUM.

4.4. Elements of the GAIA Functional Architecture: Description of FUMs

The core functions of the brokerage system are illustrated in terms of FUMs in below Figure 3. This is followed by Table 1, which contains a brief description of every FUM specified by the GAIA Functional Architecture.

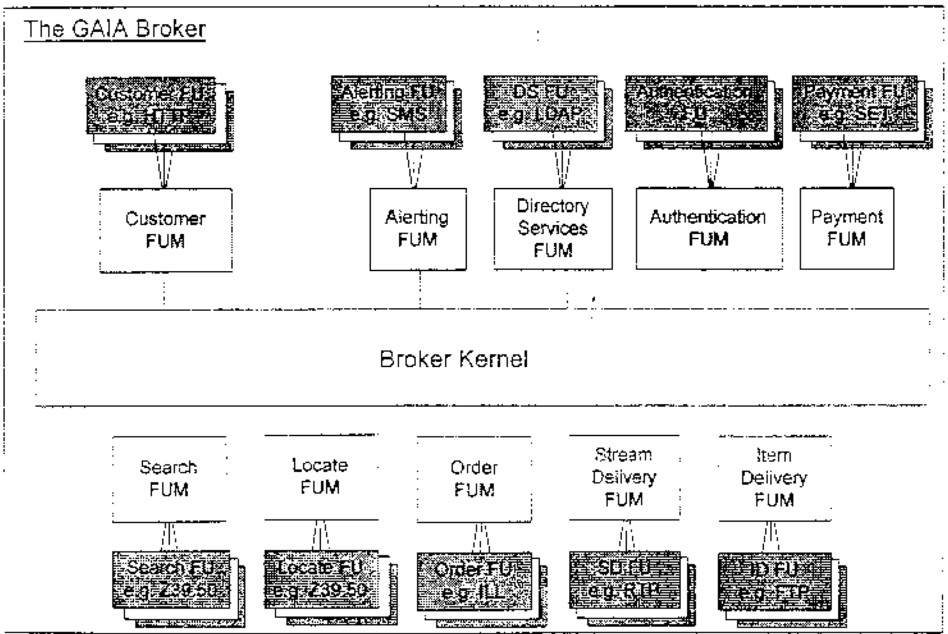

Figure 5. GAIA Functional Architecture FUMs

FUM	Responsibilities
Search	Accepts requests to carry out a search for information Products that fit a particular user description. It returns lists of identifiers of information Products that fit the description.
Locate	Accepts Product identifiers, and discovers where they may be obtained. It returns lists of Suppliers and locations for the Product.
Order	Manages negotiations between a Customer and a Supplier, in order that agreement may be reached on the terms of availability of a particular Product or group of Products. Following the negotiation phase, the order FUM accepts purchase commitments from the Customer and forwards them to the Supplier. It returns a notification of the status of the order Action.
Item Delivery	Manages the delivery of file-structured items to the Customer.
Stream Delivery	Manages the delivery of real-time multi-media data streams to and from the Customer.
Payment	Provides a mechanism for payment from one actor to another.
Authenticati on	Provides a mechanism, which allows a user to prove his identity to the brokerage system.
Alerting	Supports the alerting of users with regard to events that are of their interest
Customer	Provides an interface for the user to allow him to interact with the brokerage system. It also alerts him when a Customer-specified event occurs.
Directory Services	Provides an interface between an external directory service and the brokerage system.

Table 1. FUM Responsibilities Summary

The GAIA broker is a distributed entity, with different FUs and FUMs running on different platforms to the kernel. The required inter-module communication is achieved by the use of the CORBA model. The internal operations of the broker (the abstract primitives imported and exported by the FUMs) are encoded using the IIOP [6] inter-orb protocol, along with any data or other variables required for the operation, and then transported among the FUMs. The pilot implementation developed by the GAIA project supports this distributed model.

5. Technical Approach

This section overviews the technical approach taken towards the implementation of the GAIA functional architecture. This development was divided into two major parts: a) the selection of technology and definition of the internal interfaces of the GAIA broker and b) the selection of protocols that are going to be used in the GAIA trials. These decisions had to take into account the aims of the GAIA project.

5.1. GAIA Internal Interfaces Definition

The internal interfaces of the GAIA Functional Architecture were defined in CORBA IDL (Common Object Request Broker Architecture Interface Description Language). CORBA [4] is an emerging open distributed object-computing infrastructure being standardised by the Object Management Group (OMG). CORBA specifies a system that provides interoperability between objects in a heterogeneous, distributed environment and in a way transparent to the programmer. According to the CORBA Model an object (called Server object) can register an interface which it implements. This interface becomes available for all the distributed system. A Client object can locate available implementations of a particular interface and use methods of this interface to control the remote object. Client and server objects can actually be situated on different machines with different operation systems, they can be implemented by different program languages. CORBA core components transfer requests over the network and perform all necessary format transformation in order to hide heterogeneity of the system.

The central component of CORBA is the Object Request Broker (ORB). It encompasses all of the communication infrastructure necessary to identify and locate objects, handle connection management and deliver data. In general, the ORB is not required to be a single component; it is simply defined by its interfaces. The ORB Core is the most crucial part of the Object Request Broker; it is responsible for communication of requests. The basic functionality provided by the ORB consists of passing the requests from clients to the object implementations on which they are invoked. The ORB simplifies distributed programming by decoupling the client from the details of the method invocations. This makes client requests appear to be local procedure calls. When a client invokes an operation, the ORB is responsible for finding the object implementation, transparently activating it if necessary, delivering the request to the object, and returning any response to the caller.

Object interfaces are specified in an interface definition language, called the OMG IDL (Interface Definition Language). This language defines the types of objects according to the operations that may be performed on them and the parameters to those operations. IDL is the means by which a particular object implementation tells its potential clients what operations are available and how they should be invoked. From the IDL definitions, it is possible to map CORBA objects into particular programming languages or object systems.

5.2. Selection of protocols

The GAIA functional architecture defines the basic actions of the system and its implementation consists of components that interoperate according to the internal GAIA interfaces definition, and can be distributed and written in different programming languages. In addition to that implementation, there has to be a set of components that are customised to external protocols that are specific to the functions that are defined in GAIA.

The criteria for selection of those reflected the over-riding need to specify a workable standard, with the best chance of producing brokerage systems that are interoperable with other information navigation systems. Key selection criteria include the following:

- popularity and acceptance
- standards status
- functionality
- feasibility of integration with other information navigation systems.

The extra protocols are grouped into extension modules. The extension modules specified and the protocols they are based upon are summarised in the following table.

Extension Module	Protocols supported
Item Delivery	FTP, Internet email/MIME, GEDI
Stream Delivery	MPEG/RTP
Security	PKIX / GSS, SSL, Java Access Control System
Payment	SET
Discovery	Z39.50
Order	ISO ILL
Customer	Z39.50, ISO ILL, email

Table 2. GAIA Extension Modules

This selection of protocols is shown in the following diagram which represents the full set of components developed for GAIA (i.e. both architectural and extension components).

5.3. Does the technical approach validate the architecture? (System Designer Perspective)

The primary aim of the GAIA architecture was to create an architecture that is generic and able to cover a wide range of required services and meet the brokerage requirements of a diverse range of domains. The technical approach taken allows the distribution of the basic components of the system and their interoperability regardless of the programming language used to code the underlying programs. Whether this architecture covers a wide range of the required services is something to be evaluated by the users and suppliers that participated in the trial and will be discussed later in this paper. Nevertheless the issue of applicability among diverse domains has been achieved, as the GAIA system has been successfully trailed in three diverse application domains.

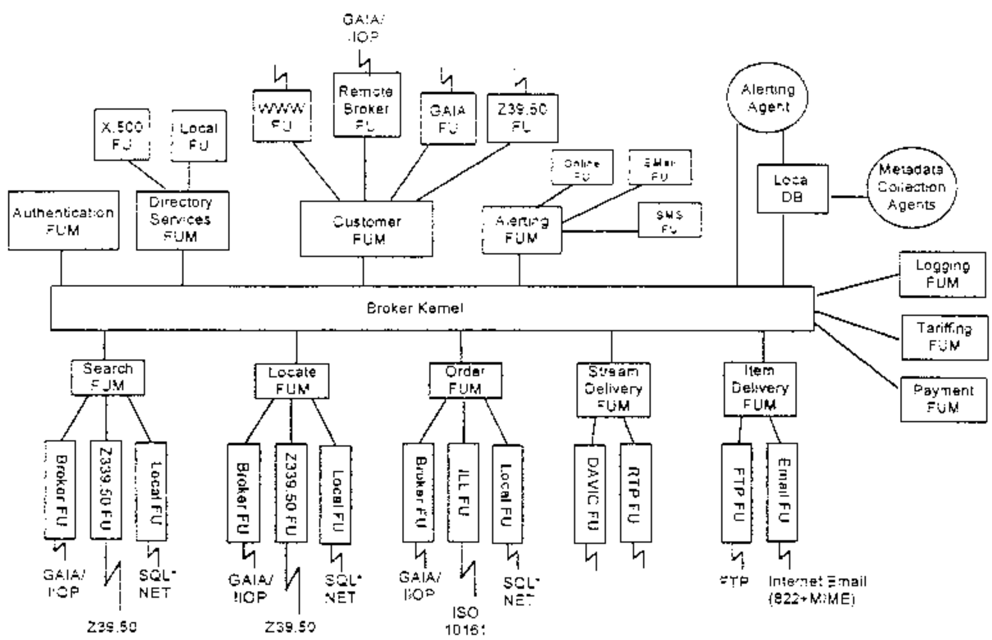

Figure 6: GAIA functional architecture and extension modules

The aim of scaleability has been achieved by the technical approach taken. The architectural foundation allows for the components of the GAIA broker to be distributed across the network. Additionally other organisations that wish to embrace the GAIA approach and develop components that comply with the GAIA specification can easily adopt this specification as it is defined in CORBA IDL. The same happens with suppliers that wish to include their system within the GAIA brokerage service. The supplier (or the owner of the brokerage service) can develop the required components according to the GAIA specification and use the dominant protocols used by the supplier's domain. This accounts for the commercial feasibility of the GAIA system. Suppliers don't need to configure other aspects of their system or take care of the issues that are already dealt by the GAIA system. This also implies that small suppliers have the opportunity to enter the GAIA brokerage service, as the owner of the service can address any technical difficulties they may have to enter the electronic market.

The heterogeneity of various supplier systems is dealt with by providing an implementation platform in a supplier and protocol independent manner. Then, as discussed earlier, suppliers can add their own customised protocols implemented as components (Functional Units in GAIA terminology). Within the GAIA project, an example of such an enhancement was the payment operation, which was enhanced with advanced on-line security features. GAIA developed an implementation of the SET payment protocol and the operation of SET based payment components was trailed in the GAIA demonstrators. The selection of these technologies and the parts of the transaction that have enhanced security features have been identified from the GAIA Domain and User Requirements Analysis [3].

Apart from these issues, there are a number of other issues that the GAIA system has targeted. These include the provision of a wide set of functions and usability. These issues, as well as some of the issues discussed above, are presented in the next section where the findings of the GAIA evaluation phase are overviewed.

6. Trials and Evaluation

The GAIA project had selected three target domains in which to demonstrate implementations of the developed GAIA compliant brokerage system. These are the Music domain, the Publishing domain - both with regard to Scientific Journals and to Statistical Data for Agriculture, and the Technical Data domain, with regard to electronic components. The span of these domain demonstrators, from the structured, highly specified, standards oriented, technical nature of the electronic components industry to the artistic, loosely structured, multimedia nature of the music industry, and the successful completion of the trials ensured that the GAIA architecture covers the needs of a wide array of other domains.

The GAIA evaluation was based on questionnaires (one per domain demonstrator) that could be accessed on line and covered various aspects of the system: usability, efficiency, functionality and benefits. Evaluation of the system was not an obligatory task for users. The users could either choose anonymity or provide the GAIA evaluation team with their personal details. Users could also provide their own comments in certain issues beyond the questionnaire structure. An evaluation based on questionnaires is certainly restrictive, as the questionnaire defines the overall structure, items of evaluation and phrasing of the questions. However the anonymity of evaluators contributes towards the honesty of answers, while the fact that users could fill their own comments throughout the questionnaire contributed to the flexibility of the evaluation process. Moreover the evaluators were the actual users of the system and certainly not a selected subset, but the subset that wished to provide us with their opinions. The GAIA demonstrators still operate [3] and the evaluation of the GAIA system continues up until now.

In the following sections we describe the evaluation results in light of the GAIA aims as described in section 2.

6.2. User Evaluation

During the main evaluation phase of the project, which spanned the two months after the start of commercial operation, about 200 users sent their evaluations to the GAIA evaluation team. These users were from over 50 organisations, companies and universities. Their background and opinions certainly varied a lot. However there are some general conclusions we can draw from the study of these answers with relevance to the key issues of usability, efficiency, functionality and benefits.

As described above, the GAIA architecture is independent of the design of the user interface of the brokerage system. However it was soon realised that the usability of the

GAIA system would be a key issue to the successful operation of the brokerage system. The demonstrators also validated this. Specific issues related to the usability of the user interface that users have reported were:

- lack of understanding concerning the context of some GAIA facilities, especially alerting;
- lack of specific customisation features of the user interface;
- difficulties during interaction with the system in specific phases (different per demonstrator user interface);

The GAIA user interface is rich in functionality, as it allows the user to access a variety of functions from one single point. Additionally the GAIA system is targeting to Internet users, which vary a lot with regard to their specific preferences and characteristics. On the other hand current Internet - based (at least) user interfaces suffer from the lack of mechanisms that would shield the user from information load, while they are totally passive and non-interactive. GAIA has developed, apart from standard (based on HTML and Javascript) user interfaces, a Java client that allows users to access the GAIA system via a user interface that offers more customisation and interactive features. This approach, despite the fact that it provided a more interactive and customisable user interface still didn't solve some of the above mentioned problems: those that are related to understanding the context of some facilities. Additionally it created other problems that had to do with the system response and efficiency: the Java client connection to the broker machine was very slow especially for users that were connected to the Internet via dial-up. Based on these results the GAIA partners are now developing an interaction agent [2] that provides assistance to users of the GAIA system towards both manipulating the user interface and proposing ideas and options with regard to the domain of application.

Nearly every user that participated in the evaluation received the level of functionality offered by the GAIA demonstrators very well. The vast majority of users reported that they were satisfied by the provided facilities. Very few were unsure if the set of facilities met completely their needs but there were no suggestions of functions that would be required in order to enhance the richness of the service.

The implementation of some functions could be more efficient in some cases and users, especially with regard to the publishing demonstrator reported this. In the case of the publishing trial, the scientific publisher that offered the content dictated that the articles had to be faxed to the customer. This detail is very important in order to realise the difficulties of coping with the different business models that suppliers use to communicate with their customers. Trial users would have preferred an electronic version of the article and several identified that the quality of fax delivery, especially within diagrams was not suitable. Traditionally publishers have been reluctant to make electronic versions of their product available to the customer. The portability and ease of reading of the printed version and the compactness and searchability of the electronic version show that there is a role and profit potential for the publisher to produce both media. Additionally electronic versions of the product open the possibility of authors producing multimedia version of articles. This is particularly applicable in scientific and medical articles where text can be complimented with video and animation.

Another interesting case of examining the efficiency of a function implementation was that of payment in the technical data demonstrator. The product in this case was the technical data descriptions and specifications in electronic formats suitable for designers' CAD systems. The existing payment scenario in this domain has been traditionally based on the subscription of a customer for a given period of time: the customer paid the supplier at registration and could make unlimited use of the service resources. However this has not proven a successful practice in the technical data domain, as there are some manufacturers that offer specifications for free. Thus the implementation of the payment in the technical data demonstrator demonstrated both payment by subscription and payment by item in order to adapt to both suppliers' business models and customers' requirements.

Possibly the most important aspect of the evaluation of the brokerage service was to identify the benefits that this service offered to users. It was seen that users found the service beneficial especially due to its availability, quality and completeness, while they were more sceptical on issues related to their own financial benefits and to the overall function of the commercial chain.

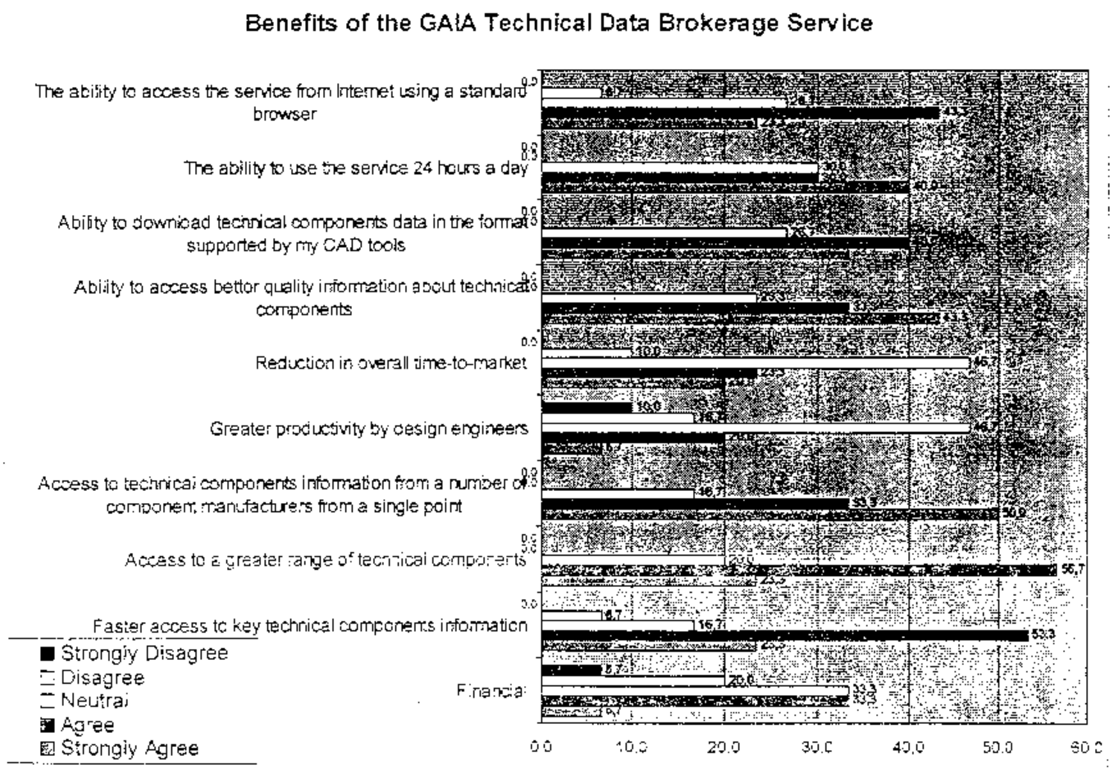

Figure 7: Benefits of the GAIA Technical Data Brokerage Service

The above figure presents the answers that the users who participated in the technical data demonstrator evaluation had towards the question: «The following are important areas where I or my organisation can benefit from the use of the GAIA technical components brokerage service». The chart highlights the areas of brokerage service the users found most beneficial (i.e. «Access to technical components information from a number of component manufacturers, from a single point», «Access to a wider range of technical components»). It is interesting to note that these users were a bit sceptical with regard to any financial benefits of such a service as well as about the reduction in overall time - to – market and greater productivity. This scepticism could be attributed to the fact that the test users tended to be design engineers and technicians rather than production engineers or sales/marketing people.

6.3. Supplier Evaluation

The suppliers that participated in GAIA had diverse differences with regard to their position in the market and their previous involvement in electronic commerce. Thus the motives towards participation in the project, their expectations and estimation of results varied a lot.

In the publishing domain, the supplier was a leading scientific publishing company with great experience in electronic commerce. The motivation for this company was twofold, first to investigate the financial benefits of brokerage services and second to investigate the new and unique electronic commerce services, which were potentially possible. The publishing demonstrator service was therefore commercial from the beginning of the evaluation period (in contrast to the other two demonstrators) and at the two - month period of operation there were about 100 users that used the service. This was considered a satisfactory number of users, especially if we take into account that these were the first two months of operation.

In the music domain, following extensive discussions, six independent labels from the European Free Improvisation genre agreed to join GAIA. The discussions ranged not so much around the potential of Internet shopping for small labels, which was generally accepted as a concept (most participants by the time of the trial were on e-mail), but more around the precarious economic balancing act faced by the labels and the effects that using an intermediary might have on this. Thus, the primary reason for the labels becoming involved in GAIA was to test out e-commerce in a hopefully supportive environment associated with like-minded people rather than risk being swamped and lost through associations with major labels. The labels also saw it as being an extension of their association with the European Free Improvisation Web site.

In the technical data domain, the supplier (the term is used according to the GAIA reference model) was a real – life broker: a major English company that commercialises technical data from more than 100 manufacturers. This company also had a large degree of experience in applying electronic commerce business models of operation. The motivations were mostly business oriented. Additionally the GAIA technical approach was of particular interest to this company. There were no commercial benefits from the technical data demonstrator since it was decided that the service would run free of charge in order to investigate the potential of the Greek market. However the influence in the company's business operations and the knowledge transfer were two major benefits to this supplier.

The diversity of the characteristics of suppliers that participated in the GAIA demonstrator proved that one major objective of brokerage services was met. That is to ensure a non monopolistic electronic commerce environment by allowing fair competition for both large and small companies, allowing quality and completeness of service to be market differentiators and promoting cultural diversity.

7. Conclusions

This paper presented the architectural and technical approach of the ACTS AC221 project GAIA (Generic Architecture for Information Availability), as well as an overview of the evaluation results of the projects in the demonstrator domains of scientific publishing, music and technical data.

The GAIA architectural approach was based on the definition of the reference model, which defined the actors, roles and basic operation functions among the broker and other stakeholders. In a lower level of detail the functional architecture defined the broker functions and protocol extensions. The key elements of the GAIA functional architecture are the GAIA kernel, Functional Unit Managers (FUM), Functional Units (FU), and abstract primitives (AP).

The technical approach was intended to provide a distributed and scaleable implementation of the brokerage architecture. Despite some development problems that were related to the immaturity of the technologies used, the use of CORBA and Java fully validated the development aims.

GAIA was demonstrated in three application domains: scientific publishing, music and technical data. The evaluation results from user and supplier perspectives were presented.

The short period of time during which the GAIA approach and system were trailed was not enough to provide the proof of concept. However the GAIA trials are still operational, results are still being collected, and the consortium partners still actively engaged in existing and new research projects related to the results of the GAIA electronic brokerage project and other electronic commerce areas.

8. References

[1] Hands J., Bessonov M., Smith R.. (1999) An Inclusive and Extensible Architecture for Electronic Brokerage, Proceedings of 32nd Hawaiian International Conference on System Sciences (HICSS 99).

[2] Koutsabasis P., Darzentas J. S., Spyrou T., Darzentas J., (1999) Facilitating User–System Interaction: the GAIA Interaction Agent, Proceedings of 32nd Hawaiian International Conference on System Sciences (HICSS 99).

[3] GAIA Web site: http://www.syspace.co.uk/GAIA

[4] IONA Technologies (1997) , Orbix 2 Reference Guide, IONA Technologies PCL, March 1997.