Migrating BDIFS from a Peer-to-Peer Design to a Grid Service Architecture

Tom Kirkham, Thomas Varsamidis

School of Informatics, UWB, Dean Street, Bangor, UK

Abstract. This paper documents the transition of a distributed Peer-to-Peer based business to business enterprise application integration framework, to one using Grid Services. In the context of an E-Business environment we examine the practical strengths of Grid Service development and implementation as opposed to Peer-to-Peer implementation. By exploring the weakness in the BDIFS Peer-to-Peer architecture and workflow we illustrate how we have improved the system using Grid Services. The final part of the paper documents the new Grid Service design and workflow; in particular the creation of the new automated trading mechanism within BDIFS.

1 Introduction

Business Data Integration Framework for the Small to Medium Enterprise (BDIFS) is a research project at the University of Wales Bangor designed to stimulate trade in local small business [1]. The BDIFS framework was founded as a method to encourage collaboration and information sharing at both a technical and commercial level between Small to Medium Enterprises (SME) in the Gwynedd region.

The original project's aim was focused on the integration of sites at a trial local business as part of a KTP partnership between the company and the university. It was hoped that the BDIFS system would build on the existing company IT framework within partners, which often lacked both technical resources and human expertise. The BDIFS system was therefore designed to be easy to use and also computationally lightweight, to address these needs Peer-to-Peer software design was chosen.

The implementation of BDIFS as a Peer-to-Peer architecture has been achieved. Issues that arose during this process and during the testing of the framework have raised questions about the scalability and functionality of the Peer-to-Peer choice of BDIFS design. As a result the decision was made to move the system to a more robust and flexible architecture and software platform. In order to achieve this we have redesigned BDIFS using Grid Services.

Kirkham T. and Varsamidis T. (2006).
Migrating BDIFS from a Peer-to-Peer Design to a Grid Service Architecture.
In *Proceedings of the 1st International Workshop on Technologies for Collaborative Business Process Management*, pages 42-51

2 Related Work

Many larger multinational companies have advanced e-business systems that demand their smaller and larger partners to integrate in order to trade. Integration of different business's, with varying data formats, business logic and even national law is the focus of academic work in various forms of computing. Many of which are focused on specific integration issues, or the creation of more general pieces of software designed to join systems in traditional client server formats [2].

The area of distributed computing seems to lend itself to addressing the issues relating to this form of integration being adopted in various forms. Distributed computing approaches allow the focus of development to be on more specific areas of the integration process. Methods to achieve this type of integration has often included either include lightweight Peer-to-Peer messaging approaches [3] or larger and more general Grid Service Middleware design [4, 5]. This latter is often achieved by exposing existing applications as services using open standards. The development of the OGSA and WSRF standards present a standardized methodology in order to achieve this type of integration [6, 7].

However despite the concepts and design methodologies being available few examples of integration using distributed systems outside of the propriety software domain can be seen. It can be argued that this is because the semantics of the actual e-business information integration is a tough area for Grid Middleware and Peer-to-Peer system designers to approach. As the integration of data from businesses often involves the orchestration of a wide and varied range of file formats and business logic, which makes the mapping and integration a fragile, complex and time consuming task. [8, 9].

This complexity therefore has influenced the growth of integration around Enterprise Application Integration (EAI) architectures provided by specific vendors. A good example of this type of application is the integration provided on an EAI level by the Microsoft BizTalk server [14]. A server that essentially provides the ability to map and manage messaging between distributed systems, the server is hosted on the company LAN. BDIFS Building on the concept of Business Service Networks [10] is attempting to design a solution for the SME. The BDIFS framework was designed to separate the messaging from the data mapping knowledge and experience that is shared in the framework. By initially using a simple Peer-to-Peer messaging framework it was the initial aim of the project to mirror the EAI mapping and messaging functionality for the SME. Presenting a solution that had its main functionality and therefore support overhead away from the LAN of the SME. Although experience has shown that the BDIFS model in order to appeal, has to develop its messaging framework to support a richer range of scenarios.

3 BDIFS Aims

The main target of BDIFS is the SME, within North Wales many of these SME's are under resourced in terms of IT skills and knowledge, yet are under huge pressure to integrate. This factor is exposed when the SME has to collaborate and integrate with a trading partner. Our first prototype for BDIFS was order processing, this simple messaging cycle involved a process that could be expressed in simple XML and is event driven. We selected this scenario on both its simplicity and also because it was a common integration scenario many businesses in the area were struggling to adopt. For example we found it common that electronic orders were received by suppliers and then keyed manually into the software on site.

Many SME's visited in the initial research of the project displayed examples of reliance on proprietary technologies. These pieces of software, databases etc, depend on specialist consultancy for upgrades and customization. It was the aim of BDIFS by focusing on using open standards and technologies, future development and upgrades would be not such a specialized task. It was hoped that BDIFS would influence an adoption of a core set of Open source projects by the SME. The key elements of the BDIFS Peer-to-Peer design were founded on the aims below.

3.1 Standardization

In BDIFS communication between trading partners has to be well defined and structured. Therefore we decided to implement our data format using the EBXML [11] standard. As simple messages in our prototype like purchase order and sales invoice can be expressed in EBXML files we decided to base our first implementation on an EBXML file sharing design.

The discovery, distribution and control of these files within BDIFS are achieved via the use of a common messaging platform. JXTA technology was the chosen method to implement this platform and has provided the system with an open and well defined platform for EBXML file distribution. JXTA provides the developer with a clearly defined security architecture and peer design structure.

Therefore by selecting EBXML and the Project JXTA as the core technologies we ensured the BDIFS platform was based on both open standards and well defined technology for that type of application. It was hope that this common base would also encourage other partners wishing to integrate to express transaction data in non proprietary formats such as EBXML.

3.2 Collaboration

BDIFS takes care of the transport and security involved in the exchange of data between partners, whilst standardizing the delivery format of the data. However the issue of legacy integration still remains in the architecture. The data within the BDIFS system initiates from Business Information Systems of different varieties at partners sites. This presents the integration architecture with the issues related to the integration of data in differing formats and contexts. Whilst BDIFS boundary can be seen to stop at the legacy system producing the data, we aim to influence the extraction of data from these systems via BDIFS.

The first influence we have is that the data needs to be presented to BDIFS either as EBXML or when receiving translated from EBXML. For partners these two translations can be seen as the pre-requisites of membership of the BDIFS framework.

The translation to EBXML is also something that can be supported in BDIFS. By providing access to information on the BDIFS website about translating and expressing data as EBXML, partner knowledge is logged, whilst integration tips and experiences are documented.

It is hoped that this knowledge will develop to be a comprehensive source for integration based on open technologies, like the ones supported in BDIFS. This information is presented on the web forms part of a BDIFS central portal. This is where the BDIFS partners join the system, in the Peer-to-Peer design; Peers are downloaded from the site once registration is completed (a Peer can be seen as essentially an agent that the partner uses to communicate with the BDIFS framework). It is hoped that this practical set of examples supported by documentation will help develop a valuable resource, for translation, mapping and also the discussion of integration issues.

3.3 Simplicity

Finally BDIFS was created using the JXTA Peer-to-Peer development platform [12]. This supplied a lightweight and secure messaging system capable of running on low bandwidths and computers with few processing resources. Once the EBXML is created, JXTA clients can share data automatically as files. Security, routing and also logging is built into the system in its internal infrastructure, core security services are provided by the JXTA platform.

JXTA provides a security mechanism for the system, this is essentially based on Peer and Peer Group ID which the BDIFS server generates in order to identify individual Peers and sets of Peers. A Peer Group in Jxta is defined by a key associated with all Peers generated in BDIFS. Peers that have the same Peer Group key therefore belong to the same group. The BDIFS portal manipulates the key ID's in order to pair partners. For example when Partner A joins, he registers with the Portal and downloads a Peer with a unique Peer ID and Peer Group ID. If a Partner B wishes to trade with Partner A, he or she must also go to the portal and download a Peer. By specifying what existing BDIFS partners the partner wishes to trade with allows the portal to produce a Peer for download belonging to the same Peer Group (by sharing the same Peer group ID). The data flow in a typical BDIFS business to business messaging exchange can be seen in Figure 1.

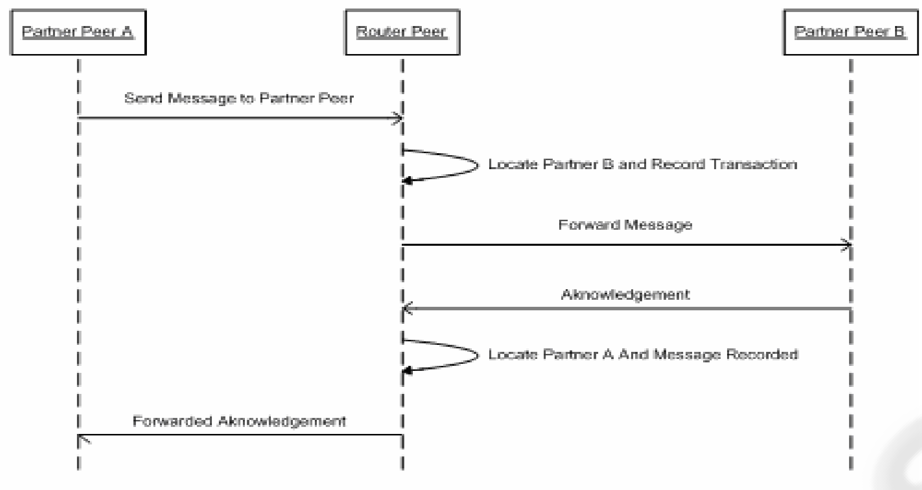

Fig. 1. BDIFS Peer-to-Peer example workflow.

Within figure 1 the simple exchange of a message is illustrated. The Peers belonging to both Partner A and Partner B are illustrated at either end of the sequence diagram and exist in the same Peer Group. The Router Peer is a core BDIFS system element and detects messages which are ready to be passed on by either Peer. The Router Peer also keeps a record of transactions in the framework and provides the mechanism by which data is collected from either Partner A or B and sent on.

4 BDIFS P2P Problems

Within the Peer-to-Peer design the framework's strength is in its simplicity, and also its collaborative spirit. However early tests using the architecture illustrated that it was very two dimensional. In particular users of the system began to request functionality which would require a redesign, for example the provisions of communications that were one to many, has raised issues in respect to thread control. As despite the Peer-to-Peer design having an ability to provide a method for sharing information in a rich and effective way, the BDIFS system had the potential to transmit old data or wrong data. This was particularly the case if Peers drifted on and off line. Whilst it is conceivable that these issues could be solved by adding increasing complexity to the Router Peer, this it can be argued would rapidly centralize a distributed system.

A key aim of BDIFS was process simplification and the increased functionality added to the central Peers in the system can be seen to remove this. Furthermore the models inflexibility in the use of any other method of data transfer other than EBXML can be seen to discourage commercial collaboration in the framework. As the ability of data to appear in native format within the system, is still a major desire for users of integration frameworks.

Finally a factor which further weakened the case for the BDIFS Peer-to-Peer framework was also the need for the provision of dynamic meeting of demand in the system. The Peer-to-Peer design was a delivery system which can be seen as based on simple message exchange. As with increased routing complexity, the addition of detailed decision making within the framework also threatened to move it away from its original aims. However failure to meet these demands, would fail to encourage the SME in the adoption of BDIFS and the main BDIFS aims of standardization and co-operation. To address these issues the decision was made to migrate the BDIFS framework to a Grid Service Architecture.

The Grid Service Architecture therefore would allow for central points of functionnality in the form of services to be added to the BDIFS framework. This would build upon the secure messaging platform presented in the Peer-to-Peer design. For example in BDIFS Grid Services data can be translated on route to a partner vis the use of a translation or mapping service. Essentially the means to accessing the framework for the SME would remain the same as the Peer would be replace by an agent. But the strength and extra dimensions and functionality is increased in BDIFS by the use of Grid Services.

5 BDIFS Grid Services

The BDIFS Grid Services were developed using the Globus Toolkit 4 (GT4) Java Core Services. The Globus Toolkit has been developed to provide the core infrastructure for designing WSRF and OGSI compliant Grid Services [13]. GT4 provides software libraries support the main infrastructure needed to construct a set of services in order to provide Grid Middleware applications that provide support for resource discovery, security and representation of state within the WSRF framework.

The BDIFS Grid Service Architecture like the Peer-to-Peer architecture has the same goals of standardization, collaboration and simplicity. However the Grid Service approach has dropped the sole use of EBXML file sharing, instead moving to a messaging system using BDIFS agents. These agents are designed to invoke remotely hosted core services which are available to all users on the BDIFS network. The core services each provide specific functionality within BDIFS and are a development on the concept of the core Routing Peer in the Peer-to-Peer design. For example security is managed in the Service architecture by an authorization and authentication service that uses a tokening system to check users.

At this stage the BDIFS Service framework is at a simple prototype stage, and the core services in the BDIFS system will now be explained. The core services make up the BDIFS virtual organization (VO). A VO can be seen to be the equivalent of the Peer Group concept, and is the domain in which the authenticated BDIFS agents and BDIFS instantiated services reside. Central to the BDIFS VO like many other Grid systems is the VO Manager. The BDIFS VO Manager contains information on users of the BDIFS systems, routes to core services and also supplied the authentication and authorization for the system. The VO Manager gains its membership information from the BDIFS portal which essentially has the same behavior as in the Peer-to-Peer design.

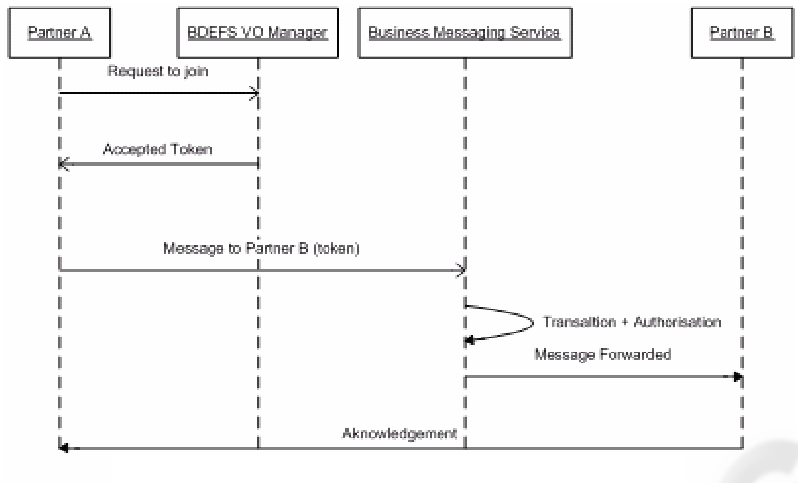

Fig. 2. BDIFS Grid Services Simple Messaging Workflow.

As illustrated in Figure 2, the Agent (Partner A) authenticates with the Base VO Manager and exchanges messages via the Business Messaging Service. The Business Messaging Service in BDIFS is used to translate the Agents call into business logic the target system can understand. In the Grid service model unlike in the Peer-to-Peer framework this can be to any format. This is to allow third party development of Business Messaging systems to ease integration on the target side. As the system gets more detailed these translations could begin to exist on specific services, such as *Format X* to *EBXML* service, rather than have them grouped on a single service.

6 BDIFS Market Place

To meet the desire for the creation of a market place within the system an additional service has been added to the BDIFS Grid Service Architecture. This service is set as a Monitoring Service that is designed to automatically start off a chain of events due to a change in state of a resource. This use and support of state is increasingly being the focus of the future development of Grid Services, thus placing their use as an advantage when opposed to standard web services or Peer-to-Peer frameworks. This is true in the respect of the adoption of the WSRF guidelines to handling statefull services supported in Globus Toolkit 4 complient Grid services and WSRF.net services.

At this stage in the development of the Grid Service version of BDIFS Market Place the Monitoring Service only demonstrates a simple workflow capable of automating a business transaction from beginning to end between partners. The process is invoked in a similar fashion to the diagram in figure 2. However the BDIFS VO manager by using a factory service creates a separate instance of the monitoring service for the agent requesting the service. This allows the service instance to be specific to the particular service requesting it, by allowing the Base VO to create a separate End Point Reference for the monitoring service and limit users accessing it to only the requesting party.

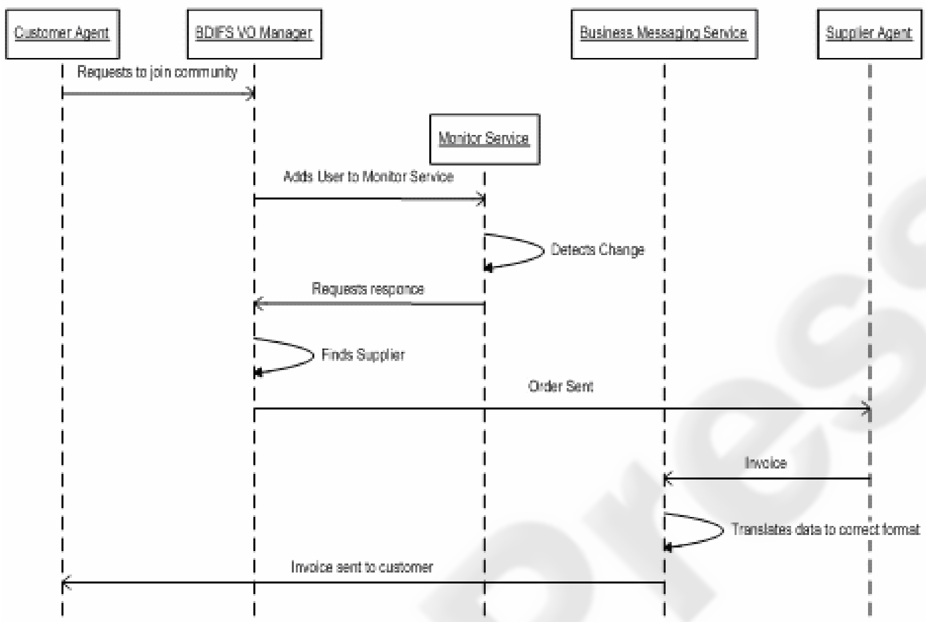

Fig. 3. BDIFS Grid Services Marketplace workflow.

The Monitoring Service is designed to monitor a statefull resource and trigger and alert if a specific change in the resources state occurs. In Figure 3 this process is started after the Partner registers for the specific service. It is envisaged that the Partner will provide at this stage a resource that the Monitoring Daemon will monitor. This resource would be presented as the Agent registering and is likely to be attached to a device such as a database. The Agent will be programmed to detect specific change in the data and notify the Monitoring Daemon. The result of which is the illustrated chain of events which involve a service being discovered to fulfill the need of the resource being monitored, in this case it is the Business Messaging Service.

7 Future Work

Initial plans for testing are aimed at internal business processors in a local SME. The aim is to simulate different trading partners between distributed company resources. The chosen area will involve communication between a manufacturing facility and the head office. It is aimed that once these tests are successful the BDIFS system can be further refined and targeted at the company's business partners.

This application will involve the monitoring service detecting that there is demand for a product at the central office. This will be done by the stock database being monitored as a staefull resource by the monitoring service. The threshold will be a certain level of stock that once reached will trigger an order for more stock to be sent to a specific agent. In Figure 3 this agent is discovered dynamically allowing for negotiation over price etc, however in our tests we plan to have one single agent present at the companies manufacturing facility. Thus the manufacturing facility will receive a purchase order via the BDIFS Agent; once the items are dispatched the Agent could then send an Invoice over to the Head Office Agent.

We aim to further enhance the market place ability of the system by adding more complexity to the simple workflow in Figure 3. Examples of such complexity could involve negotiation on price and quality etc within the Market place application. More services and Business Process specific services in particular are also under consideration for use. At the time of writing we are looking to add increased Business Process Functionality via the addition of a Business Process Enactment Language (BPEL) engine. Thus an initial evaluation of the system will be present in future papers presented about the system as we intend to start implementation this year.

8 Conclusion

The new BDIFS Grid Service architecture illustrates the strengths of Grid Service design over Peer-to-Peer development in an integration environment that demands scalability and increased workflow complexity. The BDIFS Grid Service development is being developed with the same goals of the Peer-to-Peer framework. In response to user needs the introduction of the Grid Service Marketplace in the model increases usefulness to the SME and larger business. A more detailed architecture will be presented in the near future.

References

1. Kirkham T, Varsamidis T. Introducing BDIFS, a new approach in business-to-business data integration and collaboration for the SME. Proceedings of EEE05 Las Vegas June 2005.
2. www.openeai.org

3. Ferreira D, Ferreira P. Building an e-marketplace on a peer-to-peer infrastructure. International Journal of Computer Integrated Manufacturing. Volume 17, Number 3 / April-May 2004.
4. www.lauraproject.net
5. Svirskas A, Roberts R. Distributed E-business Architecture for SME Communities - Requirements and Solutions for Request Based Virtual Organisations, IADIS International Conference - Web Based Communities 2005.
6. I Foster, C Kesselman, JM Nick, S Tuecke : Grid services for distributed system integration: Computer, 2002
7. I. Foster, C. Kesselman, J. Nick, S. Tuecke,: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration: Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
8. Medjahed B, Benatallah B, Bouguettaya A, Ngu A.H.H, Elmagarmid A. Business-to-business interactions: issues and enabling technologies. *The VLDB Journal* — The International Journal on Very Large Data Bases Volume 12, Issue 1 (May 2003) Pages: 59 – 85
9. Gosain S, Malhotra A, El Sawy O A, Chehade F. The impact of common E-Business Interfaces. Communications of the ACM Vol. 46 No 12 (Dec 2003)
10. Khare R, Tenenbaum J. Business Service Networks: Delivering the Promises of B2B. Proceedings of the IEEE EEE05 international workshop on Business services networks
11. http://www.EBXML.org
12. www.jxta.org
13. http://www.globus.org/toolkit/
14. http://www.microsoft.com/biztalk