RC22313 (W0201-101) January 21, 2002
Computer Science

IBM Research Report

e-Business Integration for an Enterprise: Selected Issues and Insights

Fredrick Wu , Rama Akkiraju, Terry Heath
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Delhi - Haifa - India - T. J. Watson - Tokyo - Zurich

1. Preface

Over the past couple of years, IBM Researchers have been working closely with IBM Software Group and Global Services teams in e-commerce and e-business integration related customer engagements. Research has two primary reasons for participation: first, to help IBM succeed in the marketplace by bringing researchers' insights and depth of knowledge directly to customer engagements; second, to bring back home some of the technical challenges that need further research and to pursue those research topics for further enhancement of IBM's software products. In this paper, we summarize our observations based on customer engagement experiences. Specifically, we talk about common e-business integration issues, integration challenges, how the IBM WebSphere product suite addresses some of these challenges, areas for improvement and finally suggested topics for future research.

Disclaimer: This is not an official paper on e-business integration or on e-commerce. Its scope is limited to the observations made by us (the authors of this paper) while working directly with a limited number of customers in the field.

2. Status

This is a Work-In-Progress document.

3. Acknowledgements

Thanks to Kumar Bhaskaran, David Flaxer, Henry Chang, Jen-Yao Chung, Richard Goodwin, Chung-Sheng Li, Liang-jie Zhang and Tian Chao for contributing to the insights mentioned in this document.

4. Executive Summary

Although the Internet offers tremendous promise to businesses trying to improve the management of their supply chains and to reach new suppliers and customers, the state of the technology has made it difficult to achieve these goals. The technical issues of enterprise integration that the industry has struggled with for over two decades have not gone away and have not been solved overnight. Businesses still need to laboriously integrate each and every application to their new e-commerce entity, whether it is a public or private exchange. While the information technology industry is forging ahead with new visions for e-commerce such as Web Services, UDDI, and ebXML, the slowing economy is forcing customers to focus their IT investments on phased implementations that are consistent with long-range trends but post quick returns. Over the past two years, some of us at Research have had the opportunity to participate and play key roles in executing e-marketplace engagements. We wish to share what have learned from our customer engagement experiences. Here is a quick summary of what we have learned and some recommendations for what we, the research and development community, can do in both the short run and the long run to improve the current state of our solution offerings.

In the Short Run we:

- **Need to Develop and Push Standards**: Strong standards remain the key to reducing confusion among customers and providers alike in the marketplace, and to avoiding duplication of effort. Technologists and industry partners need to move quickly to establish standards for conducting business on-line. Emerging standards such as ebXML, Web Services and UDDI are a good start but they need to be followed through to level of industry-specific detail required for practical implementations. In the absence of standards, the integration process will always be akin to reinventing the wheel. Although IBM is closely involved with standards bodies, as of now standards are lagging behind the demands of the IT industry.

- **Need Flexible Software Components Not Bulky Software**: Given that standards have not caught up with industry demands, we need to be ready to implement many types of e-business scenarios with flexible middleware components. Bulky, monolithic software products have proven difficult to sell, and require costly training for consultants and developers. Our middleware products need to be comprised of components that can work together or independently, resulting in a flexible set of tools that can meet the demands of the situation. Basing products on standards such as J2EE naturally leads to component-based software, but there remains significant scope for exploring how this can be done in general terms as well in the context of specific IBM products.

- **Streamline Software Products and Avoid Functional Duplication**: A major part of e-business integration involves integrating multiple legacy applications that potentially duplicate enterprise information. If, while addressing that problem, we propose a mix of software products that themselves have duplicate

functionality, we risk losing customers. We need to coordinate our middleware products better and avoid duplication in functionality. Basing the software on components helps to achieve this goal.

In the long run we:

- **Need to Explore Automatic Program-to-Program Communication**: Standards aim to ensure that all business partners' systems can talk to each other in a mutually understood language. However, it is unrealistic to assume that this will apply to all present and future scenarios. There will always be business critical applications that cannot be made to comply with the standards for some reason. An ideal way to solve to this problem in the long run is to enable software programs to communicate with each other automatically. A software application run by one business could search the Internet for a suitable business partner (running a similar software application that represents the partner), initiate negotiation terms and conduct business automatically, while being monitored by humans representing both companies. How is this different from what the eBXML standard proposes? The difference is that in this case, no specific standards are imposed on any business partner, only some broad rules. Each software application representing a business partner is written and is represented with no restrictions. The programs have the intelligence to ascertain and perform the necessary mappings internally to communicate with each other. This would be an application of Artificial Intelligence and its techniques such as semantic nets, machine learning etc.

- **Need to Develop Intelligent Value-Add Tools**: For enterprises, the promise of e-commerce is much greater than inter-company integration and cutting costs. In the long run, it is also about taking advantage of the Internet and exploring new business opportunities that could potentially improve the value of the enterprise's product or service offerings to their customers and thereby increase its revenues. This calls for exploration in the areas of:

    - *Economic Aspects of Information Economy*: How should enterprises restructure their product and service offerings (in terms of price discrimination, bundling etc.) in the new information economy?

    - *Decision Intelligence and Decision Support Tools*: While integrating with business partners is key to achieving efficiencies in the supply-chain, dynamically optimizing the operations by choosing the right service from the right partner at the right time could be at least as important for increased operations margins. This can only be achieved by investing in decision intelligence and decision support tools that help people sort out the best options from potentially hundreds and thousands of scenarios.

    - *Commercializing Value-Add Tools and Research Ideas*: How can IBM benefit from these kinds of research? In our opinion, IBM Global Services consulting division is an excellent vehicle for tying new research in with customer requirements because they are responsible for suggesting improved business strategies to their customers.

- **Need to Explore Autonomic Computing**: Paul Horn's (IBM Research's senior vice president) white paper on autonomic computing talks about this at some length (w3.research.ibm.com).

In the following sections, we discuss some of the items mentioned above in some detail.

5. Introduction

There is a broad consensus in the industry that electronic commerce brings efficiencies to business-to-business interaction by means of consolidation and aggregation of services and its potential to reach large numbers of potential business partners. One of the important aspects of bringing efficiencies to business-to-business enterprises is the need for these companies to integrate their applications with those of their partners in the marketplace. However, enterprise application integration is complex. Especially, without industry-wide standards for integration it is very difficult to achieve efficiencies in the process. On the one hand, the traditional, static, point-to-point communication model using proprietary communication systems between an enterprise and its business partners can no longer meet the emerging needs of e-commerce because of their inflexibility and maintenance inefficiencies. However, on the other hand, industry standards that can cover the entire gamut of the enterprise integration space are not ready for prime time yet. So, the enterprise integrators are often caught in this dilemma of how to design and implement enterprise integration software that is standard enough to help integrate many disparate systems while the potential new standards are still under debate and evolving? This dilemma poses significant challenges to the enterprise integration development community. In this paper, we address the general nature of such challenges from our customer engagement experiences, how we tackled those using IBM's Websphere suite of products that include Business Integrator (WBI), Secure way™ Policy Director™, MQ Series and more, what is missing and what we, the research and development community, can do to get there. We begin by explaining the general challenges in enterprise application integration.

6. A Common e-Business Enterprise Application Integration Scenario

A very typical scenario of e-business enterprise application integration would be as follows:

Company X is a reputed automobile company based in Germany. Company X has its operations worldwide and currently holds 40% of car sales in the European market and 5% in North America. Company X would like to embark on an IT project to integrate its suppliers and buyers more closely with its business operations to bring efficiencies to its supply chain process. In an effort to achieve that, company X would like to set up a private exchange to enable its suppliers and buyers to conduct business with it online. In the process, company X would like to consolidate the catalogs of all their suppliers using a third-party procurement software system in order to make it easy for its employees to do the procurement of required goods. It also would like to present a catalog view of its processed and finished goods to its buyers and link this private exchange with its existing ordering, payment and billing systems thus enabling its buyers to move all their purchases online.

Company X has the following existing assets that must to be made use of for moving operations online.

1. *In the past couple of years, the IT department of company X built a web-based procurement application (for exchanging request for quotes (RFQs) and quotes). Company X's customers have been using it for 1 year now to send RFQ's to its suppliers as part of procurement. Since this application is functional and is already in use, Company X would like to integrate it with its new private exchange to provide online RFQ capability.*
2. *Company X has an SAP enterprise resource planning (ERP) system for order management, payment and billing. It would like to integrate these systems with its new private exchange for order processing, billing and payment tracking.*
3. *Company X has just struck a business deal with a third party vendor X-Base for offering integrated catalog functionality. X-Base specializes in consolidating automobile industry catalogs. Company X would like to use X-Base's catalog software in the private exchange.*
4. *Company X has just struck a business deal with Azuba Inc. to use its procurement software for indirect procurement of automobile parts. Azuba Inc. is a market leader in procurement software. The private exchange has to be integrated with Azuba software for procurement.*

Each supplier and buyer of company X will be on-boarded to the private exchange via a secure registration process. Once registered, individual users from supplier and buyer organizations should be given role-based access control permissions to perform specific actions on the exchange. These role-based permissions on the private exchange should give users access to various applications such as procurement, order tracking, quote-provisioning etc. The private exchange will serve as the center for conducting all online business in a secure fashion. Company X estimates it will save n billion dollars using this private exchange in the first 3 years of its operation. Company X is soliciting proposals from the following IT consulting firms for this software development project….

Many of the e-business integration requirements are similar to the example given above with a few variations. In the section below, we enunciate some of the technical challenges in building this marketplace.

7. Selected High-Level Technical Issues

The scenario explained above sets the context for some of the technical challenges facing the enterprises, and the integrators in building electronic supply chains. We categorize the technical challenges in to three high-level types:

- *Data-oriented:* A large enterprise typically has many applications that are built at different times over many years. Each of these applications contains data that is required to support the running of a specific business unit or a process. This data could contain information about the enterprises' business partners, users belonging to those companies, their profiles, their roles, their current state of business with the enterprise (this includes their contracts, legal agreements, payment terms, credit history, purchase orders etc.). This information is typically spread out across many applications at times with significant amount of duplication. One of the key challenges in building an electronic exchange (public or private) is to consolidate this information either physically or virtually in order

to facilitate overall data flow in the exchange across various applications that need to share this distributed information.

- *Business Process-oriented:* Business process execution is at the heart of bringing efficiencies to the supply-chain process. In a typical supply-chain, a business process could span across many supply-chain partners' applications. Managing a business process involves not only managing the state of the process by monitoring the execution at various levels but also dealing with rollbacks and error recovery all in a transparent way to the application that depends on this business process. There are some interesting challenges in this area and we discuss these in detail in the following sections.
- *And User Interaction-oriented:* Each application, that is brought in to the mix, typically brings its own user interaction models and presentation styles. Fundamentally different programming models such as Applets vs. JSPs, thin clients vs. thick clients, local access control vs. server-based access control etc. all make the overall user interaction integration of applications challenging. Because of these fundamental differences in programming models, even the seemingly simple task of integrating the styles sometimes becomes complex.

8. Broad Solution Strategy

A recurring theme in the proposed solutions to the categories of challenges mentioned above is whether to aggregate data and features by centralizing the management or to keep the information distributed and devise innovative ways of managing the distributed systems. While aggregation and centralization brings inherent advantages of ease of maintenance and extensions, total centralization is often very expensive and unrealistic in a business setting. On the other hand, while the initial set up costs for distributed, decentralized setup is much more affordable, in the long run the demands of constant maintenance could quickly turn into a nightmare for operations personnel in the absence of sophisticated self monitoring, and self manageable software. Unfortunately, there is no single approach that works best in all situations. The answer, in our experience, really depends on the given state and environment in which the applications reside, the time and money that are required to do it one way or the other. However, there is a middle ground, that is, a combination of moderate centralization and moderate autonomy to the applications landscape. In the following sections, we will discuss the implications of each of these approaches in the context of selected aspects of an electronic exchange such as member management, single sign-on, access control, application-to-application communication, business process management and user interaction experience.

9. Member Management

Legacy Application User Registries

Typically each of the applications that need to be integrated in an exchange or marketplace has its own user registry or directory. The application might use the user registry for security purposes, i.e. authentication and authorization, as well as for maintaining users' application data or workspaces. Authentication is commonly based on user ID and password. Since these applications are usually independent, this implies that a user of multiple applications could have a different user ID and password for each

application, in addition to different user profile data in each. This situation results in duplication of user data, high administrative maintenance costs (e.g. forgotten user ID and password), and inconsistent user data across applications.

From an integration point of view, it is clearly better to have a single centralized user registry that is shared by all the applications. Recent versions of many enterprise applications can be configured to use a shared LDAP directory. This greatly facilitates the task of integrating applications as far as user management is concerned and should be exploited when possible. But in general this solution cannot be work with all applications, especially legacy applications. When a shared registry is not an option, then the user data must be replicated across applications and synchronization mechanisms must be implemented.

Synchronization of User Registries

Synchronization of user data has two modes. First there is the difficult one-time task of initial coordination of registries. Since this is a bootstrap process, it can be handled manually by identifying the corresponding fields in each registry, selecting the "master" data source for each field, and writing scripts to transfer data between them. The second mode must keep the registries synchronized as entries are added, changed, or deleted. This process can be simplified if business rules are established and enforced that restrict the sources of changes to one or a small number of registries (e.g. a master registry), which could be dependent on the type of data. For example, one registry could be considered the master for user ID and password data, while another could be the master for user contact information such as phone number and e-mail address.

10. Single Sign-On

Closely related to User Management integration is the concept of Single Sign-On (SSO). SSO is the external manifestation of the integration of users across applications in the exchange, in that it supports the illusion of logging in to a single system. When a registered member of an exchange visits the site, he/she is generally required to log in before certain functions can be accessed. If the user wishes to perform multiple functions that require access to different applications during the same session, he/she should not be required to log in separately to each application, both for the sake of convenience as well as to give the appearance of a single unified Website.

Single Sign-On vs. Global Sign-On

The design and implementation of SSO is dependent on the degree of integration of the registries of the various backend applications. If the user's accounts have different user IDs in the applications, this requires mapping of the single external user ID that the user will use to log in to the exchange to the particular IDs that are associated with the user's accounts in each application. The term Global Sign-On (GSO) is sometimes used to refer to this situation, and the term SSO to refer to the situation in which each user has the same user ID in all applications (for which he/she is authorized). Furthermore, it may be impractical to disable the authentication mechanism in some legacy applications. In that case, for GSO not only the user ID but also the user's password or other authentication

credential needs to be mapped to the external user ID; this is known as *credential mapping*.

Implementing SSO

There are three main steps involved in implementing SSO:

- Disable authentication in each of the backend applications.
- Establish a common authentication credential, e.g. a cookie that contains the user ID.
- Implement a single point of authentication, preferably outside the Trusted Zone, where the applications reside.

Disabling application-level authentication prevents additional login challenges after the user has logged in to the exchange (for the duration of the user's session). Establishing a common credential typically requires minor customization of the application so that it will read the credential and extract the user ID from it. The credential should be digitally signed and the application needs to access shared keys to verify that the credential is from a trusted source. Finally, the single point of authentication is responsible for guaranteeing the authenticity of the user by whatever means (user ID/password or certificate). This is commonly performed by a proxy server located in the demilitarized zone (DMZ).

11. Access Control

An exchange can have many different kinds of users representing different companies, and can hold confidential data that must be available only to designated users. It can also offer functionality that can have far-reaching effects involving large sums of money, such as placement of purchase orders or transfer of funds. Resources such as data and operations on data must be protected from unauthorized users, yet easily available to authorized users.

Centralized Access Control

When applications are being integrated in an exchange, there is a natural separation of data and functionality between applications. Since the applications are independent by nature, an external centralized authorization service is needed to control which users can access each application. This centralized authorization service should make use of an LDAP directory of users that is shared with as many of the applications as possible. A centralized service greatly eases the maintenance of User Management, providing a common repository and user interface for both authentication and authorization data. If the access control model is role-based, [could include a reference here?] then a centralized system can best represent roles that define permissions across multiple applications. However, frequently it is required to control access to application-specific resources that are not easily represented in a centralized system. Furthermore, representing application-specific resources in a centralized service can mean significant replication of data, with the concomitant synchronization issues.

Distributed Access Control

For all but the simplest of applications, it makes sense to define a boundary between the centralized and application-specific resources. In other words, the centralized service can

manage and enforce access controls to the applications themselves or to some coarse level within the applications, but the fine-grained access controls are left up to the applications to manage and enforce. Distributed access control can still be consistent with a shared LDAP user registry, and the registry could be customized to store application-specific user attributes, some of which may be needed for application-level access controls. An application may even have its own role-based authorization scheme that can exist independently of the exchange-wide roles.

12. Application-to-Application Communication

Two or more business applications that need to communicate with each other can do so in various ways depending on whether it is an intra-enterprise or inter-enterprise communication. They can communicate in synchronous mode or in asynchronous mode. In synchronous mode, they can choose to make object level invocations or choose a standards based invocation either using Http or SOAP protocols. In an asynchronous mode, they can choose to send messages via a message queue or via file transfer etc. They can choose to exchange messages directly with each other or go through a central routing hub that redirects the requests appropriately. They can all agree upon a standard message format and language or they can each choose to send and receive messages in their own proprietary languages thus necessitating a central hub to perform the required translations before forwarding the messages to the requested destinations. Each of these approaches has significant impact on the effectiveness of the overall solution in development, maintainability and performance. Certain modes of communication suits certain types of interaction. This has to be carefully chosen to best fit the requirements of the overall solution. Below, we outline some of the key aspects of application-to-application communication.

Service Level Agreements (SLAs)

A service-level agreement is a contract that defines the technical support or business parameters that an application service provider or other IT outsourcing firm will provide its clients. The agreement typically spells out measures for performance and consequences for failure. This concept can be extended for a contract between two business applications that are involved in the execution of a business process within or outside a company. For example, if application A sends a query to application B, a service level agreement could say that application B needs to respond to the queries from application A within a certain period of time. Sometimes, the agreement could also state that application A would no longer be obliged to accept a response from application B if it arrives after the agreed upon time has expired. Service level agreements should be defined as clearly and in as much detail as possible in order to ensure all possible scenarios are covered in a message.

Business Communication Protocols

A business protocol is a set of pre-agreed upon rules that the participating applications will follow to communicate with each other. For example, if application A needs to send a message to application B, a message communication protocol could be followed. This protocol would define details such as:

- What format the message would be exchanged in

- What transport protocol to use to send and receive the messages
- What to do when sending a message fails
- How many times to re-send the message
- Whether to send an acknowledgement for a message or not, etc.

Transaction Authentication
When a source application needs to communicate with a target application, very often it is a good idea for the target application to ensure that the source application is a trusted party and that it is authenticated. If the applications are communicating on a trusted network such as a private or a virtual private network, then the target applications perhaps can just perform authentication and be satisfied with it. On the other hand, if the applications communicate with each other on an open network such as the Internet, then the target application perhaps should not only be authenticated, but also authorized by means of either digital certificates or public-key-private-key infrastructure services.

Trading Partner Profiles
When a source application needs to communicate with a target application, the source and target need to know each other's capabilities for sending and receiving messages. For example, they need to know what transport protocols the partner can support, what message types they are allowed to exchange, etc. This pre-negotiated information can be captured in trading partner profiles.

Transaction Tracing
All the transactions need to be traceable for various reasons including audit logging, exception management and help desk support etc.

Transaction Error Handling
Many things could go wrong in the process of communication and at times the communication may not happen for various reasons. A well defined set of error handling mechanisms have to be defined in cases where the communication fails. Errors could occur due to several reasons and could classified accordingly. The types of errors that the system could deal with include:

- Server communication error
- Duplicate message error
- Unknown message type error
- Sender unidentified error.

Transaction Monitoring: Finally, in an overall application architecture it is a good idea to have a transaction monitoring system that will enable system administrators or users with permissions to monitor the transactions going back and forth between the source and target applications. This transaction monitoring system could also provide a user interface for the users and can in turn be used as a help desk tool as well.

In summary, whenever a transaction cuts across multiple applications, the responsibility for the overall application architecture increases because it needs to manage the integrity and traceability of the transactions.

Websphere Business Integrator addresses many of the above-mentioned aspects in its Information Delivery Manager module [2].

However, there are still several challenges in this area. There has been much confusion in the industry in the past few years on how to address these items in a standardized way. Standards are evolving for each of these key aspects of enterprise application communication (eg: eBXML). However, standardization of all aspects of communication is only part of the overall solution. What about configuring, managing, and adapting these protocols, agreements, and profiles in a dynamic business environment? We talk about these challenges more in depth in 'Potential Future Research Items' section.

13. User Interaction Experience

Another very important aspect of integrating multiple existing applications into a unified electronic exchange is the user experience. Typically each application has its own user interaction model and presentation style. Fundamentally different programming models, such as applets vs. JSPs, thin clients vs. thick clients, local vs. server-based access control etc., all make the integration of user interaction challenging.

The notion of centralized vs. distributed management applies here as well. At extreme of complete centralization, all application screens are re-implemented in the selected style of the exchange, and the existence of separate backend applications that actually respond to the user is completely hidden from the user. At the other extreme, the user directly accesses the URL of each application (although perhaps through a proxy server for centralized authentication), with no customization of the applications' user interfaces. In practice most implementations will be in between the extremes.

A very basic feature that centralization can offer is personalization of the user's "desktop," i.e. customization of the workspace shown to the user based on the user's roles. In an exchange that offers functionality from multiple applications, usually some users will not be permitted access to some of the applications, as discussed in the section on access control. However, it is highly desirable to personalize the user's screens to show links to only those applications that the user is authorized to access. To do this, the user's navigation links should be created dynamically, by querying the access control system with the user's identity. Once the user selects a link to an application, he/she is brought to the application's user interface (which in turn could be personalized by the application based on the user's application-specific roles).

The degree of user interface integration is clearly a tradeoff between the benefits of presenting a completely unified interface to all the applications and cost of modifying the applications' screens. At a minimum, most engagements will require a top-level role-based desktop.

14. Business Process Management

Within an enterprise, a complex business process is often executed through interactions among many entities such as internal and external business applications and internal and external human participants. Deploying software systems that can help coordinate and manage these interactions and information flows is the essence of building an electronic exchange from an enterprises' point of view. For example, a purchase order sent by a buyer goes through inventory check, credit verification, approval, billing and payment processing etc. before the buyer is sent an acknowledgement. Each of the actions is typically performed by a different enterprise application. Coordinating the activities of all these systems in the context of processing a purchase order request from a buyer is the job of a business process manager. Ideally, it knows how to handle conditional processing at each stage in the business process.
Websphere's Business Integrator's Business Flow Manager (now known as Process Broker Service) addresses this issue in conjunction with MQ Workflow. More details about this can be obtained from WBI documentation [2].

How centralized should a business process manager be? How to capture dynamic flow (where the path to take is dependent on the outcome of a previous stage) and manage such a dynamic flow are some of the interesting aspects that are worth considering in the context of a dynamic e-business setting.

15. IBM's WebSphere Suite of Products

IBM's WebSphere suite offers a good range of software products for enterprise e-commerce. It offers components from building base portals (WebSphere Portal Server) with proxy servers (WebSphere Edge Server) all the way to building sell-side e-marketplaces (WebSphere Commerce Sell-side), buy-side e-marketplaces (WebSphere Commerce Buy-Side) and end-to-end business process integration and management (WebSphere Business Integrator). Some of the more complex software products build on top many of IBM's middleware software products such as Policy Director, Web Traffic Express, MQ Series (MQAO), MQ Series Integrator (MQSI), and MQ Workflow (MQWF). More details on each of these products can be obtained from http://www.ibm.com.

Shortcomings

- *Duplication*: A quick look at WebSphere suite of products indicates that there is significant overlap in the functionality that many of these software products offer.
- *Complexity*: One of the main disadvantages of bringing in many middleware software products together into one framework is that it increases the complexity of the overall solution tremendously.
- *Lack of Integrated Development Environment*: When many products are brought together, each has its own build-time and run-time tools and offers a different look and feel. This makes the process of solution construction quite laborious because a developer has to constantly switch between development environments and load and unload updates to do the job.
- *Skills & Training Issue*: Lack of an integrated development environment combined with the need to learn many software tools to build a solution construct

add to the complexity of the solution. Therefore, it is becoming increasingly difficult for one person to know and manage all the systems. This stresses the services industry tremendously in terms of finding all the right skilled people for customer engagements.

16. Potential Future Research Items

While pointing out the challenges in enterprise application integration, we recognize the need for addressing these challenges effectively. IBM's products together with industry standards come a long way in addressing application integration. However, we are quite far from the 'holistic' view of autonomic computing in which "computing systems are capable of running themselves, adjusting to varying circumstances, and preparing their resources to handle most efficiently the workloads that we put on them". Currently, many of the tasks in business integration software such as installation, assembly, configuration, monitoring, problem identification, debugging, and problem recovery are all manual and need constant maintenance. Moreover, the services nature of the IT industry due to its complexity is reducing the efficiency gains that could otherwise be obtained by optimizing the businesses. A highly available, scalable, self-configurable, easily maintainable, self-recoverable, self-healing system is not a futuristic requirement anymore. In fact, many of IT customers have been asking for such systems for quite some time now. The requirements are well defined. It's time to make it happen. In this section, we propose some broad potential approaches that we could pursue in the immediate future to further the state-of-the-art in this area.

Please refer to the 'Executive Summary' section of this paper for our recommended research topics.

17. References

1. Autonomic computing: IBM's Perspective on the Sate of Information Technology by Paul Horn IBM Research, http://www.research.ibm.com/autonomic
2. IBM Websphere Business Integrator, http://www-4.ibm.com/software/webservers/btobintegrator/
3. UDDI specification, http://www.uddi.org
4. E-business XML (ebML) http://www.ebxml.org
5. Rosettanet, http://www.rosettanet.org
6. Decision Support for Managing an Electronic Supply Chain, Pinar Kesinocak, Richard Goodwin, Fred Wu, Rama Akkiraju, Sesh Murthy in Electronic Commerce Research Journal 2001 1: 15-31
7. Dynamic Pricing by Software Agents, Jeff Kephart, James Hanson, Amy Greenwald Computer Networks 2000.