



NRC Publications Archive Archives des publications du CNRC

Multi-model driven collaborative development platform for service-oriented e-business systems

Li, Y.; Shen, J.; Huang, Y.; Shen, W.; Ghenniwa, H.; Xu, Y.

This publication could be one of several versions: author's original, accepted manuscript or the publisher's version. / La version de cette publication peut être l'une des suivantes : la version prépublication de l'auteur, la version acceptée du manuscrit ou la version de l'éditeur.

For the publisher's version, please access the DOI link below. / Pour consulter la version de l'éditeur, utilisez le lien DOI ci-dessous.

Publisher's version / Version de l'éditeur:

<https://doi.org/10.1016/j.aei.2007.09.005>

Advanced Engineering Informatics, 22, 3, pp. 328-339, 2008-07-01

NRC Publications Record / Notice d'Archives des publications de CNRC:

<https://nrc-publications.canada.ca/eng/view/object/?id=68890260-80ba-4478-97a3-f1b170245197>

<https://publications-cnrc.canada.ca/fra/voir/objet/?id=68890260-80ba-4478-97a3-f1b170245197>

Access and use of this website and the material on it are subject to the Terms and Conditions set forth at

<https://nrc-publications.canada.ca/eng/copyright>

READ THESE TERMS AND CONDITIONS CAREFULLY BEFORE USING THIS WEBSITE.

L'accès à ce site Web et l'utilisation de son contenu sont assujettis aux conditions présentées dans le site

<https://publications-cnrc.canada.ca/fra/droits>

LISEZ CES CONDITIONS ATTENTIVEMENT AVANT D'UTILISER CE SITE WEB.

Questions? Contact the NRC Publications Archive team at

PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca. If you wish to email the authors directly, please see the first page of the publication for their contact information.

Vous avez des questions? Nous pouvons vous aider. Pour communiquer directement avec un auteur, consultez la première page de la revue dans laquelle son article a été publié afin de trouver ses coordonnées. Si vous n'arrivez pas à les repérer, communiquez avec nous à PublicationsArchive-ArchivesPublications@nrc-cnrc.gc.ca.



National Research
Council Canada

Conseil national de
recherches Canada

Canada



<http://irc.nrc-cnrc.gc.ca>

Multi-model driven collaborative development platform
for service-oriented e-business systems

NRCC-50520

Li, Y.; Shen, J.; Huang, Y.; Shen, W.;
Ghenniwa, H.; Xu, Y.

July 2008

A version of this document is published in / Une version de ce document se trouve dans:
International Journal of Advanced Engineering Informatics, v. 22, no. 3, 2008, pp 328-339

The material in this document is covered by the provisions of the Copyright Act, by Canadian laws, policies, regulations and international agreements. Such provisions serve to identify the information source and, in specific instances, to prohibit reproduction of materials without written permission. For more information visit <http://laws.justice.gc.ca/en/showtdm/cs/C-42>

Les renseignements dans ce document sont protégés par la Loi sur le droit d'auteur, par les lois, les politiques et les règlements du Canada et des accords internationaux. Ces dispositions permettent d'identifier la source de l'information et, dans certains cas, d'interdire la copie de documents sans permission écrite. Pour obtenir de plus amples renseignements : <http://lois.justice.gc.ca/fr/showtdm/cs/C-42>



National Research
Council Canada

Conseil national
de recherches Canada

Canada

Multi-Model Driven Collaborative Development Platform for Service-Oriented e-Business Systems

Yinsheng Li¹, Jianping Shen¹, Ying Huang², Weiming Shen³,
Hamada Ghenniwa⁴, Yingxiao Xu^{1*}

¹*Software School, Fudan University, Shanghai, P.R. China*
liys@fudan.edu.cn

²*IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA*
yhx@us.ibm.com

³*Institute for Research in Construction, National Research Council Canada*
800 Collip Circle, London, Ontario, Canada, N6G 4X8
weiming.shen@nrc.gc.ca

⁴*Dept. of Electrical and Computer Engineering, University of Western Ontario*
London, Ontario Canada, N6A 5B8
hghenniwa@eng.uwo.ca

Abstract

This paper proposes a multi-model driven collaborative development platform for service-oriented e-Business systems. The platform provides engineers/consultants with three views, i.e., business view, process view, and service view to support service-oriented software engineering, top-down business design and bottom-up service-oriented system development. The views are kept consistent through model-driven translation mechanisms. The platform employs three models, i.e., service meta-model, process model and business model to implement the translation. On the platform, business and technical consultants/engineers can use the designated views for their roles to collaborate for a service-oriented e-business system based at the distributed sites of, e.g., IT vendors and their clients. The collaboration is featured with rapid development and visual demonstration, compared with the traditional object-oriented development environments. The platform has been developed and deployed in an innovation centre to be evaluated by visiting customers.

Keywords: Service-Oriented Architecture, Model Driven Architecture, Service Meta-model, Service-Oriented Software Engineering.

1. Introduction

SOA (Service-Oriented Architecture) is becoming a leading paradigm for information planning and application integration. Web services ^[1] and Semantic Web ^[2] are emerging as promising technologies to promote service-oriented architecture and service-

oriented e-business systems. With open and standard application interfaces and technologies, Web services technology implements SOA's features to support loosely coupled applications and their integration. They are expected to transform the Web into a distributed business computation network. SOA promotes to bridge business and information technologies, reengineer business processes with service-oriented approaches, and design enterprise information systems with business-centric approaches. A number of information giants, such as Microsoft, Sun, BEA, HP and IBM, have joined the working groups to develop Web services' standards and technologies. IBM and HP advocated "business on demand" and "adaptive enterprises" based on service-oriented paradigm, infrastructure and a number of Web services technologies.

There is a major obstacle for Web services to be applied in enterprise-level and transactional applications, i.e., current platforms, such as those conformed to J2EE and .NET, are based on object-oriented component models. Services are developed using an object-oriented programming model. Service-oriented architectures and protocols are implemented on the object model. The common service resources at the development and runtime environments are to meet the object-oriented frameworks. The absence of service-oriented common resources partly explains the low efficiency of most service communications and operations. For example, XML based transportation and processing proves to have low efficiency without Web services oriented communication and processing resources in current frameworks.

The object-oriented programming method is neither adequately sufficient nor efficient for service-oriented systems and applications. For example, UML based

software development diagrams cannot well define interaction relationships among cross-platform and reusable services. UML creates a “program” or “software” based on objects while service-oriented processes generate a business goal based on service interaction and composition. Moreover, agent-oriented development methods are not suitable for service-oriented engineering. For example, AUML can be used to define collaboration and interaction relationship among agent entities, while service is an abstract concept that agents may provide. However, service-oriented e-Business systems require designers to give more consideration to service identification, design and operations rather than software development. Namely, service-oriented platforms should support business and process design to incorporate SOA’s features of loosely coupled, semantic description and discovery, and cross-platform composition. The service-oriented software engineering should facilitate service modeling and description, enable service-oriented business planning (top-down) and application development (bottom-up), provide technologies for common semantics and meta-model for service-oriented development methodologies, and specify technologies to enable interoperability across the service-oriented tools and platforms.

To sum up, the current application frameworks and platforms mostly focus on the development of Web services (e.g., using UML), or the provision of Web services (e.g., using AUML). The basic service composition and collaboration have been overlooked since they do not have expertise-level service design framework and tools. As a result, it is urgent for SOA/Web services to have appropriate software engineering methodology, platform and tools.

“Class” provides a basic programming model for object-oriented computing and “agent” provides a basic model for agent-oriented applications. A generic meta-model is a key for service-oriented software engineering and platform. Based on the authors’ work on a service meta-model, this work proposes and implements a business-centric, multi-model driven and multi-disciplinary collaborating platform for service-oriented e-Business systems. The platform has applied a service-oriented framework and model-driven architecture to support service-oriented software engineering. An Intelligent Web services model, BPEL4WS (Business Process Execution Language for Web services) based process model, and business-specific model have been used to support rapid e-Business development and intelligent business process integration.

To support service-oriented software engineering, top-down business design and bottom-up service-oriented system development, the platform has been developed to have three basic modeling views, i.e., business view, process view, and service view, to model the system. A ‘business’ is supposed to be committed

by one or a set of services. The platform is a collaborative development environment for engineers and consultants from IT vendors or businesses such as business consultants, technical consultants, and solution designers. They can use the special views to work with their experts and languages. The platform supports rapid service-oriented e-Business analysis, planning, process reengineering and application integration. They can use the special views designated for their roles to collaborate for a service-oriented e-Business system based at the distributed sites.

With the business view, the service view and visual tools of the platform, a business consultant, who knows well business but has no technical background on business modeling, can collect business requirements, model the business, and define business objects and their relationships using common service concepts, such as services and service providers, and plan business processes. His team mate, e.g., a technical consultant (engineer), works with him through the process view and the service view of the platform. The technical consultant, who has service-oriented knowledge and technical background, can use the consistent system information with the business consultant to improve the process model. The design information is translated among the three views based on a global system information model. Their collaborations are distributed in real-time, which can improve mutual understandings between IT and businesses, and dramatically reduce the design problems and accelerate e-Business development.

The rest of the paper will provide the related work, architecture, three views and basic models for the platform. Section 2 presents the related work. Section 3 describes the proposed architecture and views. Section 4 discusses the implementation issues based on the business, process and service model. Section 5 gives a brief introduction of a case study. Section 6 concludes the paper.

2. Scenarios and Related Work

E-Business systems are critical if a business is to be on-demand for dynamic market and customers’ requirements. These systems are required to be responsive in real time, variable in cost structure, and flexible around the world and around the clock to the needs of and changes in the marketplace and the technology landscape^[4]. Enterprises use such systems to streamline their processes, integrate separate business functions, keep processes flexible to changing situations, and reach their customers quickly. The business integration solutions are the key enabler to make on demand business a reality and deliver business value to the customers. To that end, it is important to develop proper integration technologies, e.g., SOA/Web services to support on-demand businesses. In this work,

we consider, from an implementation point of view, to introduce a service-oriented development environment designed to implement an integrated e-Business system efficiently and effectively, through collaborative work among multiple engineers of different backgrounds.

With the advances in information and communication technologies, particularly Internet and Web based technologies, collaborative environments have been recognized as a promising medium for complex engineering design projects, collaborative manufacturing management, remote monitoring and control, enterprise collaboration (including virtual enterprises), and supply chain management^[25]. To implement a service-oriented e-Business system, developers need to capture business requirements and incorporate them into IT systems. There are usually two roles in e-Business planning, design and implementation, i.e., business consultants and technical consultants. A business consultant communicates with the customer to grasp the requirements of the business. The technical consultant has the knowledge of information technologies and approaches needed to support the business. It is a challenge for a consultant to be skilled in both dynamic requirements and emerging implementation approaches. It is difficult for the two roles of consultants to communicate with each other considering the fact that they used to expressing in their own languages. A unified and collaborative platform for both roles is therefore valuable, especially in implementing service-oriented business systems. SOA aims to bridge business and IT, as previously mentioned.

Based on the above requirements, we proposed and implemented a multi-model driven collaborative development platform for service-oriented e-Business systems. The platform has three basic modeling views, i.e., business view, process view, and service view, to model an e-Business system. The three views provide basic modeling tools to support service-oriented software engineering and application developments. Thereby, the service-oriented software engineering is used to support service-oriented system and application development. The proposed views are driven by three models respectively, i.e., service meta-model, process model and business model. Out of the three models, service model is to support semantic service description and operations, process model is consistent with that of Business Process Execution Language for Web services (BPEL4WS), and business model considers services, providers, adapters and their relationships within specified businesses.

In the area of SOA/Web services, academic groups and industries put emphasis on different aspects of Web service based on their different positions in the value chain of Web services. Gartner puts emphasis upon standard interfaces and loose-coupling of Web services. Microsoft emphasizes Web services' combination with

XML, Web and component-oriented technologies. IBM emphasizes more on service-oriented architecture and composition and Web services' integration capabilities. SUN proposes to employ Web services' intelligence to provide customized service, business intelligence, and custom relationship management. In its latest specification of SOA, W3C enforces Web services' open standards and emphasizes on Web services' capabilities of cross-platform, machine recognizable, and semantic processing. In the previous works^[2, 3], we have proposed a semantic service meta-model to promote service-oriented software engineering.

Large-scale and rapid applications of Web services need a secure, reliable, and efficient platform for service-oriented application and software engineering. The platform should provide a generic software model and common infrastructure for development and runtime environments. Currently, main frameworks supporting Web services include J2EE and .NET. Based on the two frameworks, major application vendors, such as SAP, Oracle, Microsoft and IBM, have provided SOA business platforms^[4-7]. Many businesses have published Web services and deployed their information systems in terms of SOA. Several successful industrial applications of Web services have been implemented, e.g., Amazon and Google, which provide developers with Web services to get their own online bookstores and search tools. In most cases, Web services are provided as single interfaces. They do not provide users with facilities to develop sophisticated transactions and applications. Therefore, their Web services are mostly used as separate application interfaces. The proposed platform in this paper, however, is intended to support service-oriented business process planning, composition and integration.

In the area of business process modeling, there have been a number of object-oriented attempts^[8]. Some of them tried to design a single model to represent the whole business process^[9, 10]. Considering an engineer's knowledge, it is a challenge for him to have a good business sense along with IT expertise, to model a business process and develop a reliable service. Some divided a whole business process model into several sub-models. For instance, FIDO^[11] divided a business process model into four sub-models, each corresponding to one aspect of the business. BRADES^[12] divided a business process model into three sub-models according to its lifecycle. It is also difficult to determine details to be included in a sub-model. Tosi et al.^[13] tried to build on-demand information systems by KBCS (Knowledge-Based Customization System). Business processes and business objects are standardized and packaged into components with object technologies. All components are deployed independently of one another using the object-oriented modeling methodology. Users can generically and rapidly customize their enterprise applications that fit

well the business process flow of the enterprise by configuration of those required components. Liang^[14] presented an object modeling approach to help bridge the gap between business systems and OOIS (Object-Oriented Information System) by building a business system model for an organization and generating an object model from it for OOIS. Considering that the above methods are based on the object-oriented paradigm, our platform is service-oriented in process analysis, modeling and implementation. We further provide the ability to transform business requirements to IT implementation.

Huang et al.^[15] introduced a solution template-based approach for business process integration and management. A solution template is defined as a platform-independent model for business process integration and management solutions where a template is represented through a composition of a set of solution artifacts with an explicit description of the interrelationships among the solution artifacts. The proposed platform will apply the same approach along with BPEL4WS. Müller et al.^[16] described a platform independent model based on the UML2, and made a mapping to a platform dependent model based on BPEL4WS. It also demonstrated how the MDA paradigm can be employed in agent-enabled business process architecture. In this work, the process view is similar to Müller's platform specific model (PSM)^[17] in that both of them are described in BPEL4WS. However, the business view of this work is different from Müller's platform independent model (PIM). PIM in UML2 is a high-level abstraction of PSM while the platform's business view is not an abstraction of the process view. It focuses more on business services and their relationships.

3. Multi-Model Driven Development Platform

3.1 Collaborative and integrated architecture

The architecture of the multi-model driven collaborative platform is shown in Figure 1. The platform support service-oriented software engineering and application developments. It employs three views, i.e., business view, process view, and service view to support business and technical consultants' collaborative operations. Business consultants focus more on business view and process view and technical consultants on process view and service view. Consultants can collaborate from distributed sites of, e.g., customers (on the spot) and IT vendors to provide service-oriented e-Business planning and integration, process reengineering and application development. A business consultant models the business by communicating with customers to collect business

requirements, define business objects and their relationships, including services and service providers, and plan business processes through a business view and visualized tools. A technical consultant (engineer) improves the business model and uses his/her service-oriented knowledge and IT language to reengineer business processes, provide necessary information on the internal processes, and implement Web services through a process view and a service view.

The platform is built on an open source platform, e.g., Eclipse, below which is Java Virtual Machine (JVM). Eclipse is the integrated development environment for JAVA developers, which is similar to the Visual Studio of Microsoft^[28]. Business consultants use a Web-based user interface. The distributed sites are connected through the Internet. The tools and views are transferred from the platform on the vendor sites to the consultants' Web browsers on the spot. In the three views of the platform are three modelers based on the three models, i.e., business, process and service. The business modeler is used to capture business requirements, the process modeler is to design detailed process, and a service modeler is to design services. With the multi-model driven platform, there are three templates that are collected from best practices and organized based on the models. Business, process and service templates are all individual artifacts which can be used to compose a business solution for a given business on demand^[18].

3.2 Business, process and service views to support service-oriented software engineering

As shown in Figure 2 and Figure 3, the platform has three views for consultants to plan and develop e-Business solutions, i.e., a business view, a process view, and a Web services view. To keep the design information consistent among the three views, we have developed a global e-Business information model. There are matchmakings among the global model and the previously mentioned three models, i.e., business model, process model and service model. A set of rules have been established to transform an e-Business system among the three views automatically based on the matchmaking rules between the three models. Consultants collaborate through the views to design in their familiar languages.

To create an e-business solution, a business consultant starts with the business view and identifies involved business services, IT services, service providers and their relationships (No. 1 – No.10) for the given business, and based on the requirements collected requirements through communication with the customers. He can choose a business template and design by customization, if there are such templates. He

can transfer to the process view to define the business properties of the selected processes (As illustrated by Figure 3).

On the other side, technical consultants use the process view to define the selected processes, with service-oriented implementation knowledge. Based on a selected process, technical consultants can search the service directory to check the services involved. The services can be bound to the process if they are accessible. Otherwise the consultant has to transfer to service view to create those services by customization if there are service templates, or create a new one on the Web services workspace of the service view (As illustrated by Figure 2 on the left). Another way is to search other sources to find the services that are available, reliable and qualified in the given business. The above operations are repeatable and consultants can navigate three views to collaborate for a complete business solution design. Based on the identification of businesses, processes, and services, description files are being generated for the e-Business solutions. There is a compiler to organize, program, compile and deploy these files on a BPEL4WS compliant engine. A solution can be developed and demonstrated through rounds of definition adaptation and debugging.

The solution is then processed by removing business-specific properties and stored as business templates, process templates and services templates. These templates are reusable when business processes change. Compared with traditional solution development, the template-based method is more cost effective and makes business more responsive. Moreover, the above design process through three views involves (i) top-down service-oriented business planning and process design, i.e., from business to service modules; and (ii) bottom-up service discovery and composition to develop an e-Business system. Also, service-oriented integration is embodied by process modeling and composition.

3.3 Service-oriented business and process integration

Service-oriented businesses and their processes are implemented through Web services and their interactions. Web services interactions can be distributed across applications and enterprises. An e-Business solution therefore requires the platform to support service interaction by providing a number of common integration services. The business model describes both the pattern and protocol of exchanging messages between the services. Each interaction message may contain one or more type of information needed for or by the services during the interaction. Some of these services can be found at service

providers, while others need to be developed by enterprises.

The platform supports multiple types of Web services, including business services, platform services and integration services. Integration services are provided to enable template-based integration and collaboration in e-Businesses. They are: (i) coordination and cooperation: such as adapter services and process-flow-like composition services (ii) ontology and semantic integration (iii) platform services, such as messaging (iv) security services, such as authentication and authorization (v) wrapping services.

When e-Businesses providers have all the service components for e-business systems, they define business templates to compose those services into business processes and applications. Templates describe how services work with each other and how business processes execute. In the platform, business services provide business functions. Common services and platform services provide facilities required by every application. Integration services facilitate interaction between business services by providing agreed interfaces for them. Coordination services integrate all the services to form process templates and an e-business system.

As illustrated by the above scenarios, the three views of the platform support service-oriented software engineering. Supposed that SOA is promoted to bridge business and IT services, service-oriented engineering methodologies are business-centric. When comparing these methodologies with UML, service view is versus class diagram, process view is versus interaction diagram, and business view is versus case use.

4. Implementation Driven by Multiple Models

4.1 Intelligent Web Services model

Model-driven architecture separates business logics from underlying platform technologies. It provides tools to specify a system independently of its supporting platform. As can be seen from the Web services architecture [3], Service-Oriented Architecture is envisioned most efficient in designing enterprise systems and integrating heterogeneous applications in a distributed environment. Service-oriented architecture enables complex systems to be divided into loosely coupled services. Web services technologies make system components to be interoperable and extensible within an enterprise as well as across enterprise boundaries.

Basic concepts behind the proposed platform are business (or e-Business, they are envisioned consistent

in service-oriented methodology) models, process models and service models. We believe that e-Businesses and associated processes and services can be abstracted as a number of patterns. Their compositions can lead to different models of e-Businesses.

As mentioned at Section 1, a major obstacle of Web services applications in enterprise-level and transactional businesses is that current platforms, such as those conformed to J2EE and .NET, are based on object-oriented component models. Services are developed as objects. Service-oriented architectures and protocols are implemented above the object model. As a result, the service-oriented function granularity can not be implemented well to meet loosely coupled and dynamic business processes. Web services applications have performance shortages in discovery, communication, execution, transaction and security, which have hindered SOA's applications in enterprises and transactions.

Authors have proposed an intelligent service meta-model to promote service-oriented software engineering. For intelligent Web services, there are a number of other studies on the integration of software agents with Web services. However, some were on software agents with semantic Web rather than Web services technologies, e.g., specifying software agents with standard mark-up languages like RDF/Schema and RuleML^[19] or DAML-S^[20, 21]. Some are predominantly concerned with enabling the agents in existing systems to request, provide or broker Web services^[22, 23]. Some only concern one specific aspect of agent-based Web services, e.g., integrating DAML-S-based Web services and an agent communications language^[3].

With the proposed meta-model, part of agent capabilities are committed by services entities, and agent-oriented software engineering and semantics-based interaction are selectively applied in Web services construction^[27]. Web services are therefore intelligent and capable of handling communication, interaction and capability provision without agents' brokering. Essential features and functions of service-oriented architecture and Web services are being enabled by service-oriented software engineering. The proposed intelligent Web services are basically used to enforce template-based services dynamic composition in business solution creation, and flexible interaction with B2B partner adapters. For the service composition and deployment, the proposed platform used some expertise to effectively employ web services in the design activities, which was investigated by Younas in the literature^[26].

That is, intelligent Web services make dynamic business processes composition more responsive to changes while making external interface services more flexible to B2B adapters by partner systems.

4.2 Service-oriented and intelligent process model

A well-defined business process model is required to implement the proposed process modeler and view. In the context of the proposed platform, a business process can be modeled using elements such as business participants, value exchanges, internal processes, optimized processes, dependence constraints and exclusion constraints. BPEL4WS is a key in implementing this platform. We have applied its concept model into the platform. Based on it, e-business process and composition services to integrate Web services are implemented.

The business process is modeled and described based on BPEL4WS. It is implemented as intelligent Web services based on the above Web service model, as illustrated in Figure 4. Descriptions of the business processes are translated into OWL-S and become process Web services' knowledge. Processes can therefore apply its intelligence to meet dynamic process execution when representing business processes, find appropriate Web services to invoke when representing composition services, and interact with applications correctly when representing adapter services. The platform takes advantage of process templates to facilitate e-Business development. The templates can simplify the process to coordinate e-Business systems among different partners and create business solutions easily. Business process templates make a difference when creating e-Business solutions through models or templates. They can make the solution creation process repeatable, cumulative and transferable in terms of analysis, design, development and deployment knowledge, less labor and skill intensive, and more predictable for the actual engagements. With the defined templates, processes are reusable components.

4.3 Business models based on objects, services, integration and solutions

The business model concerns business participants/partners, business objects, business services and relationships among them, and the process concerns operation logics in the processes^[2]. Based on the business model, we can establish e-Business templates by the combination of process templates, integration templates, composition templates, application templates and data templates (as illustrated in Figure 5).

Traditionally, an e-Business solution is created for a single enterprise which may not provide flexibility for an extended business. It is common that different systems used by suppliers, manufacturers, sellers and customers are developed with different technologies,

running on different platforms. It is complex and costly to integrate various systems of the whole supply chain as one. It also can not handle changes quickly and effectively when the demands have been changed by another enterprise. This results from business processes being buried in program code ^[24]. However, domain knowledge is a treasure for enterprises. The proposed platform uses business templates to support business intelligence reuse. Web services and business rules engine are combined to support runtime business intelligence. Constraints between participants and optimum processes in a domain should be reused. For example, after providing Web services adapter to traditional business rules engine, we can separate the implementation of complex and inconsistent business rules from the design of business process. Developers can concentrate on inter-relationships of activities at design time, while the business rules can be modified at runtime by users. The new system can apply dependency constraints like “a bank must be involved in a cross-enterprise transaction”, exclusion constraints like “if China Construction Bank is involved in a transaction, China Commercial Bank should not be involved”, and optimized processes such as how to schedule shipment for a distribution centre. They can modify existing templates and get more templates.

4.4 Implementation driven by multiple models

This platform is built on top of the open source platform, i.e., Eclipse, which runs on a java virtual machine. It has four elements, i.e., a business modeler to capture business requirements, a process modeler to design detailed processes, a service modeler to define Web services, and a compiler to generate deployable files from an IT model. There are a number of considerations in developing the proposed platform, e.g., how to define a formal business model to use business language to capture the business-centric requirements, and provide correct information for technical consultants, and how to transform the business information among the three models and views.

The business view is concerned with relationships and value exchanges among business partners, business objects, and business services. The process view is concerns with operations and procedures among business processes. There is no direct way of constructing the process view from the business view. This work is based on BPEL4WS technology. The processes are modeled in BPEL4WS. The partner links in BEPL4WS is used to model the business view, while activities in BPEL4WS are utilized to model the process view. Rules can help in automatically building a skeleton of process view. However, we have made a number of rules to construct a process view having as much information as a business view. Each relationship

in business view contains a sequence number telling the occurrence sequence of that relationship. Base on the sequence number and the direction of the information flow, a dozen of rules can be applied in the process of constructing a process view from a business view.

After the BPEL4WS based process model is worked out, the internal compiler generates two deployable BPEL4WS script files. One is an interface description file. It describes the interface of the business system as a web service with a WSDL file. The other is a process description file. It depicts the internal logic of the business system. After these two files are deployed on a BPEL4WS engine, a business solution is ready to check involved Web services and debug. The service view is for the cases that there are no Web services available.

As shown by Figure 2, the service view and modeler is developed on the Web service meta-model, and with a unified graphical interface for users to access system functionalities. The unified interface provides users with four functional workspaces: ‘design workspace’, ‘development workspace’, ‘debugging workspace’, and ‘runtime environment’. The ‘design workspace’ is for service definition and project management. The definition function collects information about the ongoing projects and services. This workspace covers complete service description items by OWL-S in terms of profile, process and grounding. The development workspace is supported by functions to create descriptions and code files based on the predefined meta-model and user-defined service properties. The description files include OWL-based semantic description files and associated WSDL-based files. Basic service class files are generated based on the proposed service meta-model. The debugging workspace is composed of a building tool, an operation and output window and a view tool. The building tool is provided to generate and deploy executable byte-stream classes on the generated and customized service files. The operation and output window monitors the performance of the studio and integrated platforms through operation logs, outputs and error reports.

Developers can complement pertinent resources to the integrated platforms like Eclipse, Websphere or Tomcat, and make the generated services functional and refined. The ‘runtime environment’ is achieved by the application server. They work together to provide execution, monitoring and management environment for Web services.

5. Case Study and Evaluation

In a typical retail supply chain, there are consumers, retailers (stores, distribution centers, and logistics), transportation providers (carriers), credit authorities, and possibly independent warehouse operators. In retail

stores, there is an inventory tracking service to monitor the shelves and inventory. Once the product drops below a specified threshold, retail stores are notified and a supplement request is submitted to the distribution centre. The distribution centre then checks its repository. If the requested items are available, shipment and transportation will be arranged, where the retailers' distribution centre can choose third-party transportation providers and warehouse operators. Otherwise, if the stock of the requested items is insufficient, the logistics are requested to arrange procurement. In order to do that, it is expected to go through a list of partner suppliers, query prices, and check their credit through credit authorities. All these activities are implemented by web services. Through this initial selection process, the suppliers are narrowed down, and the logistics or headquarters could then start negotiations with each of them, by using negotiation service. A range of negotiation items are supported, including price, quantity, payment type, lead time and due date, etc. The automated negotiation phase carries on until an optimal agreement is reached. After that, the headquarters can place orders to the selected supplier who has a supply service to accept quotations and respond with rates. The supplier also provides services to process purchase orders and shipments. Once the deal is in place, the headquarters authorizes the bank to pay from its account to the suppliers.

Now that the suppliers are selected and the supply chain is well devised, the business integration solution needs to be developed to support the operations of the supply chain depicted above. For instance, the retailer might want to develop an Inventory Tracking Solution to keep track of the movement of goods from the source, through the Transportation Providers, all the way to the sales floor of the Retailer itself. Figure 6 is a user interface to show a process description file behind the three design views to compose the involved retail services and elements with the proposed platform, which was developed using BP4WS.

Figures 2, 3 and 7 are snapshots of the implementation of a retail supply chain on the proposed platform. To develop the retail system, a business process is modeled to describe the business operations. This process description provides details of activities and data flow. It indicates the business and IT services required for the solution. Providers for such services are discovered by searching service directories. The aforementioned business templates may facilitate the search process since they contain pre-identified service providers and their supports for these process tasks. To enable the underlying IT solution, it is necessary to identify the associated applications, data and B2B adapters. As a result, the Inventory Tracking Solution can be put together much faster and, in many cases, in an "on demand" fashion using business templates and intelligent Web services. Existing templates might have

captured the Retailer's solutions for specific issues, and the intelligent Web services could help locate the proper services to make the templates work for the specific requirements of a solution.

The above procedures have been conducted and the retail supply has been implemented through the proposed platform. The platform with the case study has been evaluated by a testing authority. The platform, and its business view, process view and service view proves to be functional, feasible, and helpful to facilitate collaboration and business-centric and service-oriented solution development. The implementation for the retail supply can be demonstrated (Figure 8 is the runtime monitoring view of the developed system). However, the intelligence in process execution and process integration are still ongoing through intelligent adapter services. We expect to implement the intelligence to make service-level, process level and integration-level of operations more flexible. And for distributed collaboration, the platform is to be Web-based.

6. Conclusions

Enterprises need to be responsive to meet dynamic customer requirements. Service-oriented architecture has been envisioned as an appropriate computational paradigm for e-business applications. Service-oriented architecture can improve e-Business applications in order for them to be integrated and flexible. Web services support sophisticated dynamic and automatic services collaboration and composition in an e-Business. This paper proposes a multi-model driven collaborative development platform for service-oriented e-Business systems. The platform has three views, i.e., business view, process view, and service view, to support service-oriented software engineering and application developments. Consultants can collaborate from remotely distributed sites of, e.g., clients and IT vendors to provide their clients with rapid development and demonstration. In itself, the proposed platform is service-oriented and driven by three models, i.e., service meta-model, process model and business model. All of the three models support semantic description and rational operations, and facilitate intelligent service discovery, process execution and business-to-business integration. This platform has been developed and is being deployed in an innovation centre to be evaluated by visiting customers.

However, there are several challenges ahead. Our future work includes the development of sophisticated matchmaking rules for views consistency. Also, we plan to introduce repository meta-schema for the proposed platform design artifacts. The repository in the platform can be an integrated holding area of information about engineered artifacts. The repository should support e-business applications development and deployment

tools and should store database descriptions, form definitions, controls, documents, interface definitions, source codes, help files, executables, and icons, because the design artifacts in this platform should be definable, loadable, and retrievable regardless of their originating environments.

References

- [1] W3C. 2003. "Web Services Architecture". [Online]. Available: <http://www.w3.org/2002/ws/arch/>.
- [2] Maria Bergholtz, Prasad Jayaweera, Paul Johannesson, Petia Wohed, "Process Models and Business Models –A Unified Framework", ER 2003 Ws, LNCS 2784, 2003, pp.364-377.
- [3] Y. Li, W. Shen, and H. H. Ghenniwa, "Intelligent Web services Framework and Development Environment", *Computational Intelligence*, Vol. 20, No. 4, 2004, pp.76-88.
- [4] IBM. 2004. WebSphere Studio Application Developer: Features and benefits. [Online]. Available: <http://www-306.ibm.com/software/awdtools/studioappdev/about/>
- [5] Microsoft Corporation. 2005. .NET Infrastructure and Services. [Online]. Available: <http://msdn.microsoft.com/netframework/technologyinfo/infrastructure/default.aspx>
- [6] SAP. 2004. SAP NetWeaver: Technical Details. [Online]. Available: <http://www.sap.com/solutions/netweaver/technicaldetails/index.epx>
- [7] Pierre Fricke. 2004. Oracle's Service-Oriented Architecture Strategy. [Online]. Available: <http://www.oracle.com/solutions/integration/dhbrownwp.pdf>
- [8] Georgem, Giaglis. 2001. "A Taxonomy of Business Process Modeling and Information Systems Modeling Techniques", *International Journal of Flexible Manufacturing Systems*, Volume 13, Number 2, pp. 209 – 228.
- [9] Irma Valatkaite, Olegas Vasilecas. 2003. "A Conceptual Graphs Approach for Business Rules Modeling", *ADBIS 2003*, LNCS 2798, pp. 178–189.
- [10] Yoshiyuki Shinkawa, Masao J. Matsumoto. 2000. "Business Modeling and Component Mining Based on Rough Set Theory", *ICSR-6*, LNCS 1844, , pp. 231–250.
- [11] Dan L. Shunk., Joong-In Kim, Hee Yerl Nam. 2003. "The application of an integrated enterprise modeling methodology-FIDO-to supply chain integration modeling", *Computers & Industrial Engineering*, 45, pp. 167–193.
- [12] Daniela Rosca, Chris Wild, 2002. "Towards a flexible deployment of business rules", *Expert Systems with Applications*, 23, pp.385–394.
- [13] S.K. Tsoi, C.F. Cheung, W.B. Lee. 2003. "Knowledge-based customization of enterprise applications", *Expert Systems with Applications*, 25, pp.123–132
- [14] Ying Liang. 2002. "Generation of Object Models for Information Systems from Business System Models", *OOS 2002*, LNCS 2425, pp. 255-266.
- [15] Ying Huang; Kumaran, S.; Bhaskaran, K. 2003. "Platform-Independent Model Templates for Business Process Integration and Management Solutions" IRI 2003. IEEE International Conference on, pp. 617–622.
- [16] Jörg P. Müller, Bernhard Bauer, and Thomas Friese.2004. "Programming Software Agents as Designing Executable Business Processes: A Model-Driven Perspective", *PROMAS 2003*, LNAI 3067, pp. 49–71.
- [17] Joaquin Miller, Jishnu Mukerji, "MDA Guide Version 1.0.1", <http://www.omg.org/docs/omg/03-06-01.pdf>
- [18]Y. Huang, S. Kumaran and K. Bhaskaran. 2003. "Platform-Independent Model Templates for Business Process Integration and Management Solutions", IEEE, IRI 2003.
- [19] Dale, J., Ceccaroni, L., Zou, Y., Agam, A. 2003. Implementing Agent-Based Web Services, *AAMAS 2003, Workshop on Challenges in open agent environments*, Melbourne Australia.
- [20] The DAML Services Coalition. 2002. DAML-S: Semantic Markup for Web Services. [Online]. Available: <http://www.daml.org/services/daml-s/0.7/daml-s.html>.
- [21] Lyell, M., Rosen, L., Casigni-Simkins, M., Norris, D. 2003. On software agents and Web services: Usage and design concepts and issues. In: *Proceedings of the AAMAS2003 Workshop on Web Services and Agent-based Engineering*.
- [22] Maamar, Z., Sheng, Q.Z., Benatallah, B. 2003. Interleaving web services composition and execution using software agents and delegation. In: *Proceedings of the AAMAS2003, Workshop on Web Services and Agent-based Engineering*.
- [23] Gibbins, N., Harris, S., Shadbolt, Nigel. 2003. Agent-based Semantic Web Services. In: *The Twelfth International World Wide Web Conference*, Budapest, HUNGARY.
- [24] X. Lu, Y. Li, and Y. Huang. 2004. "On-demand Business Process Integration Based on Intelligent Web Services", *Proceedings of International Conference for e-Business*, Tsinghua University, Beijing.
- [25] W. Shen. 2005. "Special issue on collaborative environments for design and manufacturing". 2005 (19). *Advanced Engineering Informatics*. Elsevier. pp.79.
- [26] M. Younas, K. Chao, C. Laing. "Composition of mismatched web services in distributed service oriented design activities". 2005 (19). *Advanced Engineering Informatics*. Elsevier. pp.143-153.
- [27] W. Shen, Q. Hao, S. Wang, Y. Li, H. Ghenniwa, "An agent-based service-oriented integration architecture for collaborative intelligent manufacturing". 2007(23). *Robotics and Computer-Integrated Manufacturing*. Elsevier.pp.315–325.
- [28] <http://www.eclipse.org>

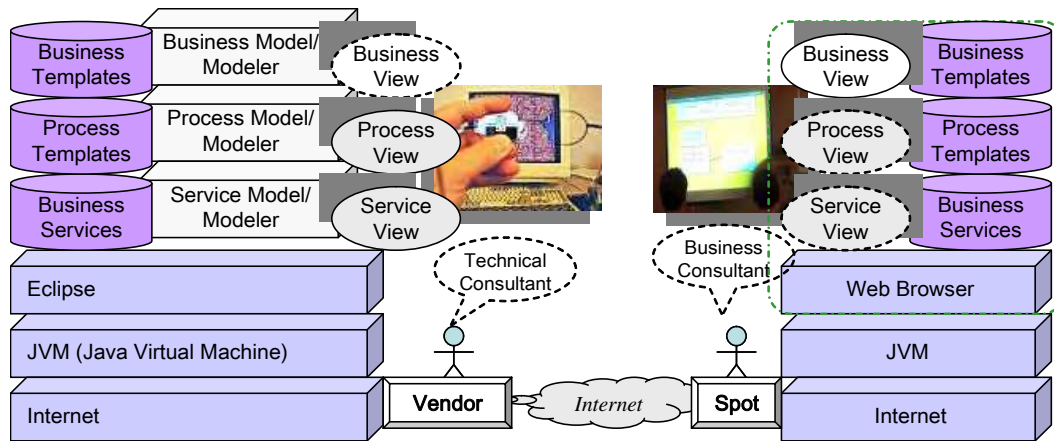


Figure 1. Architecture of the multi-model driven collaborative platform

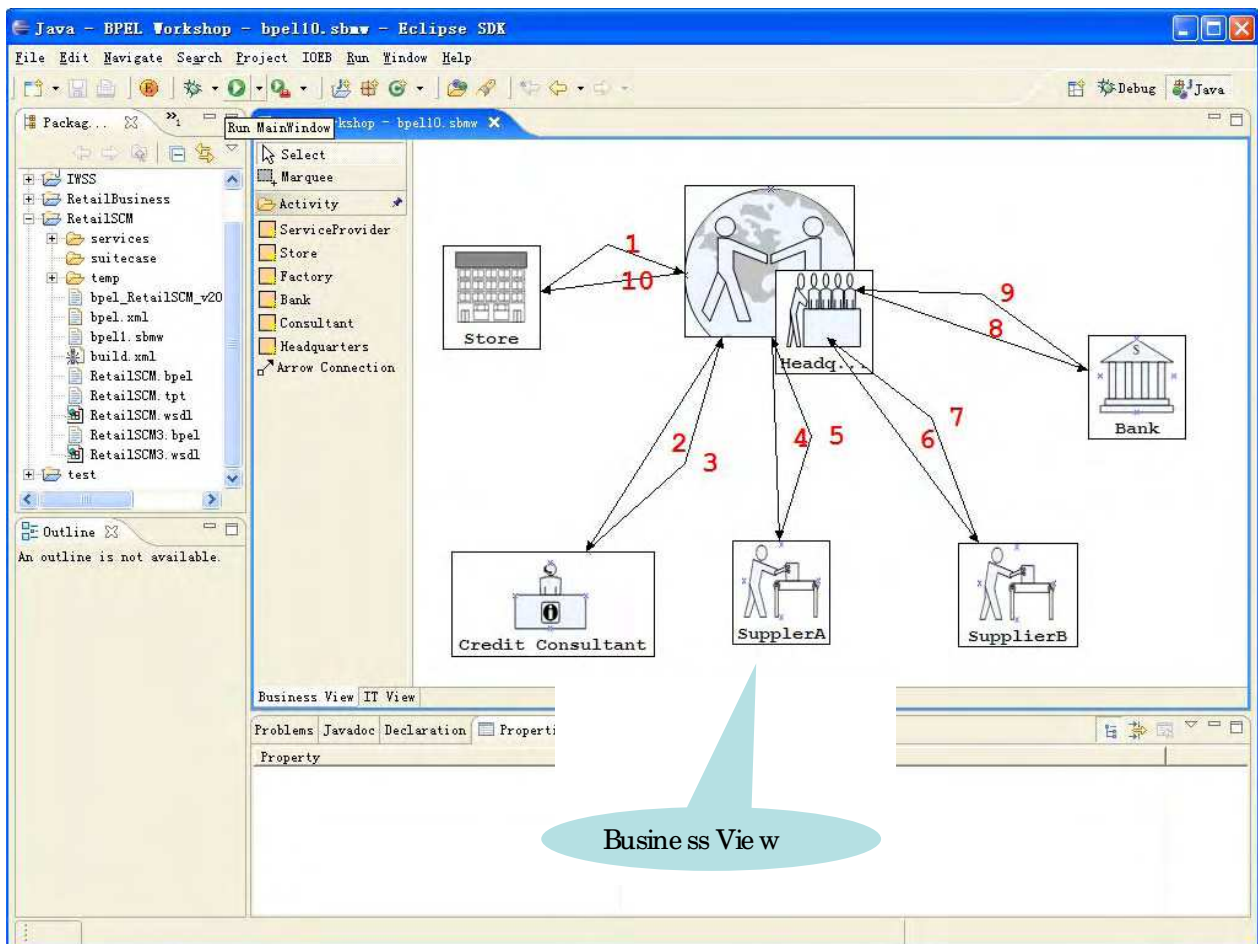


Figure 2. Multi-model driven collaborative platform for e- Businesses (Business View)

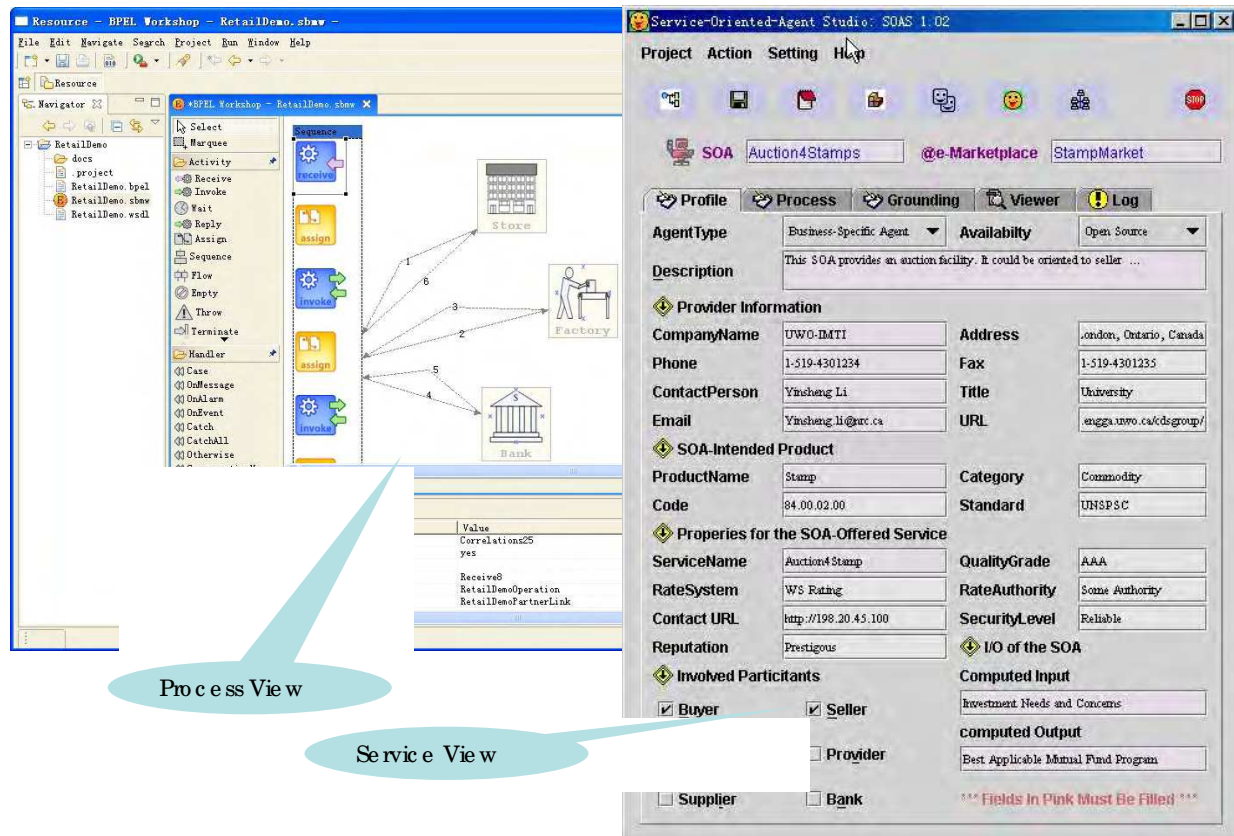


Figure 3. Process and service models driven development views

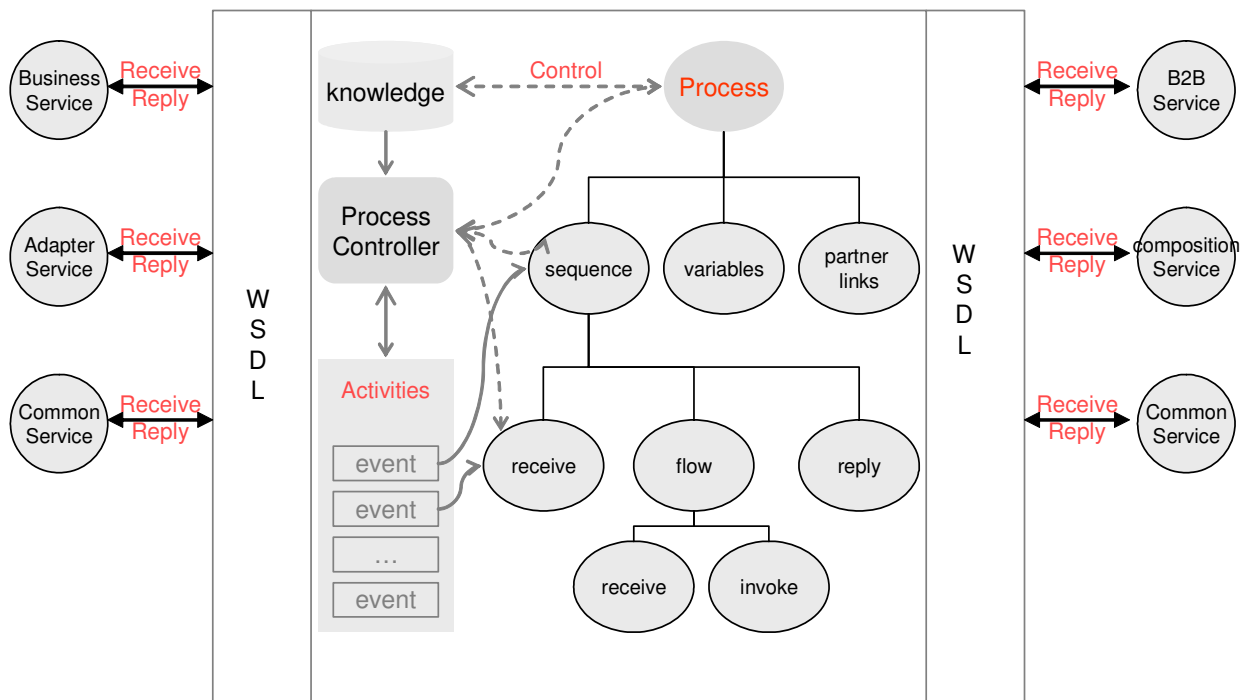


Figure 4. Process model based on BPEL4WS and intelligent Web services

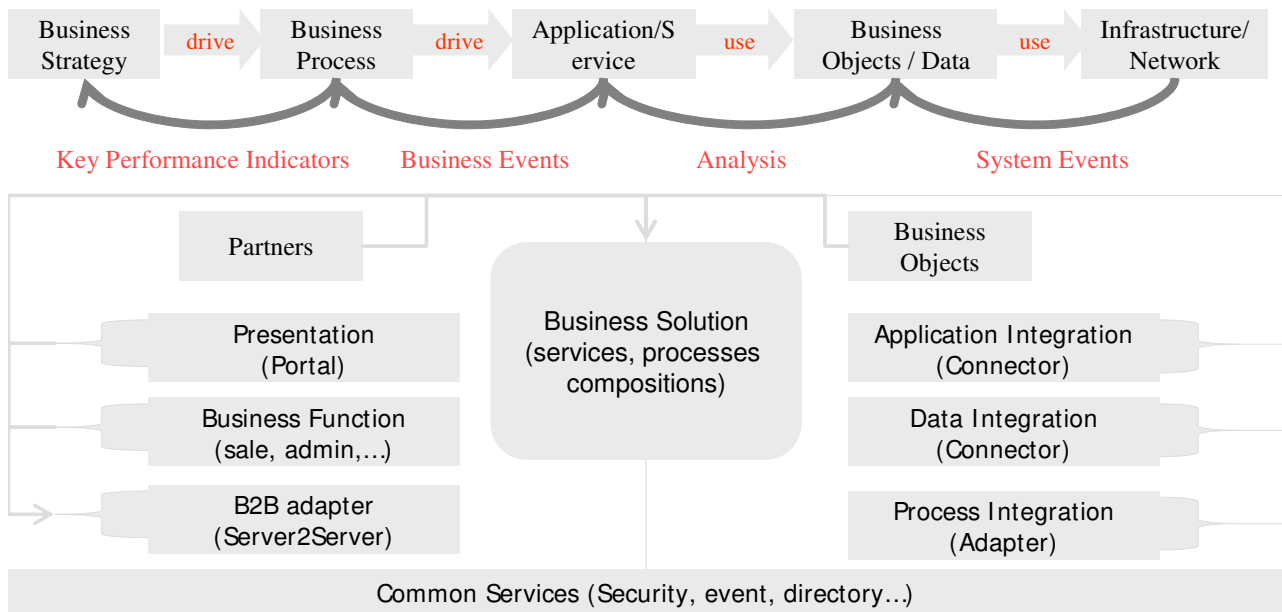


Figure 5. Business model with combinations of process, objects, services

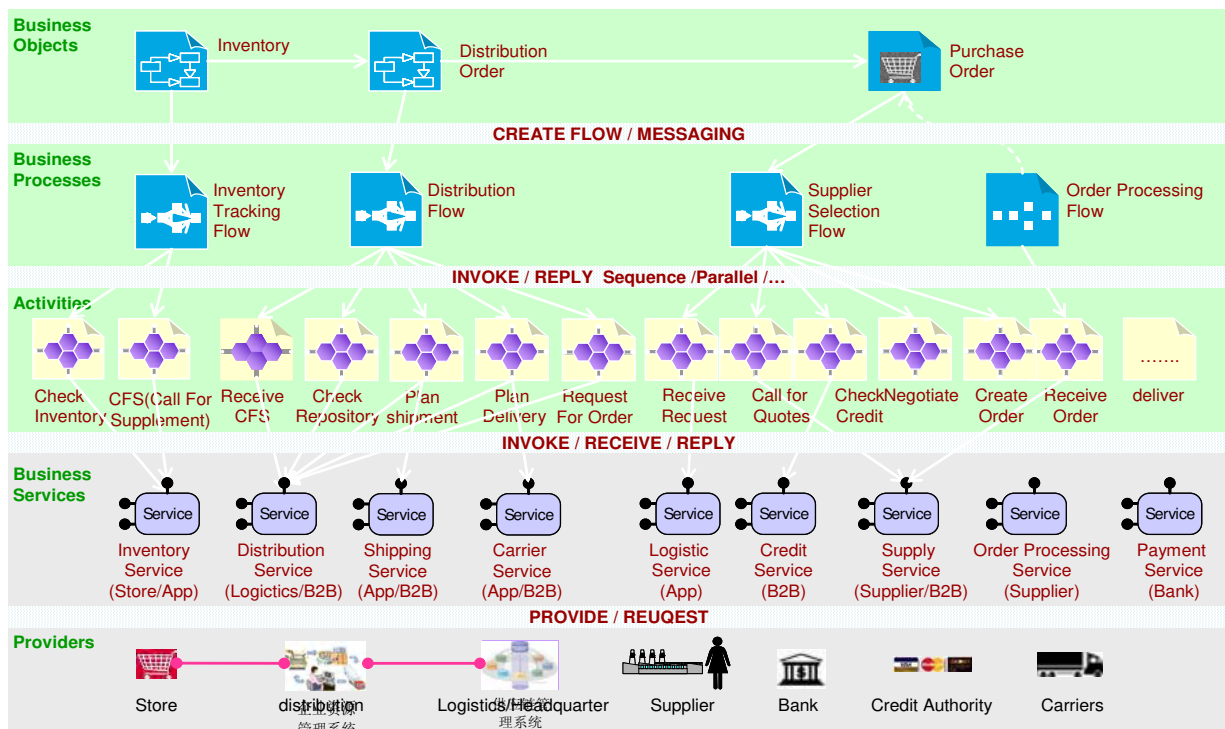


Figure 6. Design considerations for retail supply solution developments on the platform

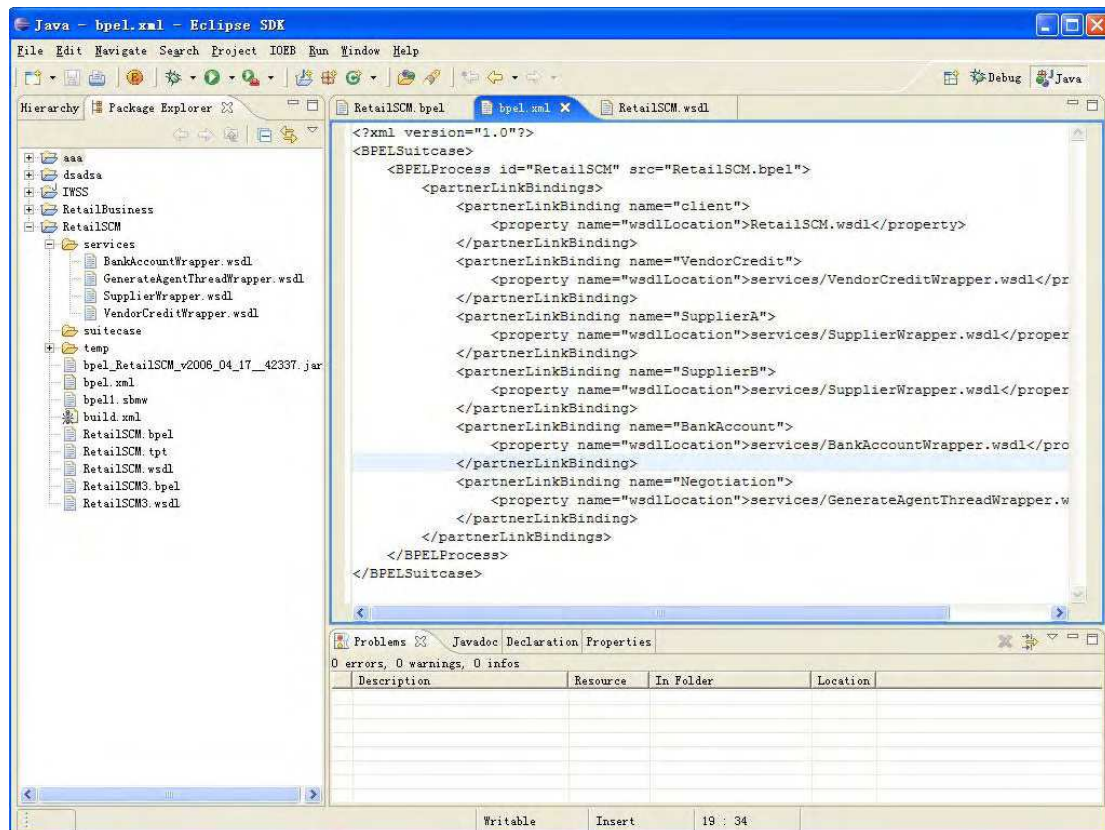


Figure 7. A BPEL-based description for retail supply process

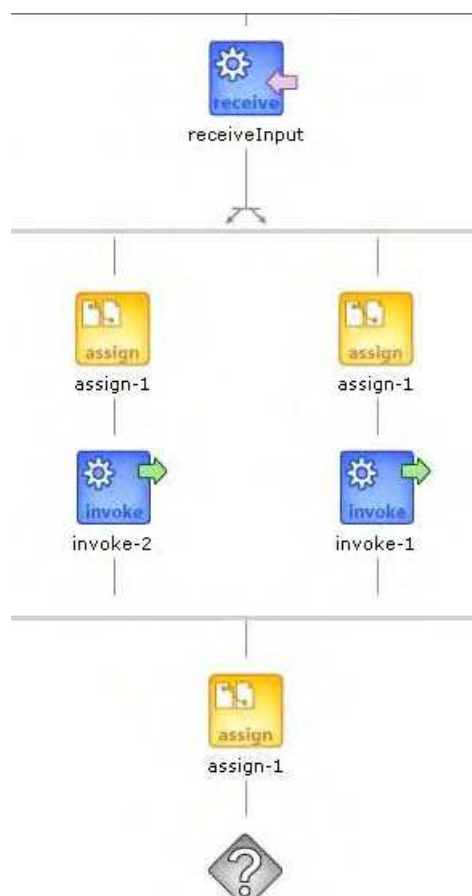


Figure 8. Runtime monitoring view of retail logistics