Lucjan Grochowski
Politechnika Warszawska, Wydział Transportu

INTEGRATION OF LOGISTICS PROGRAMMABLE SERVICES IN THE WEB

Abstract: The paper describes programming of advanced services addressed to logistics services to manage distributed functionalities in the Web environment. The program components are defined and the components operating principles. Considerations are referred to actually the most representative component technologies: Enterprise Java Bean (EJB), CORBA Component Model (CCM) and .NET of MS Corporation. They are illustrated by the implementation framework using in Web Services various types of program components independent on the data formats and programming languages which describe supplying chain management.
Key words: distributed programming, WEB, logistics

1. INTRODUCTION

Logistic services involved in transportation of the goods generate a large amount of data which has to be transferred and shared. Data are handled by the information management framework assuring proper and advanced service functionalities determined by used software application programs. In recent time for such services a new challenge makes their expansion cross of the world using the Web operations. For business purposes in the Web a key role plays the system efficiency which can be upgraded by integration of services described by arbitrary the data formats and programming languages. The example can be supplying the chain management with the third-party logistics operations based on outsourced services [1]. Then, typical de-compartmentalization requires:

- operating of the distributed databases to provide the real-time operations,
- handling of the multimodal transport; transport adjusted not only to product differentiation and tracing but also to customization on demand with WEB sites,
- ability to be matched the different legislations and customs requirements covering the financial and multi time zones mainly by a Customer Relationship Management (CRM),
- cost effectively support of low volume users in remote regions.

These and similar distributed functionalities of services from one side require more advanced network programming methods from other side the simple operating manner by re-using the sub-services described by autonomic blocks of program codes. As the single sub-services can't provide a requested full service; to perform the complex services they have to be combined with the other Web accessible sub-services. Then, advantage makes the ability to reuse the partial services as a number of separated sub-services which can be put together in various configurations making Web composite system. Such interactions over a network can significantly upgrade functionalities if proposed in the paper the component programming approach is implemented. Then, composite Web services system involves solutions to a number of problems by recognition of useful candidates for execution of required services and calculation of possible system composition.

The paper handles these problems and demonstrates implementing the programmable service-components, their operating principles and usefulness on frame implementation examples addressed to logistic services in the Web. Carried out considerations are referred to recently the most representative component technologies: Enterprise Java Bean (EJB), CORBA Component Model (CCM) and .NET of MS Corporation. Considerations are illustrated by an implementation example to apply proposed component approach to operate the advanced supplying chain management based on the hybrid components involving components belonging to various technologies.

2. COMPONENT PROGRAMMING APPROACH

The component programming methods make the transient stage to more advanced programming methodology, referred to semantic models and collective system intelligence. While semantic and collective system intelligence methods still are insufficiently mature to be implemented on wide scale, the component programming can make a tool of efficient programming acceptable by business for the Web based advanced services. Component programming deals with distributed computing driven by programs split up into parts which are run simultaneously on multiple computers communicating over a network.

The operating mechanism is based on multi-using of program units making a set of independent elements that allow to be used on plug and play principle. The contribution recommends a definition given by Clemens Szyperski [2]: *a software component is a unit of composition with contractually specified interfaces and explicit context dependencies only*. So, the software component working model can make a reference framework without of specified programming language with the data accessible from outside. Under these conditions, the input data are given by the input arguments, their values and parameters, and specify all necessary information how to use the component. If this specification is related to required functionalities, it describes required and provided services. The components can make the sub - parts of larger systems in which they make the building blocks completing the system operations described by the suitable language compositions.

In the component programming approach a source code doesn't exist in traditional sense, it is replaced by adjusting component specification to the individual requirements of

the users. Besides of that, an important role plays the component context independence that allows to the components to be easily transferable into different application contexts.

The components' interface inheritance is allowed only under strictly described specifications [2 - 4]. The component details are differentiated. They preserve only full specification of function dependencies, ability to be used as a program part and self made unit with ability by simple manner of co-operation and integration with the other components. This imposes the categorization of component interfaces, usually, divided into three groups with:

- dedicated interfaces with component dependencies injected by interface,
- interfaces described by constructor parameters related to component container,
- interfaces defined by object features, so called setter injection method.

At present time, the most representative network technologies implementing the component programming make [5]:

- Enterprise Java Bean (EJB) of Sun Microsystems Inc.,
- CCM Components based on CORBA,
- .NET Components of Microsoft Co.

The JavaBeans Components

JavaBeans use the most important Java language features; they allow building a program component by reusable software components which are manipulated visually in a builder tool. Thus, the components can overcome the features of common Java classes as the JavaBeans interfaces don't prescribe the implementation of any methods. They make just an approval that the "program bean" may be saved in persistent storage in a file or database. So, the bean can be either restored after implementation or transferred across the computer networks. Also, the bean properties allow to program operations to be accessible and modifiable by Java public methods getter and setter. The JavaBeans can be activated by well known events mechanism; it assures the action what a given component can give to other components, if specified action is undertaken.

The Java event model realizes the program design pattern assuring reduction of inter-component couplings resulting from component communication referred to interfaces. At this time, component JavaBeans technology is the most popular. This is a result both the popularity of Java language and the efforts of the SUN Inc. to increase a number of frame implementation of Java technology on the platform J2EE. The component functionality and introduced changes are combined by protocols used in bean network communication biding together needed beans - simple program modules - into a whole advanced implementations [6]. Enterprise Java Bean (EJB) uses three types of the components: the data units, the sessions and the data services. The components are initiated and managed using containers assuring the data access and ability to make transactions.

The mature version of EJB interface solves implementation problems for nontransferable component live cycles and component search out mechanism. Popular manner of dependence injections are based on active searching out components in the container registers distinguishing the component interfaces possessing differentiated dependencies for simple component selection for a given implementation.

An advantage of the JavaBeans component model is its simplicity. Developing JavaBeans is rather simple, because a lot of their behaviors, like the platform independence or packaging mechanism, are supported in the Java Programming Language by default.

Moreover, beans optionally are equipped with additional objects like BeanInfos or custom PropertyEditors responsible for a bigger flexibility of components. Substantial advantage of Java component model is, also, designed by the SUN Inc. the model of Graphical User Interface library allowing building the programs with different stage of complexity.

The CCM Components

The CCM Components are based on the COBRA Component Model (CCM). They make continuation of introduced prior to WEB the heterogeneous COBRA standard to cooperate in WEB environment each application independently from their software platform, format, and structure and programming language used.

The model CCM is referred to COBRA 3 standard and determines the cooperation frame of program components in more wide range than this can be done by the EJB standard. In this approach the containers are introduced to offer a set of services, which replace the services of single components. Thus, the programming structures are simplified, especially, if it is referred to those which are used in the classical COBRA approach [7-9]. CCM components are combined each other by the use of ports that are classified as:

- facets – defining interfaces referred to point to point principle transferring invocation programming methods from component to component,
- receptacles – making working principle dependent component interfaces,
- event sources / sinks – allowing the information transfer among a few components.

In CCM approach the container determines working environment for a one or a few of components which are selected by the vents, transactions, security etc. Each container is responsible for the component initiation and component attributes management, referred to the component biding with the component indirect services called middleware services. In distributed WEB network each component can be configured by different manner. They make so called assembly entities, which determine the virtual component server converting physical into logic objects responsible for distributed services and sub-systems.

This is equivalent to standardization of component implementations by the component packets with distinguished functional features and application mechanisms. These later are referred to Dynamic Link Library and metadata describing programming features in application context.

The .NET Components

The .NET Component Technology is a follower of earlier MS technologies. In the run environment the .NET programming platform introduces the Common Language Runtime – (CLR) and the application oriented class libraries. The characteristic feature of such platform is the universality as on this platform the component programs can be written in any, network oriented high level programming language. Such application codes can manage the WEB servers and the work stations in a given programming environment [8].

Typically the .NET platform is supported by the compilers of high level languages usually C ++ / CLI, C #, Visual Basic and J # as the compilation mechanism of the .NET is similar to compilation used in the Java. Byte code with each transient part method of codes is transferred to control the compiler responsible for the final form of codes. As it was shown above the .NET platform is distinguished by meta code Common Language Infrastructure CLI that assures to each programming language an access to .NET libraries, if a given programming language meets the Microsoft object standards.

3. EXAMPLE OF ADVANCED LOGISTICS SERVICES BASED ON COMPONENT PROGRAMMING

Below, the logistics examples of Internet based supply chain management (SCM) which make an implementation of presented above component programming methods will be demonstrated. At present time, the logistics is developed very dynamically as more and more business-to-business transactions can be made via Internet. Mainly, the current Enterprise Resources Planning (ERP) systems are supported with the Internet access, allowing the supply chain management to be integrated with information flow across each enterprise. The management effectiveness of these systems is strongly dependent on programming methods, assuring to those systems the new advanced functionalities which can be developed, using the presented above component programming. Decision support based on programming with advanced functionalities becomes increasingly important, especially, in view of the supply chain management referred to the store and product analyzing, the inventory, and the sales information. A big advantage is a fact that the component programming of WEB logistics operations hasn't to be restricted to one type of component technology. This facilitates the operation across of the logistics networks by expanding programmable co-operation with many managing branches of ERP systems in areas of collecting and sharing information, product information, and price negotiation and distributes alerts etc. [9 -11].

The system framework of the most vital logistics supply chain operations, referred to the WEB services and presented above component programming can be based on the WEB programmed services accessible via "logistic hubs" that enable to use the WEB to support fulfillment of various processes end-to-end. The proposed system will be able to control the orders, shipments and inventory in real time operations, making operation platform for supply chains under conditions of dynamic and flexible logistics flow. At each operating time this flow is adjusted to react on changing demands of supply chain management.
Integration proposed system can cover:

- cross dock operations as part of logistics flows,
- WEB-based order management, client communication and collaboration,
- enhanced "event management" capabilities,
- operations where product is shipped from import warehouses directly to end customers, rather than first going to the broker distribution centers,
- multi modal transport operations.

This system if properly programmed substantially allows upgrading the services, as flexible functionalities can be tailored to each needed application at various stages of complexity. This is a result to use to supply chain management the heterogeneous programming which accepts the different component programming origins. Such Web Services can be, also, reused and composed into larger systems in which the services can be found on the WEB without to be downloaded and deployed in order to be used by many users. Proposed the Web Service framework can assure to each authorized user to discover and access the service functionalities using Web Service the SOAP, WSDL and UDDI protocols based on XML language descriptions. In consequence, the services can be assured by different program components which are mutually integrated across different

programming component frameworks. In fig.3.1 is shown proposed the Web Service framework

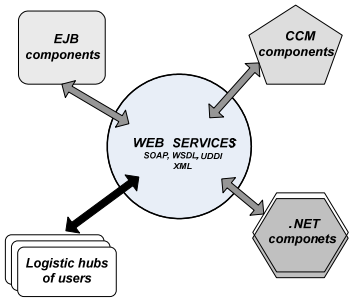

Fig. 3.1 Web Service heterogeneous components in supply chain management

Shown in fig. 3.1 the Web Services are referred to W3C requirements. To meet those goals the Web service system architecture should contain: **the Service requestor** - the user of a service, **the Service provider** that implements the service and offers to carry it out, on behalf of the requestor, **the Service registry** - sub-system in which available services are listed: they allow to providers advertising their services and requestors to search out and query for services.

For presented supply chain management the requested services are referred to programmed components. They are available using different component frameworks made by the component technologies described above. Under such conditions, the component frameworks are accessible for the users by the Web logistic access hubs associated with the distribution centers, warehouses and distribution points determining characteristic locations of logistic operations. Operating principles of proposed WEB based framework of the supply chain system management is shown in Fig. 3.2.

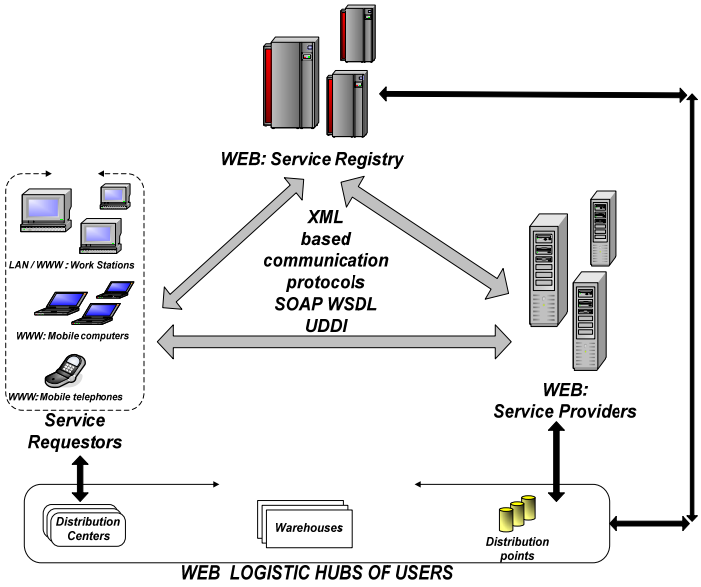

Fig. 3.2 Framework of component programmable services based on Web Service approach

Shown in fig.3.2 the framework of logistics flow from source through a number of logistic processes to the customer can cover and coordinate re-use, re-cycle, or disposal of all resources. It is adjusted to arbitrary service level including Request For Quotes, Shipping, and Tracking, and it can be addressed to both B2B (business to business) and B2C (business to consumer) processes. In the system there is proposed the available services to categorize into three layers:

- **A Higher-level Service Interface** that encapsulates the clients from multiple transportation carriers and provides an abstraction layer for available services. All shipping carriers interested in such Web integrating should implement such interface, and then the resulting Web Services can be published in the UDDI.
- **The Adaptation Layer** being a key connector between the Web Services and corresponding legacy applications. This adaptation layer works as a service dispatching broker and service aggregation broker, responsible for manipulating the requests from the user and the responses from the server.
- **Dynamic Data Binding**, a function performed by the adaptation layer. The adaptation layer is used to create a connection template while the dynamic data binding mechanism integrates real-time data into the defined connection template.

In component approach the context Web systems can based also on the wire-less solutions. Then, advantage is operating anywhere anytime disadvantage is necessity to accept limits in using programming possible to operate at reduced set of instructions. For logistics obvious profit makes the wire less system referred to context capture of data. IP addresses in the Wireless Local Area Networks (WLAN) based on the Wi-Fi or Wi-Max technologies can assure this. Such context can be captured from current location of computer device via:

- WLAN positioning system and GPS outdoor and indoor tracking system,
- component framework application for user device implementing RFID tags,

Changes in the context prompt the context broker to trigger the pre-programmed events. Events may include delivery of push-based messages to the user or an exchange of information with other applications, to make them aware of the events on the site. E.g. change of location or tasks to recalculate the available services to user in real-time [12].

3. CONCLUSIONS

The paper contributes to the area of business integration of processes by providing a new service framework model which meets the demands of logistics. Presented problems are focused on the component programming principles referred to the most representative component technologies and their mutual cooperation to build the new advanced functionalities which can be independent on programming languages, formats and data structures. Usefulness of presented approach was demonstrated by the system model framework addressed to logistics supply chain operations based on the WEB programmed services accessible via “logistic hubs” that enable to use the WEB to support the fulfillment of various processes end-to-end across the glob. Proposed framework allows,

also, on customization with many WEB sites on demand determined by configuration options equivalent to help improving the enterprise competitiveness related to integration of shipping, freight forwarding, warehousing and financial management associated with capabilities of external systems. In shipping, for example, this means integration from quotations through bookings, invoicing, manifests and documentation. In Freight forwarding it means a seamless flow across multi-modal movements, with a strong emphasis on track and trace. This demonstrates advantages of Component Programming / Web Service methodology, to build the advanced services with possibilities substantially overcoming possibilities of other methods so far used.

References

1. Web Service Integration, United States Patent 7 475123
2. Szyperski C., Component Software, ACM Press/Addison-Wesley, England, 1998
3. Heineman G., Councill W., Component-Based Software Engineering: Putting the Pieces Together. Addison-Wesley Professional, Reading 2001
4. Szyperski C., Component Software: Beyond Object-Oriented Programming. 2nd ed. Addison-Wesley Professional, Boston, 2002
5. Veryard R., Component-based business: plug and play. London : Springer, 2001.
6. http://java.sun.com/j2ee/docs.html, J2EE Documentation, 2001
7. http://www.netcraft.co.uk/Survey/ 2008
8. http://developer.netscape.com/ 2009
9. Grochowski L., Componet Programing of Transportation Services in WEB Environment, Monografia IX Konferencji Telematyka, Logistyka i Bezpieczeństwo Transportu, TL&TS '09, Katowice-Szczyrk 15- 17 październik 2009, unit 2.3
10. Alonso G., et al., Web Services, Concepts, Architectures and Applications, Springer, 2004
11. Clark M., et al., Web Service Business Strategies, ed. Export Press, Copyrighted Material, Chicago 2009
12. Balqis O., Tabarak B., Intelligent Wireless Web Services, http://www.itcon.org/2009/20 Journal of Information Technology in Constr., June 2009

INTEGRACJA PROGRAMOWALNYCH USŁUG LOGISTYCZNYCH W WEB

Streszczenie: Praca dotyczy realizacji zaawansowanych usług dla potrzeb logistyki w rozproszonym środowisku Web i wykorzystującym metody programowania sieciowego w podejściu komponentowym. Zdefiniowany jest komponent programowy i jego mechanizmy operowania w środowisku Web. Rozważania są odnoszone do aktualnie najbardziej reprezentatywnych technologii komponentowych Enterprise Java Bean (EJB), CORBA Component Model (CCM) and.NET of MS Corporation. Zademonstrowany jest również przykład szkieletu budowy aplikacji logistycznej WEB poprzez zróżnicowane typy komponentów. Komponenty te są niezależne od formatu danych i języków programowania i służą do informatycznego wspomagania zarządzania logistycznym łańcuchem dostaw.

Słowa kluczowe: programowanie rozproszone, WEB, logistyka