

# ELPIF: An E-Logistics Processes Integration Framework Based on Web Services

Liang-Jie Zhang, Pooja Yadav, Henry Chang  
Rama Akkiraju, Tian Chao, David Flaxer, Jun-Jang Jeng  
IBM T.J. Watson Research Center  
P.O. Box 218, Yorktown Heights, NY 10598  
zhanglj@us.ibm.com

## ABSTRACT

In recognition of the increasing importance of globalization and the resulting need for greater, faster and more flexible communications, a framework is required to allow any company to establish itself in no time or make optimum use of their legacy applications and run efficiently with minimal cost input. This paper presents such a framework called ELPIF for e-logistics processes integration based on Web Services via incorporating (1) common alliance layer (2) adaptation layer; and (3) dynamic data binding mechanism. This framework can be adopted as a new service delivery model which uses a design pattern and solution templates. The interaction between the e-logistics processes and business process manager that orchestrates e-logistics processes in an e-business solution will be described in this paper. A transportation planning in the purchase order management process of a B2B solution is used as an example to illustrate the usage of ELPIF by encapsulating United Parcel Service (UPS) on-line XML Tools as Web Services.

## Keywords

Web Services, E-logistics, Business-to-Business, Shipping, Common Alliance Layer, Dynamic Data Binding, Adaptation Layer

## 1. INTRODUCTION

In last couple of years, various on-line shipping tools have been developed for the e-commerce application developers. Take the example of the transportation industry, UPS provides on-line XML Tools and HTML Tools [1], and FedEx provides their own Web tools (FedEx API) [2] for their developers to enable the development of on-line shipping tools, respectively. However, we have not seen a common service interface to allow users to easily hook up existing tools. Client application developers have to manually construct different requests for different backend servers that demand much effort and time. Different shipping carriers might require different implementations and could have proprietary platform and their own implementation constraints.

In order to expedite the shipping process and minimize costs, the

shipping solutions must empower the customers and suppliers with the ability to rate, ship and track shipments. Many solutions in today's competitive market have been able to achieve this but they are:

- Platform dependent and unique to the specific shipping carrier. The solutions are not generic so that they could not be considered as a standard and followed by the rest of players
- Windows-based applications that are mostly standalone applications and users are forced to purchase them before actually using them [4].

With the development of Web Services, to define a uniform interface for the solution developers becomes technically feasible and itself leads to potential business opportunities. The framework, ELPIF, presented in this paper proposes to have a common generic interface for all the Shipping Service Providers and all providers could build their Web Services on such a standard interface and then deploy those services in the UDDI Registries for other companies to find and use them. Though we are focused on the shipping industry, the principles embodied in ELPIF can be applied to other domains.

Blending the Web Services and the common interface approach would result in allowing shipping services to adhere to a model of what could be considered a generic shipping service. This is critical since this allows a shipping service client to design and deploy code to use the generic shipping model, and then at run time use dynamic data binding mechanism to invoke a specific implementation of a shipping service. Because Web Services can be implemented in any programming language, developers are not obligated to change their development environments in order to generate or use Web Services. Consequently, any client application can benefit from the characteristic of architectural independence that is embraced in our framework.

For most integration architecture, XML plays a role of trivializing the exchange of business data among companies by providing cross-platform approach in the areas of data encoding and data formatting. For example, SOAP, built on XML, defines a simple way to package information for information exchange across system boundaries. UDDI Registries, on the other hand, allow programmable elements to be placed on Web Sites where others can access remotely. By adopting the above technologies, not only do we get interoperability for our customers but we can use our multi-platform approach to provide better offerings and solutions with the help of which any industry can accomplish their transactions efficiently and profitably.

ELPIF serves as a Web Services Model such that any user could easily access the services provided by ELPIF through a standard SOAP protocol. ELPIF helps businesses act more quickly and more efficiently, and it also provides a methodology of automating process integration resulting in reducing integration time and cost, increasing the efficiency of service delivery, and gaining competitive advantage in the marketplace.

This paper is organized as follows: Section 2 discusses e-logistics processes integration after reviewing e-logistics processes and also presents our proposed integration framework, ELPIF, by introducing common alliance layer, adaptation layer and dynamic data binding mechanism. Section 3 gives an integration example using UPS On-Line XML Tools in a purchase order management process. Section 4 shows a working B2B system using e-logistics Web Services. The conclusion and future direction will be given in the end of the paper.

## 2. E-LOGISTICS PROCESSES INTEGRATION

### 2.1 E-Logistics Processes

When it comes to logistics, the challenge has always been how to deliver products to customers as quickly as possible. Logistics is concerned with the flow of materials in the supply chain, from source through the industrial process to the customer, and then on to re-use/re-cycle or disposal. By coordinating all resources, logistics have to ensure that service levels agreements with customers are honored.

E-logistics is defined to be the mechanism of automating logistics processes and providing an integrated, end-to-end fulfillment and supply chain management services to the players of logistics processes. Those logistics processes that are automated by e-logistics provide supply chain visibility and can be part of existing e-Commerce or Workflow systems in an enterprise.

The typical e-logistics processes include **Request For Quotes (RFQ)**, **Shipping**, and **Tracking**. As shown in Figure 1, e-Logistics interacts with the business process manager in an e-Commerce server.

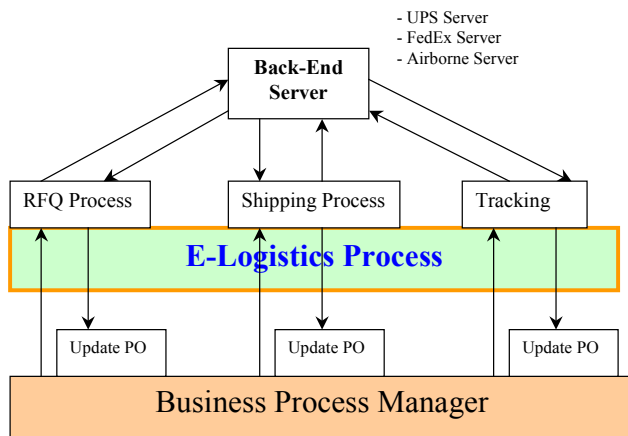


Figure 1. High-Level View of ELPIF.

The business process manager invokes the RFQ process to get the basic services such as getting the quotes in an e-logistics process. Whenever the response is obtained, the purchase order (PO) is

updated. Shipping process is also invoked by the business process manager and upon completion updates the PO. Once goods are shipped, the tracking number is given to the customer and that tracking number is mapped to the PO number in an e-commerce system. Customers can track their shipment with the help of that number. The interaction diagram of e-logistics and business process manager is shown in Figure 1.

## 2.2 ELPIF Components and Services

### 2.2.1 Common Alliance Layer

Common Alliance Layer is a higher level service interface that shields the clients from multiple transportation carriers and provides an abstraction layer of available services. Such interface contains the method signatures of the functions that need to be implemented by the Web Services that are published by different Shipping Service Providers such as UPS, FedEx, and Airborne that directly talk to their legacy applications. These methods require XML as an input and their result is also a XML String. As a multi-carrier connector, the Common Alliance Layer makes the overall shipping processes simpler than current practices because:

- A set of common interfaces are available to all the shipping carriers, and they would ease the work of a service requestor who is required to issue a single service request using standardized interfaces as opposed to composing and sending several complex requests to multiple targeted service providers.
- It would also ease the job of the client application developers who would have to just code once for all the Shipping Service Providers.
- From a shipping carrier's perspective a set of standard interfaces provided by Common Alliance Layer of ELPIF can be used as a means to increase the customers' awareness of their quality services and products in terms of reliability and efficiency.
- It would reduce development effort by alliance layer reuse and allow easier adaptation to new service requirements or technologies.

The common interface is implemented by all the shipping carriers and the resulting Web Services is published in the UDDI Registry so that trading partners and customers can search and retrieve those services.

### 2.2.2 Adaptation Layer

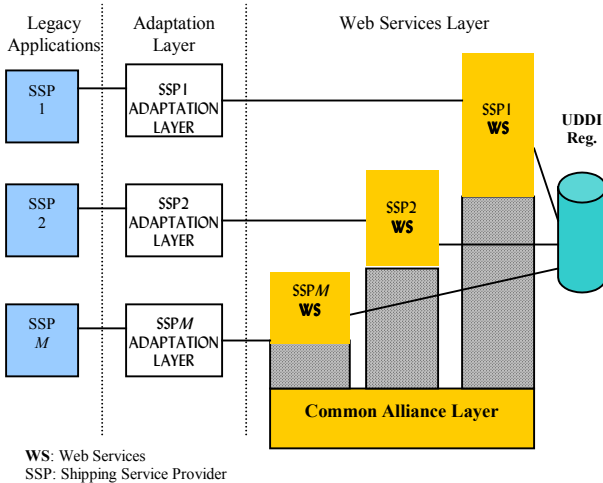
Adaptation layer is the key connector between the Web Services and the legacy applications of an industry. The adaptation layer works as a service dispatch broker and service aggregation broker --- it is responsible for manipulating the request from the user and response from the server. When a user invokes a Web Service, the request is sent to the adaptation layer which then does the method signature mapping between application client and Web Services methods, protocol transformation (e.g. from SOAP to HTTPS), and dispatches it to the appropriate shipping carrier server (legacy application). Response from the server is aggregated by this layer and is sent back to the requestor. Adaptation layer plays a key role in aggregation of responses - resulting from a single request that was split and sent to multiple Service Providers- into a single response back to the requestor.

### 2.2.3 Dynamic Data Binding

Adaptation layer binds the “dynamic data” i.e. the live and updated data from the Shipping Service Provider’s server to the response XML.

### 2.2.4 ELPIF Architecture

The ELPIF architecture (Figure 2) shows the interaction between the different layers .The Web Services of different Shipping Service Providers provide implementation to the methods that are defined in the Common Alliance Layer. The user finds the appropriate service with the assistance of UDDI Registry and sends its XML-based service request to invoke a service. The request is then sent to the appropriate Shipping Service Provider (SSP) server and response (XML String) is sent back to the requestor by the adaptation layer and Web Services Layer.



**Figure 2. E-Logistics Processes Integration Architecture**

For example, for RFQ Web Service, the Common Alliance Layer could contain a method such as:

```
public String getServicesQuotes(String xmlInput)
```

This method could be implemented in the Web Services of the various Shipping Service Providers. This method would then be called by the Service Requestor to obtain quotes from different providers. This method takes XML String as an input and returns XML String as an output. Example of such Request and Response XML is given later in the paper.

ELPIF serves as a service model using a design pattern in which all Web Services are built on the Common Alliance Layer and the requestor invokes a desired Web Service after finding an appropriate one. The whole complexity of the interactions among the Web Services, Adaptation Layer and the legacy applications are transparent to the application developer or the requestor of the service.

ELPIF provides each engaged Shipping Service Provider three Web Services in order to connect their existing applications to the shipping e-marketplace that is empowered by UDDI registries and Web Services: (1) RFQ Web Services, (2) Shipping Web Services, and (3) Tracking Web Services. They are briefly described as follows.

### 2.3 RFQ Web Services

Any B2B application can send a request to the RFQ Web Services. The RFQ Web Services then dynamically bind the data entered by the requestor such as shipping destination, weight to the input XML template and sends the request to the adaptation layer. The adaptation layer dispatches the request to the appropriate server and gets response from the backend server. It then binds the live data received from the server with the response XML template and sends it back to the B2B application. The authentication of user will be verified at each step by examining the user id and password that are associated with the application. This RFQ process is actually used by the service requestor to compare the different available services so she can select the most favorable one.

### 2.4 Shipping Web Services

After the user selects the transportation service provider, the next process is sending a shipping request to the shipping Web Services. The Shipping Web Services provided by service provider then dynamically bind the data entered by the service requestor such as shipping destination, weight to the input XML template and sends the request to the adaptation layer. The adaptation layer then sends the request to the appropriate server and gets response from the server. It then binds the live data received from the server with the response XML template and sends it back to the B2B application.

The customer will be given a tracking number embedded in the response. Once the goods are shipped, the tracking number is mapped to the purchase order ID in a B2B application. Similarly, authentication of the service requestor application is performed by verifying her user id and password.

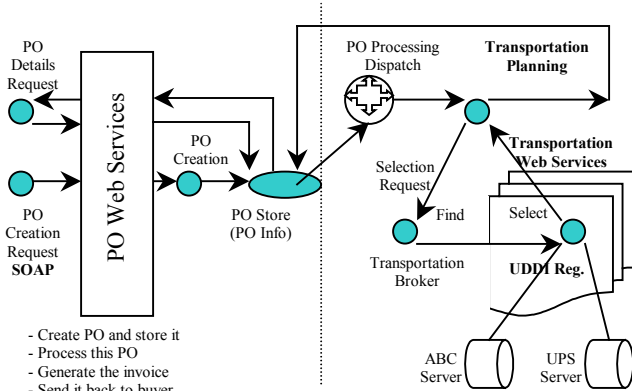
### 2.5 Tracking Web Services

The supplier, buyer or any parties in the supply chain may want to check the status of the shipment corresponding to a specific purchase order. After the commerce server gets the shipping status from the Tracking Web Services, it will update the manifest information in the purchase order database. The Tracking Web Services talk to the backend server to retrieve the detailed shipping status through adaptation layer and hence can track the status of the shipment.

## 3. E-LOGISTICS EXAMPLE: UPS INTEGRATION

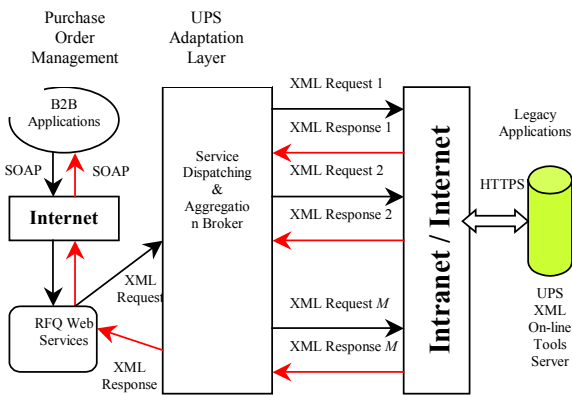
ELPIF promotes online enterprises to create new Web Services and also help them to leverage exiting ones. ELPIF enables all the enterprises with various platforms to be integrated into a common platform and interact with one another. ELPIF is particularly suitable for small- and medium-size companies that often seek a quick and cost-effective solution for business process integration.

The left portion (left to the dotted line) of Figure 3 represents the creation and details checking of a purchase order.



**Figure 3. Business process integration diagram**

Once we have created the purchase order, it will be processed further. One issue that needs to be looked into is the transportation planning. This is a vital issue as it is concerned with many factors such as lowest shipping cost, easy availability, and so on. Transportation broker is an intelligent agent that dynamically creates response list from the available service providers such as UPS and FedEx who meet the criteria of the requestor most efficiently. All shipping carriers should have their own Web Services published in the UDDI registry. The selection request made by the customer is sent to the transportation broker which finds the appropriate service from the UDDI Registry and the result is bound back to the requestor. The transportation Web Services may include ABC Transportation Web Services, UPS Web Services.



**Figure 4. UPS integration architecture (RFQ)**

Figure 4 depicts the integration architecture of a shipping carrier. UPS is taken as an example of a shipping carrier. UPS Web Services like RFQ, Shipping, Tracking are explained in this paper though there can be many more services such as Warehouse management, transportation management and so on. These Web Services interact with the UPS XML Online tools.

When a Web Service, such as RFQ Web Service for UPS, is invoked by a SOAP call it binds data such as country of dispatch, source, and weight to the XML Request template. XML Request is then sent to the Adaptation layer. This adaptation layer works as a service dispatch broker and service aggregation broker. Once it receives an XML Request, which might be the aggregation of multiple requests such as lookup for Next Day Air Service by UPS, Standard service, it sends these requests to the UPS Server.

The responses received are aggregated by the adaptation layer and after mapping the live data received to the response XML template, it sends the response back to the requestor.

The Request XML template can have following structure:

```
<?xml version="1.0">
  <AccessLicenseNum>123</AccessLicenseNum>
  <UserId>pyadav</UserId>
  <Password>2ppp4</Password>
  <Shipment>
    <SourceAdd>
      <City>Charlotte</City>
      <State>NC</State>
      <PostalCode>28213</PostalCode>
      <CountryCode>US</CountryCode>
    </SourceAdd>
    <DestinationAdd>
      <City>White Plains</City>
      <State>NY</State>
      <PostalCode>10603</PostalCode>
      <CountryCode>US</CountryCode>
    </DestinationAdd>
    <ServiceCode>11</ServiceCode>
    <ServiceCode>14</ServiceCode>
    <PackageWeight>100</PackageWeight>
    </UnitOfMeasurement>lbs</UnitOfMeasurement>
  </Shipment>
```

This request is dispatched to the appropriate server by the adaptation layer and it sends the response received to the requestor. If the request is an aggregation of multiple services such as Quote for NextDayAir and Ground Service, the response from the server is aggregated by the adaptation layer and is sent back to the requestor of the service. Service code number is mapped to the Service Name.

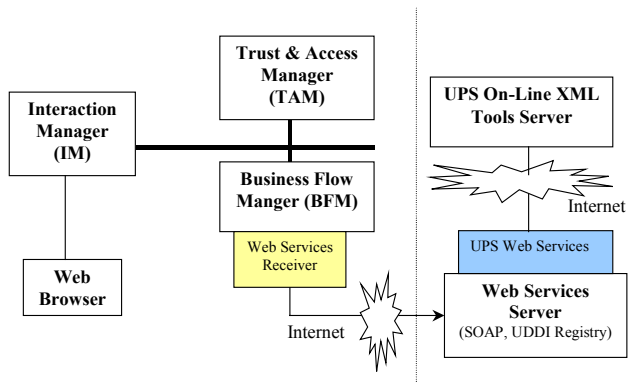
The Response XML template can have following structure:

```
<?xml version="1.0">
  <Response>
    <PackageWeight>100</PackageWeight>
    <UnitOfMeasurement>lbs</UnitOfMeasurement>
    <RatedShipment>
      <ServiceCode>11</ServiceCode>
      <Charges>
        <CurrencyCode>USD</CurrencyCode>
        <MonetaryValue>300</MonetaryValue>
      </Charges>
      <DaysToDeliever>1</DaysToDeliever>
      <DelieverTime>10:00AM</DelieverTime>
    </RatedShipment>
    <RatedShipment>
      <ServiceCode>14</ServiceCode>
      <Charges>
        <CurrencyCode>USD</CurrencyCode>
        <MonetaryValue>250</MonetaryValue>
      </Charges>
      <DaysToDeliever>2</DaysToDeliever>
      <DelieverTime>11:00AM</DelieverTime>
    </RatedShipment>
  </Response>
```

#### 4. A WORKING B2B SYSTEM USING ELPIF

The B2B system is shown on the left side of Figure 5. The detailed description can be found in [3].

The Interaction Manager (IM) houses solution parts responsible for driving the client interaction. This is a model-view-controller framework consisting of a set of Java Beans, JSP (Java Server Pages) templates, and Servlets. The Trust and Access Manager (TAM) houses the Organization Model. These form the basis for client authorization, which is the process of determining whether an authenticated client has the right to perform an operation on a specific resource in a secure domain. The Business Flow Manager (BFM) externalizes the flow definitions (control flows and data flows) and the business rules that drive the process choreography.



**Figure 5. Working B2B system using e-logistics.**

Besides the above mentioned B2B platform, there is a Web Services Server including a SOAP server and a private UDDI registry. As an example, UPS Web Services is deployed and registered on it.

Request from the Web browser will be sent to the TAM first to finish the single-sign-on (SSO) process. Then the subsequent pages will automatically attach the user credential information to the BFM. The BFM will invoke a command to search UDDI server or invoke a Web Service based on the business context.

Figure 6a illustrates the screen of the details of a specific PO. This information is derived from the purchase order database.

[Supplier Portal -> PO Details - Select Transporter](#)

PO ID: PO-06262001-0938-59593

GCB Ref Number	Customer Name	Dispatch Country	Buying Delivery Terms	Port of Loading	ETD	Qty	Article Description	Supplier Article No	Transport Service
30-350.993562740812/1/1	Fabulous Furniture Outlet	USA	FOB	new york	2001-10-11	12	bookcase	100	

Look Up Transporters

**Figure 6a. Purchase order processing (before)**

The Dispatch Country is USA and the port of loading is New York. Based on this information, the transportation broker will search the UDDI registry to get a transporter list of service providers in the United States. For each service provider, they may provide multiple services. For each service, the user can invoke the Web Services to get the quotes for this service, for example.

Figure 6b illustrates the response from three service providers: ABC Transportation Inc, Sunshine Transportation Inc and UPS.

[Supplier Portal--> Transporter Quotes List for PO](#)

PO ID: PO-06262001-0938-59593

Transporter	Service Name	Service Invoked	Quote (USD)	Select
ABC Transportation Inc.	ABC_Transport_Service	getTransportationQuote	2.4570808	<input type="radio"/>
Sunshine Transportation Inc.	Sunshine_Transport_Service	getTransportationQuote	2.5681615	<input type="radio"/>
UPS	UPS Next Day Air	getTransportationQuote	309.57	<input checked="" type="radio"/>

Accept

**Figure 6b. Transporter look up, quotes & selection**

After the user review the quotes, he/she can select one as a transportation service provider for this specific purchase order. Then the purchase order information will be updated. The updated purchase order information is shown in Figure 6c.

[Supplier Portal PO: Selected Transporter Review](#)

PO ID: PO-06262001-0938-59593

GCB Ref Number	Customer Name	Dispatch Country	Buying Delivery Terms	Port of Loading	ETD	Qty	Article Description	Supplier Article No	Transport Service
30-350.993562740812/1/1	Fabulous Furniture Outlet	USA	FOB	new york	2001-10-11	12	bookcase	100	UPS

This PO is finalized. The notification will be sent to the buyer, shipping company and supplier.

Details about UPS Services & Rates

**Figure 6c. Purchase order processing (after)**

At the same time, the notification will be sent to the buyer, supplier and shipping company.

## 5. CONCLUSIONS

In this paper we presented ELPIF, a framework which can be adopted by the shipping carriers or marketplaces and hence shipping services could be designed in a standard way. Shipping Service Providers can make optimum use of their legacy applications and run efficiently with minimal cost input.

We have introduced and explained the concept of Common Alliance Layer, Adaptation Layer and Dynamic Data Binding mechanism. Common Alliance Layer serves as an isolation layer separating the service requestors and service providers and hence reduces complexity by minimizing the number of interfaces to requesting applications. The Adaptation Layer, which is the mediator between the alliance layer and the service providers, manipulates the request and response accordingly.

ELPIF exploits Web Services as the building blocks that distinguish ELPIF from the rest of the existing integration solutions most of which are either standalone applications or hard-wired with existing platforms. Hence, we argue that ELPIF contributes to the area of business process integration by providing a new service delivery model using a design pattern and solution templates, for the shipping industry in particular and whole industry that demands business integration in general.

## 6. REFERENCES

- [1] UPS On-line E-Commerce Tools, <http://www.ec.ups.com>.
- [2] FedEx API, <http://www.fedex.com>.
- [3] Kumar Bhaskaran, Jen-Yao Chung, Terry Heath, Santhosh Kumaran, Raja Das and Prabir Nandi, An e-business Integration & Collaboration Platform for B2B e-Commerce, IEEE Conference, 2001.
- [4] Shipping Automation: Clippership, Kewill, <http://www.kewill.com>