Specifying Conflict of Interest in Web Services Endpoint Language (WSEL)

PATRICK C. K. HUNG
CSIRO Mathematical and Information Sciences
GPO Box 664, Canberra, ACT 2601, Australia
Patrick.Hung@csiro.au

A Web service is an autonomous unit of application logic that provides either some business functionalities or information to other applications through an Internet connection. A business process contains a set of activities, and service locators assign an appropriate Web service for each activity. Recently IBM proposed a new XML language called Web Services Endpoint Language (WSEL) for describing endpoint properties of Web services. Web services will be described through endpoint properties by the appropriate extensibility elements in WSEL in order to facilitate the processes of matchmaking and delegation. It is obvious that security issues are important endpoint properties at Web services. This paper extends the WSEL to specify the conflict of interest in the processes of matchmaking and delegation as one of the endpoint properties. In addition to this, this paper also introduces the concepts of separation of duties and security risk factor in the Web service assignments.

Additional Key Words and Phrases: Web Services Endpoint Language, Conflict of Interest, Separation of Duties, Security Risk Factor.

1. INTRODUCTION

In the past few years, many companies have been forced to reorganize their businesses by using heterogeneous technologies in order to remain competitive in a business world. Current trends in information and communication technology (ICT) may accelerate the widespread use of Web services in business [Aversano et al. 2002]. In this paper, a Web service is defined as an autonomous unit of application logic that provides either some business functionalities or information to other applications through an Internet connection. In many cases, users may want to combine more than one Web service for fulfilling their own needs. Thus Web services must evolve to an environment in which interactions with people and applications, and in which value-added processes are enabled in addition to simple procedures [Rossi 2002]. In particular, value-added Web services are required to be enacted by long duration multi-step activities. Activities represent both business tasks and interactions between Web service providers. The information processed in a Web service might be valued and it is important to protect this information against security threats.

One of the major security problems with Web services is that they often use heterogeneous and distributed hardware and software systems to execute a given activity. This gives rise to decentralized security policies and mechanisms that need to be managed. Since security is an essential and integral part of activities, the Web service has to manage and execute the activities in a secure way. In other words, it is necessary to study the security issues of Web services by their endpoint properties. In particular, Leymann [2001] proposes a new language that is called Web Services Endpoint Language (WSEL) for describing endpoint properties.

Based on the idea of Leymann [2001], this paper extends the WSEL to specify the conflict of interest in the processes of matchmaking and delegation as one of the endpoint properties. In addition to this, this paper also introduces the concepts of separation of duties and security risk factor in the Web service assignments. The remainder of this paper is organized as follows: Section 2 discusses related work in the literature. Next, Section 3 presents the conflict of interest in Web services in the context of WSEL. Lastly, Section 4 discusses the conclusions and future research.

2. RELATED WORK

Web services have become more and more popular in the research community as well as industry. Web Services Description Language (WSDL) is an XML language proposed by Web Wide Web Consortium [W3C] for describing Web services as a set of endpoints operating on messages that contain either document-oriented or procedure-oriented information. A WSDL document defines services as collections of network ports. A port is associated with a reusable binding by a network address, and a collection of ports defines a service. However WSDL only describes the endpoint properties of Web services, it does not consider the execution sequence and security issues of Web service activities (i.e., business processes).

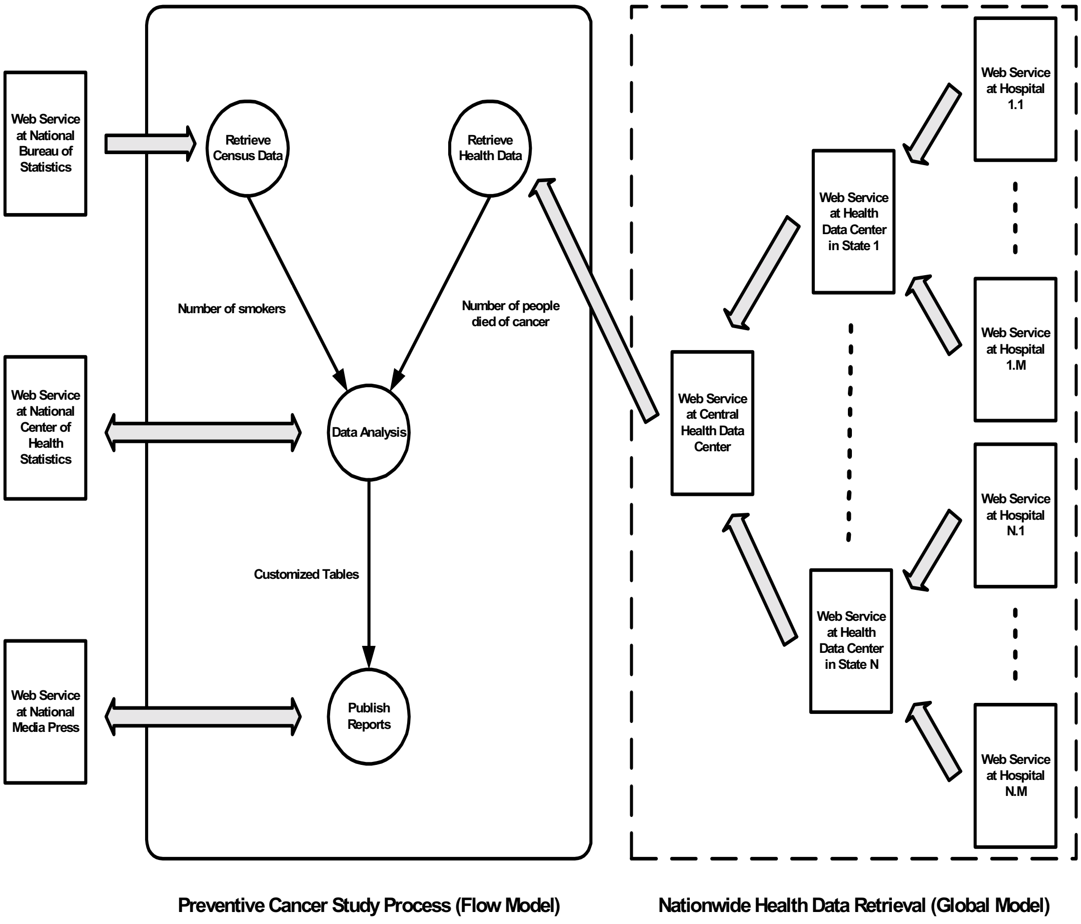
Web Service at National Bureau of Statistics
Retrieve Census Data
Retrieve Health Data
Number of smokers
Number of people died of cancer
Web Service at National Center of Health Statistics
Data Analysis
Customized Tables
Web Service at National Media Press
Publish Reports
Web Service at Hospital 1.1
Web Service at Health Data Center in State 1
Web Service at Hospital 1.M
Web Service at Central Health Data Center
Web Service at Hospital N.1
Web Service at Health Data Center in State N
Web Service at Hospital N.M
Preventive Cancer Study Process (Flow Model)
Nationwide Health Data Retrieval (Global Model)

Fig. 1. An Example of Flow and Global Model.

Thatte [2001] describes XLANG as a notation for the specification of message exchange behavior (i.e., interactions) among participating Web services in business processes. XLANG is based on the WSDL service description with an extension element that describes the behavior of the services as a part of a business process. In XLANG, the behavior of each Web service is specified independently, and the interaction between Web services is only through message exchanges expressed as operations in WSDL. However, XLANG does not specify security issues of Web service activities. On the other hand, Leymann [2001] describes Web Services Flow Language (WSFL) that is also layered on the top of the WSDL. The WSFL is an XML language for the description of Web services compositions. WSFL specifies the appropriate usage pattern of a collection of Web services in order to achieve a particular business goal (i.e., business processes), and it also specifies the interaction pattern of a collection of Web services.

Figure 1 shows an example of flow and global model in WSFL format. The "Preventive Cancer Study" process (i.e., a flow model) is to investigate the demographic and geographical factors of the people (i.e., smokers) who died of cancer. In the flow model, an activity is represented as a circle. WSFL models the execution sequence of the activities as specification of the flow of control and data between Web services. The "Retrieve Census Data" and "Retrieve Health Data" are two activities to retrieve relevant data from outside sources. The "Retrieve Census Data" activity is assigned to a Web service at "National Bureau of Statistics." On the other hand, the "Retrieve Health Data" is assigned to a Web service at "Central Health Data Center." In this example, the Web service at "Central Health Data Center" (i.e., a global model) requires a hierarchical composition of Web services. The global model provides a description of how the composed Web services interact with each other as links between operations of the Web services' interfaces. In this case, the Web service has to retrieve the relevant data from each state (e.g., from *1* to *N*). Further, each Web service at health data center in each state is also composed by a set of Web services at different hospitals (e.g., from *1* to *M*). Once the data from both activities (i.e., "Retrieve Census Data" and "Retrieve Health Data") are received, the consequent "Data Analysis" activity is executed by a Web service at "National Center of Health Statistics." As a result, the Web service generates a set of customized tables. Finally the "Publish Reports" activity is executed by a Web service at the "National Media Press."

OASIS proposes an XML language called Security Assertions Markup Language (SAML) for making authentication and authorization decisions at Web services. This XML-based security information is expressed in the form of assertions about authentication performed by subjects, attributes of subjects, and authorization decisions to access certain resources. Web service providers submit SAML to security servers for requesting authorization decisions. In addition, a Java-based toolkit called JSAML [JSAML Whitepaper] is developed for supporting SAML in e-business applications. However, SAML only considers authentication and authorization in Web Services. SAML does not consider the specific structure of Web services as well as matchmaking between Web service requestors and providers. Recently Leymann [2001] proposes Web Services Endpoint Language (WSEL) for describing endpoint properties, where it matches the expectations from WSFL to the promises from WSDL. Web services are described through endpoint properties and the processes of matchmaking and delegation can be done through service locators.

3. SPECIFYING CONFLICT OF INTEREST IN WSEL

In a flow model, one of the key tasks is *matchmaking*, that is, an appropriate Web service is assigned to execute an activity by a service locator. The service locator may use the service directory [UDDI] to find the most appropriate Web service that can provide the operations to satisfy the activity's requirements. To allow for the corresponding matchmaking, both activities on one side and operations, port types, ports, or services on the other side must be described by endpoint properties. In this scenario, here are four assumptions for the security properties at the Web services:

A1: the Web service is trusted as a means of injecting multilevel security into applications such as command, control, and intelligence systems [Hinke 1989].

A2: the communication channel along the Web service is secure against security threats.

A3: the activity is executed in a secure manner by the Web service.

A4: the information at the Web service is protected against security threats.

In a global model, Web service providers may invoke other Web services in their execution logic. The intermediary Web service providers may also invoke other Web services. As a result, the final Web service providers accept delegation from other Web service providers and the final Web service providers make authorization decisions. The security properties at the endpoint in a global model are inherited along the structure of delegation. Referring to Figure 1, the security properties at the Web service at "Health Data Center" in state *1* to *N* are inherited from the Web service at "Central Health Data Center" and so on. In addition to those four assumptions (i.e., *A1*, *A2*, *A3* and *A4*), here are other two assumptions to be enforced in a global model:

A5: the Web service requestors are eligible to know whether delegation is performed by the Web services being used.

A6: the Web service provides delegation on behalf of requestors (i.e., it passes along the requestors' identity) if and only if it is authorized.

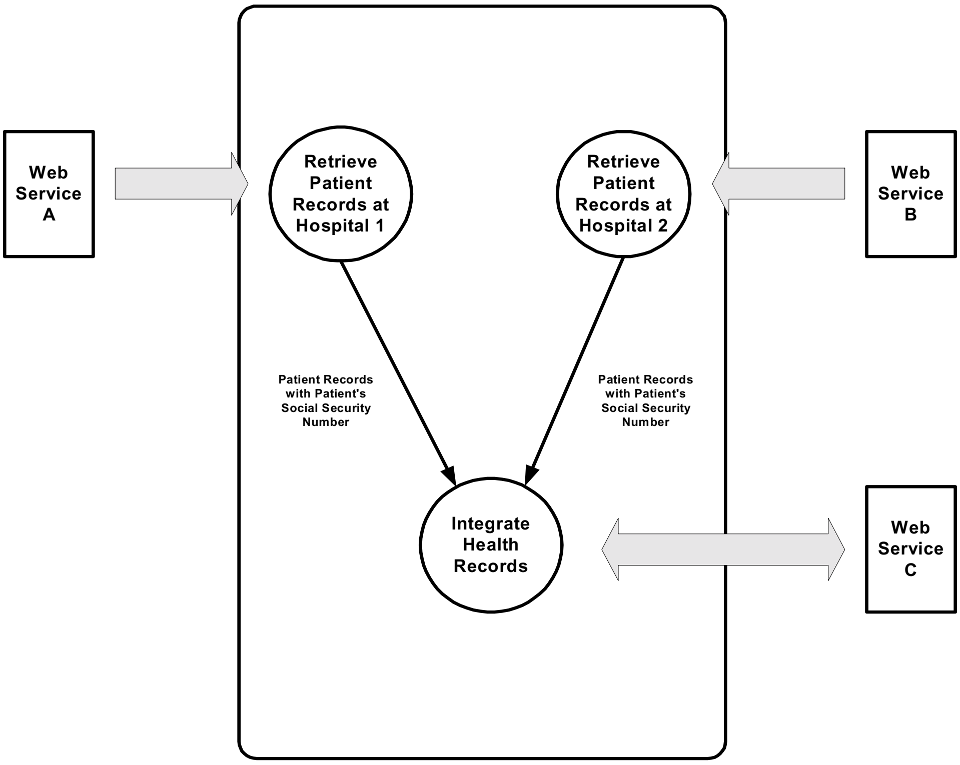
Web Service A
Retrieve Patient Records at Hospital 1
Retrieve Patient Records at Hospital 2
Web Service B
Patient Records with Patient's Social Security Number
Patient Records with Patient's Social Security Number
Integrate Health Records
Web Service C

Fig. 2. Health Records Integration Process.

However, those assumptions are not adequate to ensure a secure execution at the endpoint. In many circumstances, the business process modelers have to know whether there exists any conflict of interest in every activity execution. If a conflict of interest does exist, the business modelers must specify it as a security property at the endpoint. Referring to Figure 2, there is a business process called "Health Records Integration" that includes two activities "Retrieve Patient Records at Hospital 1" and "Retrieve Patient Records at Hospital 2." In these two activities, either Web service *A* or *B* is not allowed to execute the consequent "Integrate Health Records" activity because there is a conflict of interest among those activities. Neither Web service *A* or *B* is allowed to release those records to each other because of patient's privacy. Thus it is required to have another third party such as Web service *C* to execute the "Integrate Health Records" activity. Formally, a business process (BP) is represented into a flow model (FM) that contains a partially ordered set of activities (A) that is coordinated by a set of data/control flows. The order of activity execution is orchestrated by matching the input and output flow(s) of each activity. Each activity represents a piece of work (i.e., a sequence of operations) that needs to be done by a Web service. Each Web service may be a global model that contains a set of Web services (WS) in a hierarchical or peer-to-peer structure. Let entities of a flow model, namely, sets of activities (A) and Web services (WS), respectively, be:

- $A = \{a_1, a_2, \ldots, a_m\}$ is the set of *m* activities.
- $WS = \{ws_1, ws_2, \ldots, ws_n\}$ is the set of *n* Web services.

The relationships among these entities are the following:

- BP: FM → A gives a set of activities decomposed from a flow model.
- C: WS → A gives a set of activities that a Web service is capable to execute. To illustrate, $C(ws_i) = \{a_{i1}, a_{i2}, , a_{ik}\}$ is the set of *k* activities that can be executed by the Web service ws_i, i.e., $C(ws_i) \subseteq A$.
- M: A → WS is a one-to-one mapping that gives a Web service that is assigned to execute the activity. To illustrate, $M(a_i) = ws_i$ is the Web service that is assigned to execute the activity a_i.
- G: WS → BOOLEAN (i.e., true or false) tells whether a Web service is a global model or not. To illustrate, $G(ws_i)$ = "true" means that the Web service ws_i is a global model.
- GM: WS → WS gives a set of activities invoked in a global model if and only if the Web service is a global model, i.e., $G(ws_i)$ = "true." To illustrate, $GM(ws_i) = \{ws_{i1}, ws_{i2}, , ws_{ik}\}$ is the set of *k* Web services invoked in the global model ws_i.

In particular, the conflict of interest is represented as a sequence of notation in the format of first order predicate calculus. Referring to the example of "Health Records Integration" in Figure 2, the modeler can define the conflict of interest in matchmaking:

$$\mathbf{(M(a_1) = ws_a \Rightarrow (M(a_2) \neq ws_a \wedge M(a_3) \neq ws_{a)}) \wedge}$$
$$\mathbf{(M(a_2) = ws_b \Rightarrow (M(a_1) \neq ws_b \wedge M(a_3) \neq ws_{b)}) \wedge}$$
$$\mathbf{(M(a_3) = ws_c \Rightarrow (M(a_1) \neq ws_c \wedge M(a_2) \neq ws_{c)})}$$

where a_1 = Retrieve Patient Records at Hospital 1," a_2 = "Retrieve Patient Records at Hospital 2" and a_3 = ""Integrate Health Records." As a result, this paper proposes the conflict of interest in matchmaking in the endpoint properties as shown in Figure 3. The element is named "matchmaking" and it contains an attribute "exclusive-or" that specifies a set of activities (i.e., $a_1, a_2, \ldots, a_k$) that has conflict of interest. Moreover, there also exists conflict of interest in delegation (i.e., a global model) as discussed above. The business process modeler is allowed to define an exclusive set that identifies a set of Web services that is not allowable to execute any piece of work in an activity. For example, one of the Web services invoked in a global model is a competitor to the Web service requestor. Let the exclusive set be:

- $ES(a_i) = \{ws_{i1}, ws_{i2}, \ldots, ws_{ik}\}$ is the set of *k* Web services that is not allowed to execute the activity a_i by any means.

```
<activity name="processPO">
    <wsel:duration limit="30" metric="minutes"/>

    <wsel:retry maxNumber="10"/>

    <wsel:escalate>
        <wsel:staff
              who="select PID from Person where skill > 15"
              Invoke="c:\programs\org_query.exe"/>
    </wsel:escalate>

    <wsel:observed>
        <wsel:staff
              who="select PID
                   from Flows
                   where FlowName= "TotalSupplyFlow" "
              Invoke="c:\programs\org_query.exe"/>
    </wsel:observed>

    <wsel:conflict-of-interest>
        <wsel:matchmaking
              exclusive-or="a1, a2, …, ak"/>
        <wsel:delegation
              exclusive-set="ws1, ws2, …, wsk"/>
    </wsel:conflict-of-interest>
</activity>
```

Fig. 3. Encoding in WSEL (Based on the example on pages 83-84 in Leymann [2001])

Thus the business process modeler may specify the conflict of interest in delegation:

$$\mathbf{(GM(ws_i) \cap ES(a_i) = \varnothing) \wedge (M(a_i) = ws_i) \wedge G(ws_i)}$$

It is interpreted as follows. If the Web service ws_i is assigned to execute the activity a_i and the Web service ws_i is a global model, the delegation performed by the Web service ws_i is not allowed to invoke those Web services in the exclusive set. Referring to Figure 3, the element is named "delegation" and it contains an attribute "exclusive-set" that specifies a set of Web services (i.e., ws_1, ws_2, …, ws_k) that has conflict of interest with the activity. Referring to Figure 3, Leymann [2001] introduces four endpoint properties as extensibility elements in WSFL:

- *Execution Limits*: It specifies a duration controlling the maximum time of execution by the <duration> element, and it also sets the maximum number of attempts by the <retry> element.
- *Escalation*: It specifies a contact person to be notified once the thresholds set in <duration> and <retry> are violated.
- *Observation*: It specifies a person who has the right to track the execution of an activity by the <observed> element.

- *Contacts*: It specifies a contact person to be notified once there is any violation in the endpoint properties by the <staff> element.

Separation of duties requires that several parties be involved in performing a specific process independently and no individual party can misuse privileges by acting alone. Under the principle of separation of duties, a complex process is decomposed into several activities, which are executed by different parties (e.g., Web services). In this scenario, the level of separation of duties increases as the number of Web services involved increases. In addition, Hung et al. [1999] introduces the concept of Security Risk Factor (SRF) to separate a set of tasks (i.e., activities) to a set of agents (i.e., Web services) as evenly as possible in order to increase the level of separation of duties. Essentially, the SRF measures the level of risk associated with a set of agents executing a set of inter-dependent tasks. For a simple illustration, there is a set of Web services executing a set of activities. Hence in order to reduce the SRF one needs to assign the set of activities across as many appropriate Web services as possible. This partitioning of a set of activities into disjoint sub-sets of activities, each having as few activities as possible, and each disjoint sub-set being done by a single Web service, provides for lower SRF. Though this paper does not investigate the SRF theory in a full scale, further details can be found elsewhere, e.g., Hung et al. [1999]. However, one may imagine that it is possible to have an element for SRF with different levels (e.g., 1, 2, …, 10) in WSEL.

```
    <wsel:security-risk-factor value="5"/>
```

Here is an example to demonstrate a situation where the level of separation of duties increases while the level of conflict of interest increases. In consequence, the level of SRF decreases. As a first cut, this example assumes that a global model is theoretically identical to a single Web service. Let the set of Web services involved in a business process bp_i be:

- $S = \{ws_{i1}, ws_{i2}, \ldots, ws_{ik}\}$ is the set of *k* Web services that is invoked in the business process bp_i, where $\forall_{j=1..k}\, C(ws_{ij}) \cap FM(bp_i) \neq \emptyset \wedge M^{-1}(ws_{ij}) \cap FM(bp_i) \neq \emptyset$.

In this case, let assume that there is one Web service, say ws_{org}, that is assigned for two activities, i.e., $|M^{-1}(ws_{org}) \cap FM(bp_i)| = 2$. If there is a conflict of interest occurred in those two activities, there are two options that a service locator can perform:

Option I: Assign one of the activities to another new appropriate Web service.

Option II: Assign both activities to two other new appropriate Web services.

In either case, as long as the new appropriate Web service(s) is/are not belong to the existing set S, it will definitely increase the level of separation of duties. Let the new appropriate Web service(s) be ws_{new} (i.e., $ws_{new} \notin S$ for Option I) or ws_{new-1} and ws_{new-2} (i.e., $ws_{new-1}, ws_{new-2} \notin S$ for Option II). In either case, it is obvious that the number of Web services invoked will be increased by one. That is $S' = S \cup ws_{new}$ or $S' = (S - ws_{org}) \cup ws_{new-1} \cup ws_{new-1}$. As a result, it is obvious that $|S'| > |S|$. As a result, the level of SRF will also be decreased.

4. CONCLUSIONS AND FUTURE WORK

Based on Leymann [2001], this paper extends the WSEL security properties to specify the concept of conflict of interest, where it is also related to the concept of separation of duties. Further, we introduce the concept of Security Risk Factor (SRF) that is used to measure the level of risk. This work can be expanded in several directions. We are currently investigating other major security properties regarding the threats against confidentiality, integrity, anonymity and availability in the context of WSEL.

ACKNOWLEDGMENTS

Many thanks to Dr. Kerry Taylor for her valuable comments and suggestions.

REFERENCES

AVERSANO, L., G. DE CANFORA, A. LUCIA AND P. GALLUCCI. 2002. Integrating document and workflow management tools using XML and web technologies: a case study. In *Proceedings of Sixth European Conference on Software Maintenance and Reengineering*, 24 -33.

BERFIELD, A., P. K. CHRYSANTHIS, I. TSAMARDINOS, M. E. POLLACK AND S. BANERJEE. 2002. A scheme for integrating e-services in establishing virtual enterprises. In *Proceedings of Twelfth International Workshop on Research Issues in Data Engineering: Engineering E-Commerce/E-Business Systems*, RIDE-2EC 2002, 134 -142.

BIZTALK ORGANIZATION: www.biztalk.org.

HINKE, THOMAS H. 1989. Trusted server approach to multilevel security. In *Proceedings of Annual Computer Security Applications Conference*, 335-341.

HUNG, PATRICK C. K., KAMALAKAR KARLAPALEM AND JAMES GRAY III. 1999. Least Privilege Security in CapBasED-AMS. *The International Journal of Cooperative Information Systems*, vol. 8, no. 2 & 3, 139-168.

LEYMANN F. 2001. Web Services Flow Language (WSFL 1.0). IBM Corporation.

LEYMANN F., D. ROLLER AND M.-T. SCHMIDT. 2002. Web services and business process management. *IBM Systems Journal*, vol. 41, no. 2, 198-211.

JSAML WHITEPAPER: http://www.netegrity.com/files/JSAMLwhitepaper.pdf

OASIS SAML: http://www.oasis-open.org/committees/security

ROSSI, MICHAEL. 2002. Process Management: A Fundamental Component of Successful Web Service Execution. *Workflow Handbook 2002 WfMC*, 117-132.

THATTE, S. 2001. XLANG - Web Services for Business Process Design. Microsoft Corporation.

UDDI ORGANIZATION: www.uddi.org

WEBMETHODS: www.webmethods.com

WORLD WIDE WEB CONSORTIUM (W3C): www.w3c.org