行政院國家科學委員會專題研究計畫　成果報告

動態複合式電子服務系統平台之研究

計畫類別：個別型計畫
計畫編號：NSC91-2416-H-009-008-
執行期間：91 年 08 月 01 日至 92 年 07 月 31 日
執行單位：國立交通大學資訊管理研究所

計畫主持人：劉敦仁

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 92 年 10 月 27 日

行政院國家科學委員會補助專題研究計畫成果報告

※※※※※※※※※※※※※※※※※※※※※※※※※
※　　　　　　　　　　　　　　　　　　　　　　　※
※　　　動態複合式電子服務系統平台之研究　　　　※
※　　　　　　　　　　　　　　　　　　　　　　　※
※※※※※※※※※※※※※※※※※※※※※※※※※

計畫類別：☑個別型計畫　　□整合型計畫
計畫編號：**NSC 91-2416-H-009-008**
執行期間：91 年 8 月 1 日至 92 年 7 月 31 日

計畫主持人：劉敦仁
共同主持人：

本成果報告包括以下應繳交之附件：
□ 赴國外出差或研習心得報告一份
□ 赴大陸地區出差或研習心得報告一份
□ 出席國際學術會議心得報告及發表之論文各一份
□ 國際合作研究計畫國外研究報告書一份

執行單位：國立交通大學資訊管理研究所

中　華　民　國　　92　年　9　月

行政院國家科學委員會專題研究計畫成果報告
動態複合式電子服務系統平台之研究
Research on System Platforms for Dynamic Composition of e-Services

計劃編號：NSC 91-2416-H-009-008
執行期間：91 年 8 月 1 日至 92 年 7 月 31 日
主持人： 劉敦仁 交通大學 資訊管理研究所
計劃參與人員：廖秋婷、沈民新、吳怡瑾、柯志坤

摘要

企業在網際網路上提供各種電子服務，已是企業電子化之重要趨勢。由不同的服務提供者所提供的電子服務所組成的複合式電子服務，可以為顧客帶來更大的價值。從流程的觀點來看，複合式電子服務成為電子服務提供者與顧客的一種價值流程。本研究提出一個新的複合式電子服務平台，增加新的功能於傳統的電子服務平台。主要包含三項延伸功能：設計電子服務的 Metadata、提供表示複合式電子服務的方法、推薦複合式電子服務的流程。本研究使用 UML 分析語言中的活動圖形表示複合式電子服務，以 ECA 規則控制複合式電子服務流程之執行順序及選擇電子服務提供者。並採用 UDDI 標準，設計適合的屬性以描述電子服務，以作為電子服務之 Metadata，提供語意搜尋及動態選擇電子服務提供者。最後，本研究利用資料探勘技術發掘電子服務之頻繁條件集合，以及發掘電子服務之間的頻繁順序集合，並進而推薦複合式電子服務流程，以提供顧客在選擇電子服務屬性及複合式電子服務流程之依據。

關鍵詞：電子服務；複合式電子服務；工作流程系統；電子化企業

Abstract

Providing various e-services on the Internet by enterprises is becoming an important trend in e-business. Composite e-services, which consist of various e-services provided by different e-service providers, are more valuable for customers. From the workflow viewpoint, composite e-services can be viewed as value-added processes for service providers and customers. This work presents a novel platform capable of supporting e-service metadata, and modeling and recommending composite e-services. Composite e-services are modeled by using the activity diagram of Unified Modeling Language, in which ECA (Event/Condition/Action) rules are employed to control the sequence of e-services enactment and to select e-service providers. Moreover, e-service metadata is designed to extend UDDI standard to enable semantic search and selection of e-services. Finally, data mining approach is proposed to discover the frequent predicates of e-services and the frequent orderings between e-services. Based on the mining result, the proposed platform provides a recommendation of top *N* composite e-services to customers.

Keywords：E-service, Composite E-service, Workflow System, E-business

1. Background and research objective

The Internet became a platform for business transactions recently. Enterprises provide *e-services* via the Internet to generate new revenue or create new efficiencies. Increasing e-services are provided, for instance, on-line subscriptions, on-line payments, on-line travel reservations, real-time news services, etc. Customers can also search and acquire e-services easily on the Internet. Although e-services have received an increasing attention, several issues are still open. Effectively advertising e-services to customers is critical for e-services providers. Besides, individual e-service cannot accomplish a customer’s goal; a complete service generally involves several basic e-services. For instance, a travel service may integrate an airline reservation, a hotel reservation, a car rental, a package delivery and a ticket reservation. Thus, this work aims to aid customers to discover and compose desired e-services.

Hewlett-Packard (HP) investigates several technical details of e-services [7]. *Web service* is another term that is used to define the services provided via the Internet. Web services and e-services are similar but Web services place an emphasis on Web technologies. Several e-service platforms are proposed. For example, HP e-speak [7], IBM WebSphere [8] and Microsoft .NET [11] are such platforms and share many concepts and features. Basic features of these platforms are registering, advertising, monitoring, and managing e-services. E-speak is an open software platform designed specifically for the development, deployment and intelligent interaction of e-services. WebSphere is capable of hosting Web services based on standards such as SOAP (Simple Object Access Protocol) [14], WSDL (Web Services Description Language) [18], and UDDI (Universal Description, Discovery and Integration) [16]. SOAP, WSDL, and UDDI have been accepted as the de facto standards for Web services. UDDI provides directory services for registering and searching e-services. WSDL is a XML-based language used to describe the usage (behavior) of e-services. SOAP is an XML-based protocol for exchanging request/response messages between e-services providers and customers. Detail introductions of these standards can be found in [6, 15]. XML (Extensible Markup Language) [21] has been widely used in electronic commerce, as it uses a flexible, open, and standard-based format to provide interoperability of data exchange over the Internet.

Composite e-services are composed by several e-services, and, intuitively, it can be seen as a workflow. Some vendor protocols for constructing composite e-services are available, such as Web Service Flow Language (WSFL) [19], XLANG [20], and Business Process Execution Language for Web Services (BPEL4WS) [3]. Composite e-service issues are widely discussed. As a workflow is consisted of many tasks, existing approaches, such as Casti and Shan [4, 5], generally model a composite e-service as a process that contains many basic e-services. Balakrishnan [2] proposed a Service Framework Specification to compose e-services. Moreover, Piccinelli and Mokrushin [12] proposed a DysCo model that employs an ontology-based approach to describe the semantics and characteristics of a service. Their investigations do not provide recommendation facilities to help the design of composite e-services. Additionally, existing recommender systems, e.g., [10,13], focus on recommending top *N* relevant items (documents or products) regarding a given item. However, this work advises the flow schema of a set of given items (e-services). The objective of this research is the following.
(1) Investigate how to dynamically compose and enact composite e-service. A system platform will be proposed to manage composite e-services; (2) Analyze and design appropriate metadata for describing the characteristics of e-services. The metadata will be integrated into the system platform to facilitate dynamic composition and enactment of composite e-services. (3) Investigate the recommendations of composite e-services.

2. Research result

The main research results are summarized as follows.

(1) This work designs metadata of e-services that describes the characteristics of e-services. The metadata of e-services can be described by adding attributes of e-services to UDDI (Universal Description, Discovery and Integration) standard [16]. Thus, customers can discover e-services through semantic predicates rather than content-independent queries, since providers can expose more meaningful description of e-services by using the proposed metadata.
(2) This work proposes a workflow model to represent composite e-services, closely like workflows. Standard UML (Unified Modeling Language [17]) activity diagram and ECA rules [9] are adopted to model composite e-services. The model describes the flow schema of composite e-services, and provides an event-driven way to control the execution of basic e-services by using ECA rules.
(3) This work proposes a novel data mining approach to recommend the means of composing customer-selected e-services. This work analyzes instance execution logs by using association rules to discover frequent predicates of e-services and frequent orderings between e-services for basic recommendations. Furthermore, the mining results are used to score default flows of composite e-services. The top *N* composite e-services are discovered for advanced recommendations. Based on the mining result, the proposed platform can recommend the top *N* composite e-services for customers to support their decisions.

3. System platform

3.1 System overview

Generally, an e-service platform provides various mediating facilities for e-service providers and customers. For e-services providers, an e-services platform is capable of *registering* e-services descriptions, *advertising* e-services in directories, *monitoring* e-services and *managing* e-services. For customers, an e-services platform is capable of *searching* proper e-services and *accessing* e-services. Moreover, UDDI is employed to provide directory services for registering and searching e-services. WSDL is used to describe the usage of e-services, while SOAP is employed to exchange request/response messages between e-service providers and customers.

This work proposes three enhancements to conventional e-service platforms, including design of e-service metadata, modeling of composite e-services, and recommendations of predicates of each e-service and orderings between e-services, and composite e-services. The architecture view of the proposed e-service platform is shown in Fig. 1, which shows the components, outbound standards, and users of the platform. Novel features are described below. Besides conventional search, the *search tools* adopt metadata of e-services to provide semantic search. *Recommender* is responsible for recommending the top *N* composite e-services according to the mining results provided by *data mining tools*. E-Service composition is handled by *composite e-service definition tool* and *engine*; the former defines and stores composite e-service definitions in design-time, while the latter manages composite e-service instances in run-time.

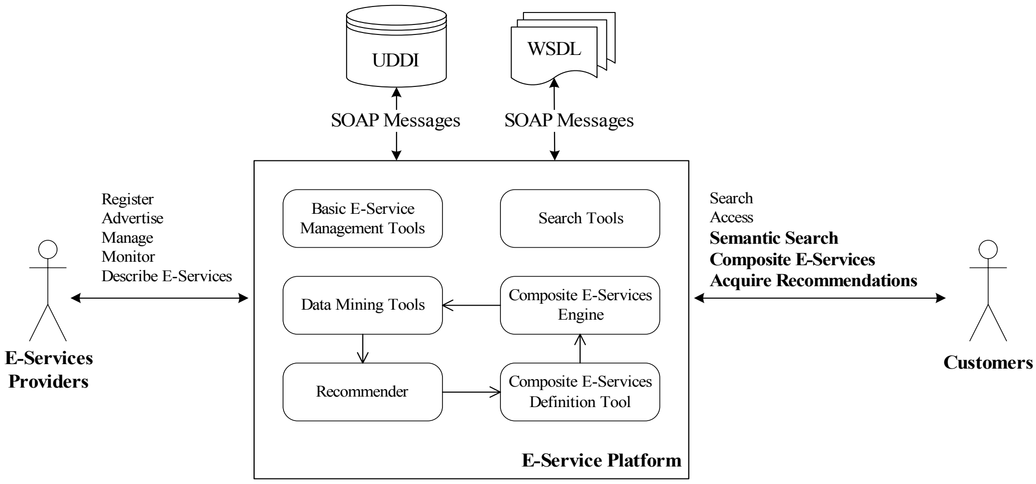

Fig. 1. Architecture of proposed e-service platform

3.2 Metadata design

E-service metadata refers to a detail description about e-services and providers; it is used to advertise or discover e-services in registries. The metadata consists of two types of metadata: business and service level metadata. Business level metadata refers to the description of e-service providers, while service level metadata describes the detail characteristics of e-services. To facilitate the discovery and advertisement of e-services and providers, the enhancement of business and service level metadata are proposed as follows.

Business level metadata

Business level metadata describes the basic information about e-service providers, such as name, phone number, address, and e-mail. UDDI is a repository which contains descriptions of e-services and their providers. The businessEntity [16], one type of UDDI, contains all known information about a business or entity that publishes descriptive information about the entity as well as the services that it offers. Therefore, the data structure of businessEntity can be used to describe the business level metadata.

Service level metadata

The service level metadata contains the characteristics that can describe the detail information about the e-services. Through the service level metadata, customer can identify the content and fulfillment conditions of available e-services. The businessService structure of UDDI represents a logical service classification [16], and only states the name, description and binding information of an e-service. However, the information of businessService in UDDI can be extended to provide semantic search. We add extended e-service metadata as listed below in the businessService type of UDDI to keep compatible with existing standards. *ServiceCategory* refers to high-level metadata. E-services that provide the same services will be grouped in a ServiceCategory; *ServiceLocation* refers to the place of the service; *LocatedNear* refers to the geographic area of the service; *Facility* refers to special offers by the service; *QualityRating* refers to an expandable list of rating properties that may accompany a service; *Guarantees* refer to promises of services; *TotalCapacity* refers to the total numbers the service can provide; *OfferCapacity* refers to the last numbers that the service can provide; *Price* refers to the cost per service.

Using metadata for semantic search

UDDI provides find_business and find_service APIs to search for businesses and services, respectively [16]. The extended metadata, added in businessService, can be searched by XQuery [22] statements. The UDDI find_business API requets and returns a businessList message that matches the specific conditions.

3.3 Composite e-services

A composite e-service, composed by several e-services, is similar to a workflow. A workflow specifies the ordering of tasks, and is generally represented as a directed graph. Similarly, a composite e-service coordinates the enactment sequence of member e-services. We applies UML activity diagram to describe the flow schema of a composite e-service. Additionally, ECA (Event/Condition/Action) Rule provides a flexible event-driven manner to trigger the enactment of activities and select activity performers during workflow run-time. Thus, we use ECA rules to control the e-service enactment and select e-service providers.

Flow schema of composite e-services

The flow schema of a composite e-service should be defined to describe the enactment sequence of its basic e-services. Fig. 2 shows the flow schema of a composite travel e-service as represented by an activity diagram.

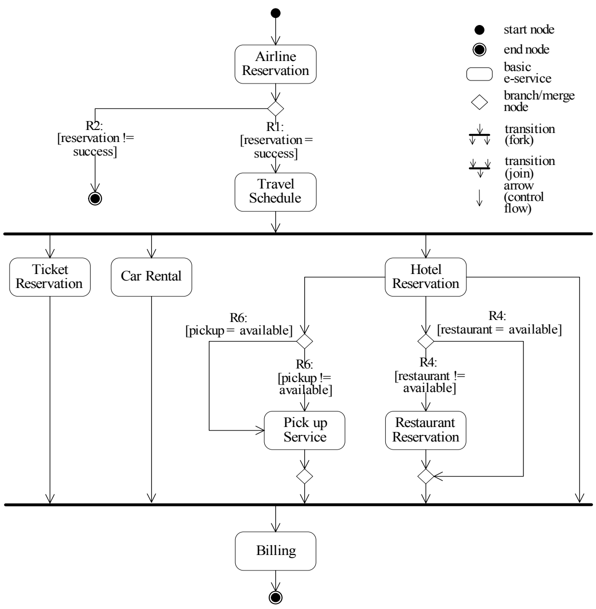

Fig. 2. Composite e-service example

As representing an activity in a workflow, an activity state in an activity diagram denotes a *basic e-service* in the flow schema of a composite e-service. Basic e-services are the e-services provided on the Internet to access. Different e-service providers can provide the same basic e-service. Additionally, a *start* node represents the beginning of this flow, while an *end* node represents the completion of this flow. A *branch* node is a set of transitions leaving a single state such that exactly one guard condition on one of the transitions must always be satisfied. *Arrows* represent the flow dependences and ordering between e-services. A *composite e-services instance* is an enactment of a flow schema of a composite e-service. A composite e-service can be instantiated several times.

Table 1. Syntax of routing rules and e-Service selection rules

(a). routing rule **(R)**	(b). e-service selection rule **(SR)**
E: Events C: Conditions A: Actions	E: Events C: Conditions A: Actions
Define Rule R On ⟨**Events**⟩ Do If ⟨**Conditions**⟩ is True Then Execute ⟨**Actions**⟩ Activated for **preceding Basic E-Service**	Define Rule SR On ⟨**Events**⟩ Do If ⟨**Conditions**⟩ is True Then Execute ⟨**Actions**⟩ Activated for **Notify E-Service Provider**

ECA rules

ECA (Event/Condition/Action) rules have been widely used in workflow management systems as an activity scheduler [9]. This work uses ECA rules to control the routing of basic activities and the selection of e-service providers.

Using ECA rules for routing rules

When the input arrow is fired, a fork node fires all the output arrows in parallel, while a branch node fires the output arrows that satisfy the routing conditions. ECA rules can be used to describe the routing decisions. The syntax of routing rules is illustrated in Table 1 (a). In composite e-services, the completion of preceding e-services is regarded as an event of routing rules. For example, Table 2 shows a routing rule R1 derived from Fig. 2. The routing rule R1 is fired after the completion of Airline_Reservation and the reservation is successful. R1 then notifies a travel schedule provider to prepare Travel_Schedule e-service.

Table 2. Routing rules of composite travel e-services

Define Rule R1 On **Airline Reservation completed** Do If **reservation = success** is True Then Execute**Travel_Schedule_Service;** **Notify_Travel_Schedule_Provider** Activated for Airline Reservation
Reservation = success depends on the AirlineReservation:OfferCapacity; if the OfferCapacity is enough then Reservation = success

Using ECA rules for e-service selection rules

We can also use ECA rules as e-service selection rules. Each e-service associates with an e-service selection rule that determines an e-service provider. Table 1 (b) is the syntax of e-service selection rules, and Table 3 shows the e-service selection rules of basic e-service in Fig. 2.

Table 3. E-service selection rules

SR2: Travel Schedule Selection
Define Rule SR2 On **Pre Notify_Travel_Schedule_Provider** Do If **SystemSelection** is True Then Execute **XQuery for Travel Schedule Provi. Selection;** **Add: Qualified_Providers; SystemSelection** Activated for **Notify_Travel_Schedule_Provider**
XQuery for Travel Schedule providers selection:

```
 For $esp in EZTravel-UDDI//businessEntity
where $esp//serviceExt:ServiceCategory
  ="Travel_Schedule" and
   $esp//serviceExt:Facility="Personal Monitor"
     and
   $esp//serviceExt:Price=50
return $esp
```

Take SR2 for example, before notifying a travel schedule provider, SR2 executes an XQuery to identify the Qualified_Providers, that is, the providers who fit the query predicates. An e-service selection rule executes an XQuery to discover several e-service providers who satisfy the predicates. The service selection rule further needs a selection policy that is capable of selecting one from the qualified e-service providers during run-time.

3.4 Recommend composite e-services

The customer must specify the ordering and predicates of selected e-services to define a composite e-service. This work proposes a novel mining approach to simplify the composition step. The extended e-service platform can recommend the predicates of each basic e-service, the ordering between e-services, and the top *N* matching composite e-services that contain the desired e-services. To provide advanced recommendations of complete composite e-services, this work uses a scoring approach to recommend the top *N* composite e-services for customers.

This work uses a data mining approach to acquire *frequent predicate sets* of each basic e-service from the Instance Execution Log Database. The frequent attributes of basic e-services are termed as frequent predicate sets in this work. This work also acquires *frequent ordering sets* from the Instance Execution Log Database. The frequent orderings between e-services are termed as frequent ordering sets. Notably, the *Flow Schema Database* stores the existing composite e-service definitions. The *Instance Execution Log Database* records previous executions of flow schema definitions. A flow schema may be instanced several times. Each instance can be stored as a log.

3.4.1 Mining instances of composite e-services

This work employs Apriori algorithm [1] to find frequent predicate sets and frequent ordering sets from instance execution logs.

Mining frequent predicate sets

The system can recommend the frequent predicate sets of each basic e-service. According to the log data, this work uses the Apriori algorithm [1] to discover the frequent predicate sets. *Support* of a predicate set *PS*, denoted by *sup*(*PS*), is the ratio of those instances that contain all predicates in *PS*. Predicate sets with support values greater than minimum support value are called *frequent* predicate sets.

Mining frequent ordering sets

The *ordering* between e-services A and C, denoted by ⟨A, C⟩, means that A precedes C in the composite e-services. Obviously, if there exists a path from A to C, then A precedes C. The ordering lists can be acquired according to the instance execution log and flow schema definitions. Every element ⟨X, Y⟩ in the ordering list is a candidate item for deriving frequent ordering sets. The support of an ordering set *OS* is the ratio of instances that contain all orderings in *OS*. Ordering sets with support values greater than the required minimum support values are called *frequent* ordering sets. This work also uses the Apriori Algorithm [1] to generate the frequent ordering sets that satisfy the required minimum support value.

3.4.2 Recommendations

Assume a customer wants to define a new composite travel e-service. He may select several travel e-services from the travel services pool, including travel schedule, hotel, airline and restaurant reservation. According to the mining results, the system acquires the frequent predicate sets of each basic e-service, and acquires the frequent ordering sets between e-services. The extended e-services platform should then recommend these frequent predicate sets and frequent ordering sets to the customers.

Basic recommendations cannot suggest a complete flow of composite e-services. A customer needs advanced recommendations about complete composite e-services. This work uses a scoring approach on advanced recommendations. The Flow Schema Database records the existing composite e-service definitions. We first extract the composite e-services that include the selected e-services. The total score of a composite e-service is the summation of its predicate and ordering scores. Finally, the top *N* composite e-services are recommended to customers according to the total scores.

Predicate score: The system gives each composite e-service a predicate score. The predicate score of a composite e-service is the summation of the predicate scores of its basic e-services, as the following formula.

$$\text{Predicate Score of } CS = \sum_{e \in CS} \sum_{p \in e.predicates} \sup(p)$$

where *e* denotes a basic e-service in a composite e-service *CS*; *p* represents a predicate in *e*.

Ordering score: The system also computes the ordering score of composite e-services. The Ordering score can be derived by the following formula.

$$\text{Ordering Score of } CS = \sum_{<x,y> \in CS.ordering} \sup(<x, y>)$$

where $\langle x, y \rangle \in CS.ordering$ means that the ordering $\langle x, y \rangle$ holds in the flow schema of the composite e-service *CS*.

Total score: The total scores are derived by summing up the predicate scores and ordering scores. A weighted score can be computed by multiplying the scores with corresponding weights. That is, Total Score = ($w_p \times$ Predicate Score) + ($w_o \times$ Ordering Score), where w_p and w_o are the weights of predicate and ordering score, respectively. The system ranks extracted composite e-service based on the total score.

4. Discussions

This work contributes to propose novel enhancements in e-service discovery and composition. First, this work designs an e-service metadata for semantic search. We extend UDDI businessService type with e-service metadata. The metadata allows customers to discover desired e-services based on e-service attributes. Second, the proposed system supports e-service composition for delivering value-added e-service. UML activity diagrams and ECA rules are used to describe the flow schema of composite e-services. Finally, this work proposes a data mining approach for recommending composite e-services. Through mining instance execution logs and referring to flow schemas, the extended e-service platform can recommend the ways of composing the discovered e-services.

5. Project evaluation

We have accomplished 90% of the work described in the proposal. Dynamic composition of e-services is crucial for supporting e-business and e-commerce. Our work will be a basis for further research on composite e-services. Our work not only contributes to further research on the practice of e-business but also contributes to the application of e-service enabled electronic commerce. In summary, we have proposed novel idea, investigated new technology and designed a system platform for composite e-services.

References

[1] R. Agrawal, T. Imielinski, and A. N. Swami, "Mining Association Rules between Sets of Items in Large Databases", *Proc. of the 1993 ACM SIGMOD Intl. Conf. on Management of Data*, pp. 207-216, Washington, D.C., May 26-28, 1993.

[2] R. Balakrishnan, "A Service Framework Specification for Dynamic E-Services Interaction", *Proc. of the 4th Intl. Enterprise Distributed Objects Computing Conference*, pp. 28-37, Makuhari, Japan, September 25-28, 2000.

[3] BPEL4WS, Business Process Execution Language for Web Services, http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/.

[4] F. Casati and M.-C. Shan, "Definition, Execution, Analysis, and Optimization of Composite E-Services", *IEEE Data Engineering Bulletin*, 24(1), pp. 29-34, 2001.

[5] F. Casati and M.-C. Shan, "Dynamic and Adaptive Composition of E-Services", *Information Systems*, 26(3), pp. 143-163, 2001.

[6] F. Curbera, M. Duftler, R. Khalaf, W. Nagy, N. Mukhi, and S. Weerawarana, "Unraveling the Web Services Web: An Introduction to SOAP, WSDL, and UDDI", *IEEE Internet Computing*, pp. 86-93, Mar/Apr 2002.

[7] HP (Hewlett Packard), E-Speak, http://www.e-speak.hp.com/.

[8] IBM Inc., Websphere Application Server, http://www7b.boulder.ibm.com/wsdd/products/platformoverview.html.

[9] G. Kappel, P. Lang, and S. Rausch-Schott, "Workflow Management Based on Objects, Rules, and Roles", *IEEE Data Engineering Bulletin*, 18(1), pp. 11-18, 1995.

[10] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and R. Riedl, "GroupLens: Applying Collaborative Filtering to Usenet News", *Communications of the ACM*, 40(3), pp. 77-87, 1997.

[11] Microsoft, .Net, http://www.microsoft.com/net.

[12] G. Piccinelli, G.-D. Vitantonio, and L. Mokrushin, "Dynamic Service Aggregation in Electronic Marketplaces", *Computer Networks*, 37(2), pp. 95-109, 2001.

[13] P. Resnick and H. R. Varian, "Recommender Systems", *Communications of the ACM*, 40(3), pp. 56-58, 1997.

[14] SOAP, Simple Object Access Protocol, http://www.w3.org/TR/soap.

[15] A. Tsalgatidou and T. Pilioura, "An Overview of Standards and Related Technology in Web Services", *Distributed and Parallel Databases*, 12(2-3), pp. 135-162, 2002.

[16] UDDI, Universal Description, Discovery and Integration, http://www.uddi.org/.

[17] UML, Unified Modeling Language, http://www.uml.org/.

[18] WSDL, Web Services Description Language, http://www.w3.org/TR/wsdl.

[19] WSFL, Web Service Flow Language, http://www-3.ibm.com/software/solutions/webservices/pdf/WSFL.pdf.

[20] XLANG, Web Services for Business Process Design, http://www.gotdotnet.com/team/xml_wsspecs/xlang-c/.

[21] XML, Extensible Markup Language, World Wide Web Consortium (W3C) At URL: http://www.w3.org/XML

[22] XQuery, XML Query, http://www.w3.org/XML/Query.