SESSION

WEB SERVICES + TOOLS AND APPLICATIONS

Chair(s)

TBA

Information, Knowledge, and Task Web Services Using Generic Service Representatives

M. Najafi[1], **K. Sartipi**[2], **and N. Archer**[2]
[1]Department of Computing and Software, McMaster University, Hamilton, Ontario, Canada
[2]DeGroote School of Business, McMaster University, Hamilton, Ontario, Canada

Abstract—*In this paper, we propose an extension to the SOA model by adding a new component called a Service Representative. This SOA component, which simulates an enterprise agent, is modeled by a generic software agent and stays at the client site to be employed by different service providers. Moreover, different types of web services are required to model actual services in the business domain. Consequently, we propose to categorize business services into Information, Knowledge, and Task services. While a service client calls an information service to receive processed data, a knowledge service provides enterprise knowledge for either the service client or service representative. However, a service provider can assign a task to the service representative to process the client's data locally. In this paper, we introduce a prototype implementation of the extended SOA model that we used to develop a healthcare enterprise system which offers different types of services.*

Keywords: Information Services; Knowledge Services; Task Services; Generic Software Agent; Service Representative

1. Introduction

Service Oriented Architecture (SOA) proposes service computing as a solution for enterprise organizations where each enterprise business functionality is provided by one or more services. In the real-world business domain, an enterprise organization usually sends its agent or personnel (e.g., trainer, installer, and maintainer) to the client site to deliver services locally. This role has not been modeled efficiently in SOA. Consequently, the services provided by current SOA are limited to services which process the client's data completely at the server side. This limitation introduces severe constraints in the applications of web services, such as:

- *Security and privacy*: confidential client data are processed at the server side.
- *Response time*: processing the client's data at the server-side may increase the service response time.
- *Required bandwidth*: transferring large client data volumes to be processed at the server-side increases the network traffic.
- *Composability*: collaborating web services can not be composed at the client site.

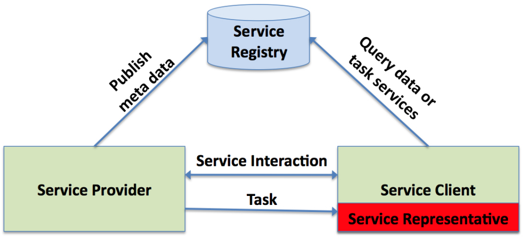


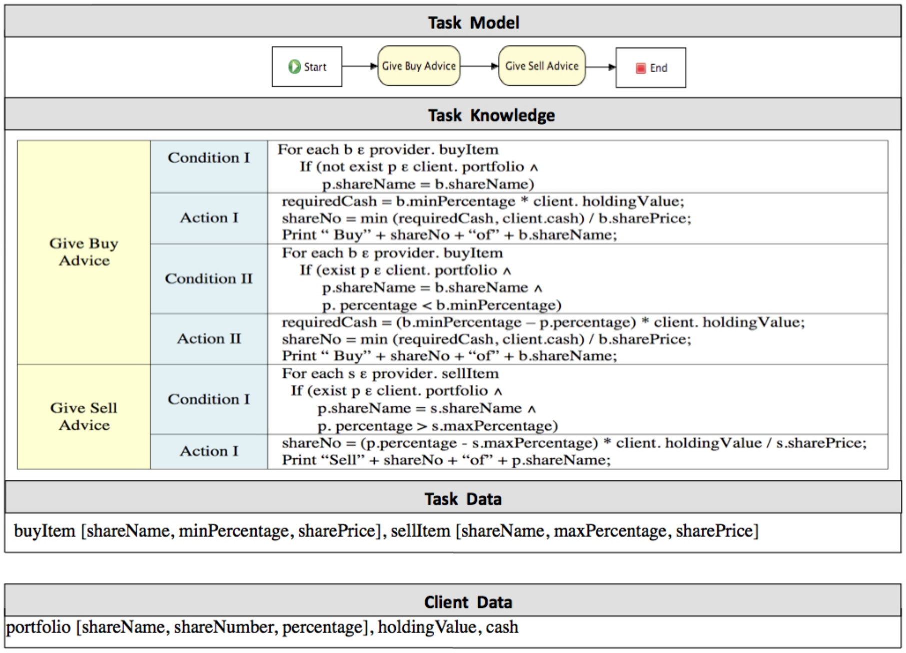
Fig. 1: Proposed extended SOA model. The proposed generic service representative receives tasks from different task services to locally process the client's data.

To have client-side web services, we need to simulate enterprise agents efficiently. In this paper, we propose to simulate an enterprise agent by a generic software agent which resides at the client side and is called the *Service Representative*. Then, we extend the SOA model with this new component (Figure 1). Each business service uses enterprise knowledge to process the client's data and provide information (i.e., processed data) for the client. A traditional SOA model offers *Information Services* where all the client's data processes are performed at the server side and the client just receives the resulting information. However, the extended SOA model enables service providers to process the client's data completely or partially at the client side using a novel type of service called *Task Services*. Moreover, the enterprise knowledge can be provided with *Knowledge Services* to be used by service representatives or clients.

The organization of this paper is as follows. Different types of business services are proposed in Section 2. The proposed architecture is discussed in Section 3 and the developed prototype is described in Section 4. A case study of a health-care enterprise system is presented in Section 5, and related work is discussed in Section 6. Finally, future work is discussed in Section 7.

2. Business Services Categories

A business service is a process of applying enterprise knowledge to the client-side and server-side data to provide information for the clients. As opposed to the traditional SOA models that consider business services to be identical, we propose to categorize the business services into information, knowledge, and task services, as follows.



Task Model

Task Knowledge

Give Buy Advice	Condition I	For each b ε provider. buyItem If (not exist p ε client. portfolio ∧ p.shareName = b.shareName)

	Action I	requiredCash = b.minPercentage * client. holdingValue;
		shareNo = min (requiredCash, client.cash) / b.sharePrice;
		Print “ Buy” + shareNo + “of” + b.shareName;

	Condition II	For each b ε provider. buyItem If (exist p ε client. portfolio ∧ p.shareName = b.shareName ∧ p. percentage < b.minPercentage)

	Action II	requiredCash = (b.minPercentage – p.percentage) * client. holdingValue;
		shareNo = min (requiredCash, client.cash) / b.sharePrice;
		Print “ Buy” + shareNo + “of” + b.shareName;

Give Sell Advice	Condition I	For each s ε provider. sellItem If (exist p ε client. portfolio ∧ p.shareName = s.shareName ∧ p. percentage > s.maxPercentage)

	Action I	shareNo = (p.percentage - s.maxPercentage) * client. holdingValue / s.sharePrice;
		Print “Sell” + shareNo + “of” + p.shareName;

Task Data

buyItem [shareName, minPercentage, sharePrice], sellItem [shareName, maxPercentage, sharePrice]

Client Data

portfolio [shareName, shareNumber, percentage], holdingValue, cash

Fig. 2: A task service to personalize general financial advice based on the client’s personal information at the client site. *Task Model* specifies a task with two steps to customize the general advice (*Task Data*) based on the client’s information (*Client Data*); Finally, *Task Knowledge* defines the required customization knowledge in each step where conditions are expressed in First Order Logic (FOL) and actions are defined using Java statements.

2.1 Information Services

Information services represent the typical web services in the current SOA model where the service provider processes the client’s data completely at the provider site. Therefore, a client has to send its data as web service parameters to the service provider. Then, the service provider returns the generated information as its service response which will be consumed directly by the client.

2.2 Knowledge Services

Enterprise knowledge is a valuable asset for enterprises. However, enterprises can offer this knowledge via knowledge services to be applied remotely by the service representatives or other services. Knowledge management provides techniques to represent, store, transfer, and apply different types of knowledge. For example, descriptive and procedural knowledge can be managed by business rules and actions, respectively. A knowledge service receives a knowledge request from a service client, service representative, or service provider. Then, it uses a knowledge representation technique (e.g., PMML for descriptive knowledge or Dynamic-Link Library for procedural knowledge) to encode and send the requested knowledge. It will be the knowledge receiver’s responsibility to use the knowledge to process the data and generate information.

2.3 Task Services

Task services enable enterprise systems to process the client’s data and resources partially or completely at the client site. In this case, the service provider initially processes the client request and returns a *task* to be performed at the client site. The task will be executed by a service representative and the generated information will be used by the client as the service response. We propose to model a task formally by the following triple.

$$Task =< Model, Knowledge, Data >$$

The *Task Model* specifies a task by an abstract business process model. This model defines the order of applying knowledge to the client-side and server-side data; *Task Knowledge* provides the required knowledge (business rules and actions) to concretize the specified abstract process; and *Task Data* are the server-side data (business objects) that are consumed by the business rules and actions during the business process. A task service performs a task at the client site that implies some of the required data should remain at the client site since they cannot be transmitted efficiently to the provider, such as confidential or large amounts of client data.

A missing part in the current SOA-based models is an enterprise agent that enables a service provider to delegate task services to process client’s data and resources locally. To support task services, we propose to extend the major com-

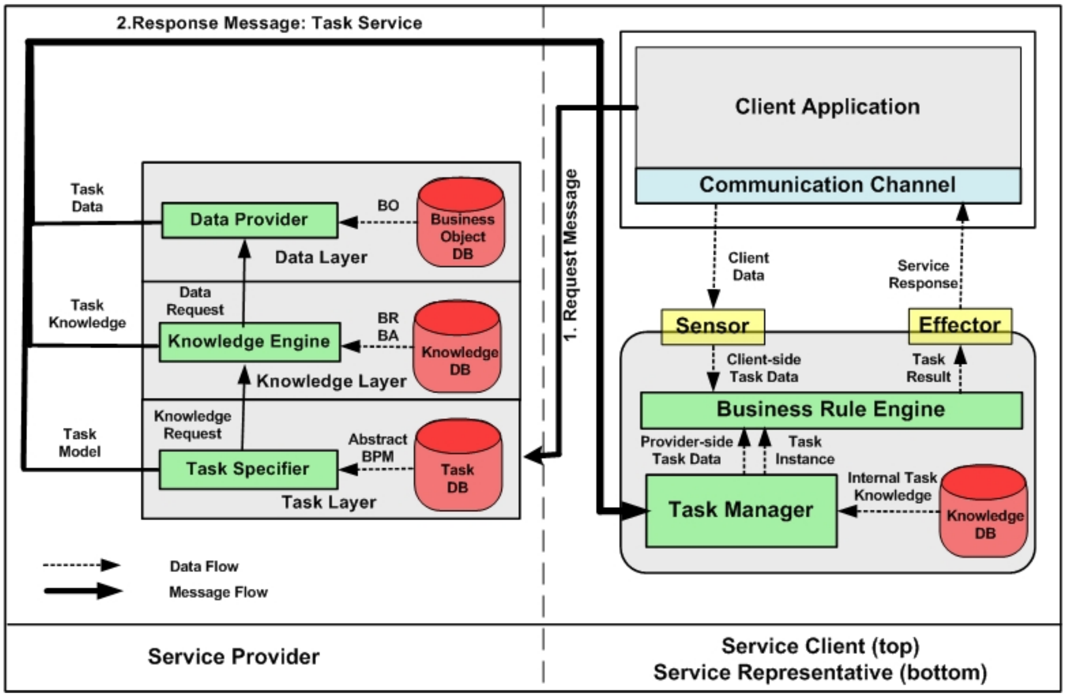


Fig. 3: Proposed extended SOA architecture with service representative and the notion of task services. Based on the client's request message, the provider generates a 3-segment response message (task message) to customize a generic agent as its service representative to perform the assigned task.

ponents of the SOA model (service provider, service client, and service registry) with the generic service representative, as it is shown in Figure 1. In this context, the service providers send *task components* to the service representative to customize it for the processing of the client's data. The service representative executes at the client site with access to the client's local resources, which can potentially violate the client's security and privacy. Therefore, the client controls the required computer resources (e.g., CPU time, storage, and memory) for the service representative.

Figure 2 illustrates different components of a task generated by a task service that provides personalized financial advice without asking the client to send its personal information. This service receives the client's general preferences such as: category of investment (stock, option, or mutual fund); term duration (short term or long term); and risk level (low, medium, or high), while the client keeps the sensitive information local and private, such as the client's financial information (portfolio, saving, debt, and salary). Then, this service generates a set of general financial advice (stock buy and sell advice) according to the client's preferences. A stock buy (or sell) advice recommends the client to have minimum (or maximum) percentage of a specific share in their portfolio. Moreover, it assigns the defined task to the service representative to personalize the generated general advice based on the local client's financial information.

3. Architecture

In this section, we extend the typical architecture of SOA implementations to enable service providers to provide information, knowledge, and task services. The proposed architecture in Figure 3 consists of three main components as follows.

3.1 Service Client

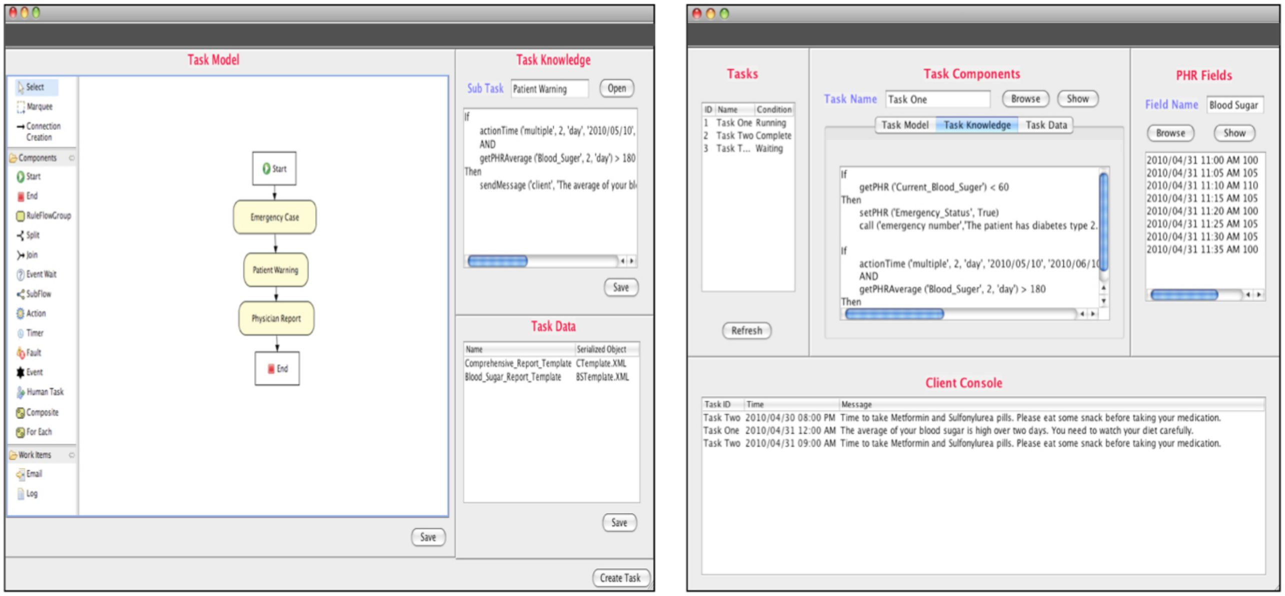
Each service client (or simply client) consists of a client application and a communication channel as follows.

3.1.1 Client Application

This is a traditional client application which can directly call and use information and knowledge services. In case of a task service, the client application puts the required client data and resources into the communication channel based on the schema received from the service registry. This schema describes the type and order of client resources that must be made accessible through the communication channel ports for the service representative. Then it sends a task request message to the service provider and waits to receive the task results through the communication channel.

3.1.2 Communication Channel

This consists of a number of ports that are connection links to the internal resources of the client application, as



Sub Task: Patient Warning

```
If
    actionTime ('multiple', 2, 'day', '2010/05/10',
    AND
    getPHRAverage ('Blood_Suger', 2, 'day') > 180
Then
    sendMessage ('client', 'The average of your bl
```

Name	Serialized Object
Comprehensive_Report_Template	CTemplate.XML
Blood_Sugar_Report_Template	BSTemplate.XML

ID	Name	Condition
1	Task One	Running
2	Task Two	Complete
3	Task T...	Waiting

Task Name: Task One

```
If
    getPHR ('Current_Blood_Suger') < 60
Then
    setPHR ('Emergency_Status', True)
    call ('emergency number','The patient has diabetes type 2.

If
    actionTime ('multiple', 2, 'day', '2010/05/10', '2010/06/10
    AND
    getPHRAverage ('Blood_Suger', 2, 'day') > 180
Then
```

Field Name: Blood Sugar

```
2010/04/31 11:00 AM 100
2010/04/31 11:05 AM 105
2010/04/31 11:10 AM 110
2010/04/31 11:15 AM 105
2010/04/31 11:20 AM 100
2010/04/31 11:25 AM 105
2010/04/31 11:30 AM 105
2010/04/31 11:35 AM 100
```

Task ID	Time	Message
Task Two	2010/04/30 08:00 PM	Time to take Metformin and Sulfonylurea pills. Please eat some snack before taking your medication.
Task One	2010/04/31 12:00 AM	The average of your blood sugar is high over two days. You need to watch your diet carefully.
Task Two	2010/04/31 09:00 AM	Time to take Metformin and Sulfonylurea pills. Please eat some snack before taking your medication.

Fig. 4: (Left): defining a task service using *TaskService* package. (Right): client application GUI, provided by *ServiceRep package*, to call and execute a task service.

well as the means for the client application to receive the result of the requested task through the service representative. The client grants permission to the service representative to read/write a number of its resources through these ports where ports can be input, output, or bi-directional (from the client's point of view).

3.2 Service Provider

Information and knowledge services can be provided efficiently in the traditional SOA model where service responses have only one segment (i.e., either information or knowledge). However, a three layer service provider is required to provide three-segment response messages of task services. These layers are described as follows.

Task Layer. The *task specifier* receives a task service request from a client and retrieves the task model from the *task database*. The task model, represented by an abstract business process model (BPM), describes the process steps to achieve a specific business goal according to abstractly defined business rules and actions. Then, the task specifier sends a request to the knowledge layer to provide the corresponding business rules and actions. The rationale for having a separate task layer is to mimic the situation in real-world enterprise organizations, where the agents are assumed knowledgeable when they are assigned tasks and the required knowledge to perform the task is acquired from other sources (e.g., during the training phase).

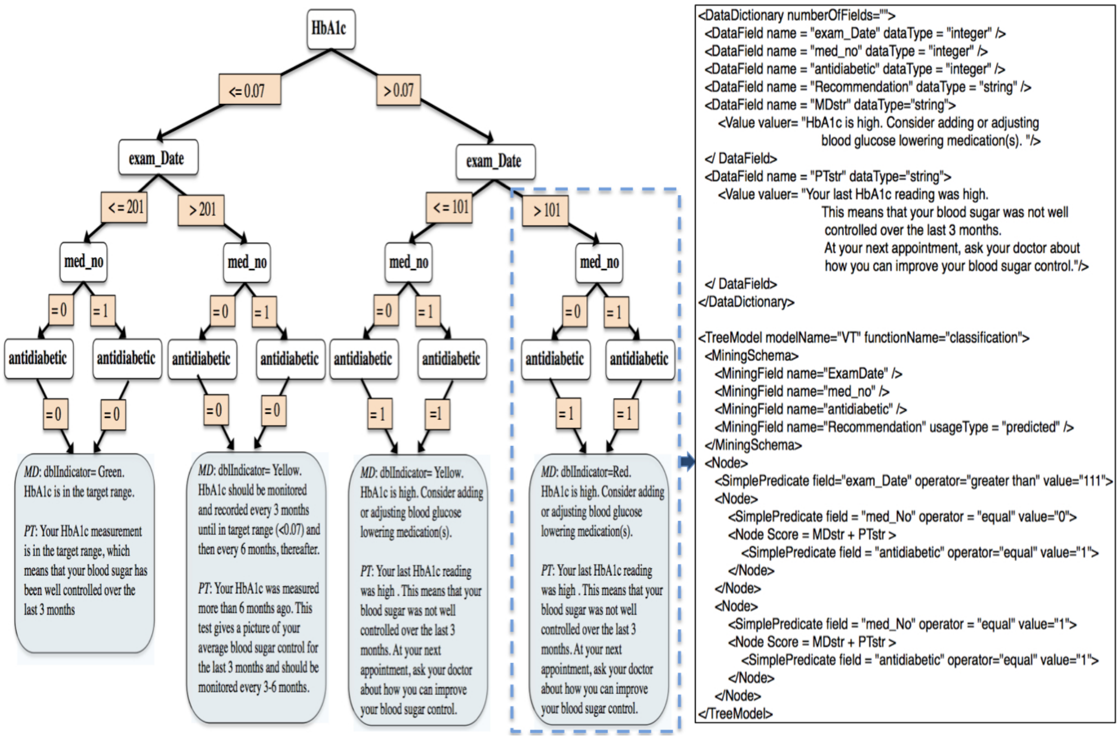
Knowledge Layer. Knowledge layer provides the required task knowledge of a task service, as follows. The *knowledge engine* extracts the requested task knowledge from the *knowledge database* and fills out the task knowledge segment of a task service response message. Task knowledge includes business rules and actions that are abstractly defined in the task layer. A business rule or action may require server-side data that are requested from the data layer.

Data Layer. This layer provides the required server-side data of a task service, as follows. The *data provider* extracts the requested task data from the *business objects database*. Then, the objects will be serialized to be sent by messages and form the task data segment of the task service response message.

3.3 Service Representative

The service representative is modeled by a generic software agent which has a number of *sensors* to receive the client's data through a communication channel and *effectors* to send task results to the communication channel. Other components are introduced below.

- The *Task Manager* supports the entire life cycle of a task instance (i.e., from creation to termination) that is divided into three phases as follows.
    1) *Customization phase*: sets up the agent configuration (including sensors and effectors) based on the task service description published on the service registry and creates an abstract business process based on the task model segment of a task message.
    2) *Training phase*: instantiates a task instance from the abstract process using internal and external task knowledge.



MD: dblIndicator= Green. HbA1c is in the target range.

PT: Your HbA1c measurement is in the target range, which means that your blood sugar has been well controlled over the last 3 months

MD: dblIndicator= Yellow. HbA1c should be monitored and recorded every 3 months until in target range (<0.07) and then every 6 months, thereafter.

PT: Your HbA1c was measured more than 6 months ago. This test gives a picture of your average blood sugar control for the last 3 months and should be monitored every 3-6 months.

MD: dblIndicator= Yellow. HbA1c is high. Consider adding or adjusting blood glucose lowering medication(s).

PT: Your last HbA1c reading was high . This means that your blood sugar was not well controlled over the last 3 months. At your next appointment, ask your doctor about how you can improve your blood sugar control.

MD: dblIndicator=Red. HbA1c is high. Consider adding or adjusting blood glucose lowering medication(s).

PT: Your last HbA1c reading was high . This means that your blood sugar was not well controlled over the last 3 months. At your next appointment, ask your doctor about how you can improve your blood sugar control.

```
<DataDictionary numberOfFields="">
 <DataField name = "exam_Date" dataType = "integer" />
 <DataField name = "med_no" dataType = "integer" />
 <DataField name = "antidiabetic" dataType = "integer" />
 <DataField name = "Recommendation" dataType = "string" />
 <DataField name = "MDstr" dataType="string">
   <Value valuer= "HbA1c is high. Consider adding or adjusting
                  blood glucose lowering medication(s). "/>
 </ DataField>
 <DataField name = "PTstr" dataType="string">
   <Value valuer= "Your last HbA1c reading was high.
                  This means that your blood sugar was not well
                   controlled over the last 3 months.
                   At your next appointment, ask your doctor about
                   how you can improve your blood sugar control."/>
 </ DataField>
</DataDictionary>

<TreeModel modelName="VT" functionName="classification">
 <MiningSchema>
   <MiningField name="ExamDate" />
   <MiningField name="med_no" />
   <MiningField name="antidiabetic" />
   <MiningField name="Recommendation" usageType = "predicted" />
 </MiningSchema>
 <Node>
   <SimplePredicate field="exam_Date" operator="greater than" value="111">
   <Node>
     <SimplePredicate field = "med_No" operator = "equal" value="0">
     <Node Score = MDstr + PTstr >
       <SimplePredicate field = "antidiabetic" operator="equal" value="1">
     </Node>
   </Node>
   <Node>
     <SimplePredicate field = "med_No" operator = "equal" value="1">
     <Node Score = MDstr + PTstr >
       <SimplePredicate field = "antidiabetic" operator="equal" value="1">
     </Node>
   </Node>
</TreeModel>
```

Fig. 5: Left: decision tree corresponding to a VT medical guideline, consisting of three types of nodes: decision, split, and action (leaf boxes). This tree gives recommendations to both patient and physician about the result of a blood test (i.e., Hb1Ac) by considering four patient PHR fields. Right: a part of the generated service response (encoded in PMML V3) by *getGuidelineWS*.

3) *Execution phase*: passes the task instance with the relevant task and client data to the business rule engine to be executed. Finally, the task results are given to the effectors to be written into the communication channel.

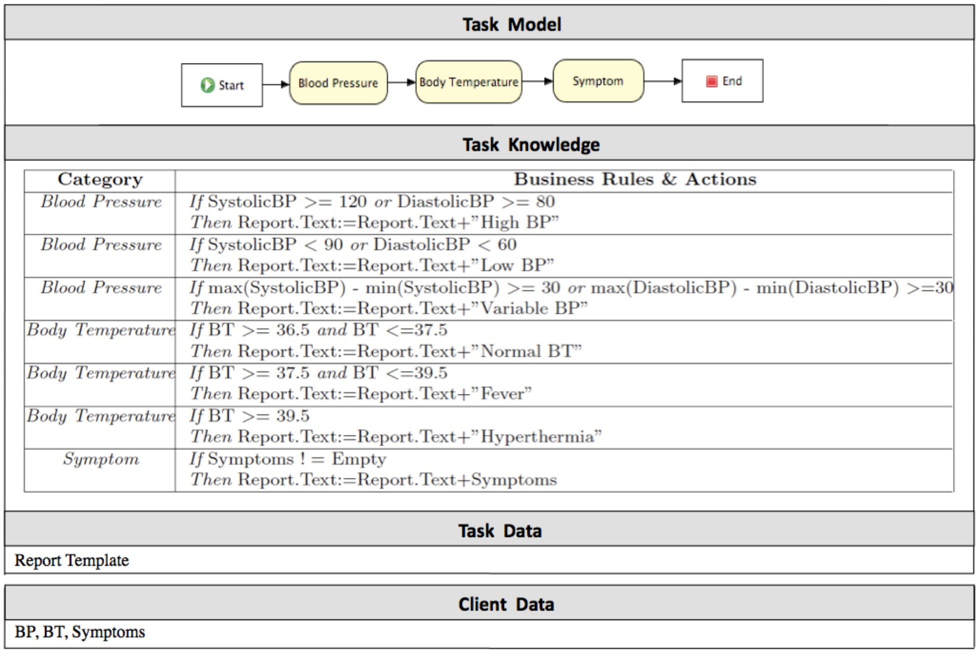
- The *Business Rule Engine* executes a task instance by applying business rules and performing business actions in a defined order.
- The *Knowledge Base* stores domain-based business rules and actions to relieve the service provider from sending them each time.

4. Prototype Tool

We implemented a prototype of the proposed architecture including the service representative, three-layer service provider, and client application. This prototype, namely *Enterprise Representative version 1.0* (EntRep v1.0), is implemented based on J2EE 1.5 technologies. We also used *Drools* APIs [1] to design and execute task services. Drools is a Java-based and open source rule engine that works based on the forward chaining strategy. In addition to the rule engine, Drools has a process engine that allows integration of processes and rules using a process model called *ruleflow*. The EntRep v1.0 uses Drool's rule flow as its task model and Drool's business rules and actions as its task knowledge. Moreover, EntRep v1.0 can receive and understand knowledge sentences that are compatible with PMML (Predictive Model Markup Language) V3. EntRep v1.0 is provided as two Java packages: *TaskService* and *ServiceRep* that can be imported into any service provider and client applications. While a task service developer uses the *TaskService* package to develop a task service graphically, a client application developer uses the *ServiceRep* package to generate one instance of the service representative and communication channel. Figure 4 shows snapshots of two applications developed by these two packages.

5. Case Study: Healthcare Enterprise System

Using the implemented prototype tool (EntRep v1.0), we developed a healthcare enterprise system that offers a variety of information, knowledge, and task services. This system acts as a Clinical Decision Support System (CDSS) to assist physicians and patients with decision-making tasks. Moreover, it preserves the patient's privacy, increases the system availability, and simulates healthcare resources.



Task Knowledge

Category	Business Rules & Actions
Blood Pressure	*If* SystolicBP >= 120 *or* DiastolicBP >= 80 *Then* Report.Text:=Report.Text+"High BP"
Blood Pressure	*If* SystolicBP < 90 *or* DiastolicBP < 60 *Then* Report.Text:=Report.Text+"Low BP"
Blood Pressure	*If* max(SystolicBP) - min(SystolicBP) >= 30 *or* max(DiastolicBP) - min(DiastolicBP) >=30 *Then* Report.Text:=Report.Text+"Variable BP"
Body Temperature	*If* BT >= 36.5 *and* BT <=37.5 *Then* Report.Text:=Report.Text+"Normal BT"
Body Temperature	*If* BT >= 37.5 *and* BT <=39.5 *Then* Report.Text:=Report.Text+"Fever"
Body Temperature	*If* BT >= 39.5 *Then* Report.Text:=Report.Text+"Hyperthermia"
Symptom	*If* Symptoms ! = Empty *Then* Report.Text:=Report.Text+Symptoms

Task Data

Report Template

Client Data

BP, BT, Symptoms

Fig. 6: Virtual Remote Nursing task service. The task model guides the service representative to check the patient's blood pressure, body temperature, and recently observed symptoms, respectively. The task knowledge component provides the required guidelines for each step and the task data provides a report format defined by the patient's physician. The service representative applies this task to the PHR information specified in the client data section.

5.1 Enterprise Specification

The developed healthcare enterprise system incorporates the three-layer architecture presented in Section 3. The data layer has databases containing drug and disease information. Moreover, the knowledge and task layer have databases containing medical guidelines and procedures that belong to Vascular Tracker (VT) [2]. Medical guidelines represent descriptive knowledge in healthcare where each guideline is applied through the patient's Electronic Medical Records (EMR) to generate recommendations for both physician and patient. Similarly, a medical workflow includes some medical guidelines in a specific order. VT is a decision support system that assists physicians to observe and control patient's different risk factors within the domain of cardiovascular diseases. The VT guidelines are categorized into diabetes, hypertension, coronary artery disease, cerebrovascular disease, peripheral vascular disease, and healthy.

5.2 Information Services

The developed system provides several traditional information services. For example, *getDrugInfoWS (drug name)* and *getDiseaseInfoWS (disease name)* return drug and disease information requested by a client. Moreover, *getRecommendationsWS (guideline name, EMR fields)* applies the requested guideline to the received EMR fields and returns the resulting recommendations and diagnosis.

5.3 Knowledge Services

Each VT medical guideline is converted into a decision tree format to be stored in the enterprise knowledge base. Then, the VT medical guidelines can be provided via web services as knowledge services. For example, *getGuidelineWS (guideline name)* returns the requested guideline where it is encoded in PMML standard. The service client can apply this guideline to the patient's EMR data using any PMML engines. Consequently, the service caller always has access to the up-to-date and trusted version of the guidelines. Moreover, in a cloud environment, dynamic CDSSs can be developed using knowledge services from different sources. Figure 5 represents this service response for a VT guideline.

5.4 Task Services

The proposed service representative enables CDSSs to control and process a patient's EMR information remotely at the client's site. This feature introduces several novel services for healthcare systems such as the *Virtual Remote Nursing (VRN)* service. In this case, the service representative is customized by the CDSS to act as a private nurse for a patient (i.e., service client). For this purpose, the client feeds his/her health information (e.g., blood pressure and body temperature) regularly to a PHR system (e.g., Google Health in this case). Moreover, the client enables the service representative to access their health profile through the communication channel. Then, the client's physicians can

Table 1: Comparing different types of web services.

Type	Applications	Pros	Cons
Information Services	Server-side processing of the non-sensitive client's data	- Integrated service logic - Easy to asses the service reliability - Maintaining the enterprise privacy	- Violating the client's privacy - Increasing the network traffic - Increasing the service response time
Knowledge Services	Service-to-Service (S2S) and Service-to-Representative (S2R) transactions	- Maintaining the client's privacy - Reducing the network traffic - Improving the service response time	- Revealing the enterprise knowledge - Client's responsibility to apply the knowledge
Task Services	Client-side processing of the confidential, large volume, or real time client's data.	- Maintaining the client's privacy - Reducing the network traffic - Improving the service response time	- Distributed service logic - Difficult to assess the service reliability

assign different tasks to the service representative by defining the required task components. These tasks include, but are not limited to:

- Make a report of the client's PHR fields over a determined period of time to represent the client's health condition. These tasks guide the service representative about the required PHR fields as well as how and under which condition they should be reported. Figure 6 illustrates an example of these tasks.
- Give proper recommendations and necessary warnings to a patient in specific situations. These tasks include one or more situation specifications as well as the corresponding recommendations and warnings.

Thanks to mobile application platforms, a smart phone can take advantage of this service to guarantee that the client's health condition is always under control. A prototype version of VRN was developed and described completely in [3].

6. Related Work

The proposed work in this paper covers different topics related to web services, software agents, and business processes. A software agent is a piece of software that acts on behalf of an agency to serve a user. Software agents may have the authority to decide which action is appropriate to take. They have been integrated into web services and similarly, web services have been modeled by software agents [4]. Mobile agents can physically travel across a network and perform tasks on different nodes. Therefore, they can be considered as an alternative to the proposed resident generic agents for modelling service representatives. There are several security and privacy issues to be considered in mobile agent computing. Mobile agent architectures also suffer from low efficiency as they need to send the entire computer program or process. These issues motivated us to employ resident agents at the client site as opposed to sending them to the client.

Each business service can be modeled by a business process [5]. The proposed task services divide the service business process into server-side and client-side processing. Then, the client-side portion can be assigned to the service representative agent to be performed at the client-side. Software agent technology has been applied to model and execute business processes. The idea of the agent-based process management system is to split business processes into parts and assign the control over such part to individual software agents [6]. Integration of agents and services is proposed to model the business aspects of enterprise systems, where each role in an enterprise system is considered as an agent [7]. Finally, the proposed architecture extends our previous work in [8] where it introduces a generic software agent that resides at the client side and customizes service responses based on the client's context.

7. Conclusions and Future Work

In this paper, we extended the SOA model to support three types of services. In addition to traditional information services, we introduced knowledge and task services to efficiently model business functionalities of enterprises. However, these web services are complementary and have their own applications and features (Table 1). For future work, we plan to enable service representatives to receive their required knowledge via distributed knowledge services. Finally, collaborating web services will be able to employ the service representative to perform composite tasks at the client's site.

References

[1] M. Proctor, M. Neale, P.Lin, and M.Frandsen, "Drools documentation," JBoss.org, Tech. Rep., 2008.

[2] *Compete III Vascular Tracker website*, http://www.competestudy.com/.

[3] M.Najafi, S.Aghtar, K.Sartipi, and N.Archer, "Virtual Remote Nursing System," in *The 1st IEEE Conference on Consumer eHealth Platforms*, USA, 2011, pp. 101–105.

[4] Y. Cheng, A. Leon-Garcia, and I.Foster, "Toward an Autonomic Service Management Framework: A Holistic Vision of SOA, AON, and Autonomic Computing," *IEEE Communications Magazine*, vol. 46, no. 5, pp. 138–146, 2008.

[5] R. Lu and S.Sadiq, *Lecture Notes in Computer Science*. Springer Berlin/ Heidelberg, 2007, ch. A Survey of Comparative Business Process Modeling Approaches, pp. 82–94.

[6] R.Miller, U.Greiner, and E. Rahm, "Aw: a workflow system supporting rule-based workflow adaptation," *Data and Knowledge Engineering*, vol. 51, no. 2, pp. 223–256, 2004.

[7] L. Xiang, "A Multi-Agent-Based Service-Oriented Architecture for Inter-Enterprise Cooperation System," in *Second Conference on Digital Telecommunications*, USA, 2007, pp. 22–32.

[8] M. Najafi and K.Sartipi, "A Framework for Context-Aware Services Using Service Customizer," in *The IEEE Conference On Advanced Communication Technology.*, Korea, 2010, pp. 1339–1344.

The Commoditization of IT Services with Cloud Computing

F. Muhss[1], **R. Neumann**[1], **and A. Schmietendorf**[2]

[1]Department of Distributed Systems, Otto-von-Guericke University, Magdeburg, Germany

[2]FB II, School of Economics and Law, Berlin, Germany

Abstract - *The commoditization of IT services has reached a new quality with the growing use of cloud services. Based on overarching standards as REST, these services can be accessed on-demand via internet and billed depending on the actual usage at a defined service level. However, compared to other commodities as electricity, cloud services still lack a number of general standards regarding quality and pricing as well as technical definitions to implement these standards. The goal of this paper is to describe and evaluate the current level of commoditization in the cloud industry, leading to recommendations for a continuous success of cloud services.*

Keywords: cloud computing; commoditization; services; SLA; pricing; standards

1 Introduction

Cloud Computing is one of the most discussed IT topics in recent history. Despite concerns about security and availability, analysts expect annual growth rates of almost 22% with a global market volume increasing from 40.7 billion US$ in 2011 to 241 billion US$ in 2020 [1]. One of the drivers behind this development is the standardization of IT services, often described as commoditization in order to emphasize the similarity to other standardized products as gold, electricity or wheat.

Commodities lack qualitative differentiation across an often global market, i.e. that to the customer it doesn´t matter or is unknown who is producing the good. Commodities are often categorized into agricultural products as soybeans or coffee, mining raw materials as copper or silver and energy commodities as oil, electricity and gas. In recent years, products based on intellectual capital also have become commoditized, forming a fourth group containing for example generic pharmaceuticals and even complex technical products as silicon chips.

Main reason for the commoditization is the loss of differentiation of goods or services across a supply base, mostly due to the diffusion of intellectual capital necessary to acquire or produce it efficiently. This spreading of knowledge is often driven by overarching organizations within an industry, defining a set of standards for product quality and terms of delivery to achieve simplified trading of goods. Consequently, these goods formerly carrying premium margins become commodities due to standardization. This leads to a growing competition and importance of price regarding the sourcing of these products.

With the arrival of cloud computing and the concomitant trend to outsource non-core competencies in many industries, the commoditization of IT services has become more important. Especially if services providers fail to deliver the stipulated services as Amazon´s Web Service did in April 2011, it has become vital to possess backup options that are based on similar technologies and standards to allow for on-demand access.

Cloud services have five essential characteristics: on-demand self-service, access via broadband network, resource pooling by providers, rapid elasticity and the capability to measure a service with regards to usage and quality [2]. In order to use cloud services efficiently and reach the aspired improvement of the cost-benefit ratio, cloud providers and users need to agree on transparent standards to allow for on-demand pricing and allocation of resources.

The main areas of standardization of any commodity are the product or service quality, the terms of delivery and the form of contracting including pricing mechanisms. This paper will give a comprehensive overview of current standards already in use in the context of cloud computing with focus on pricing mechanisms. Also, an outlook on what still needs to be achieved to make cloud services a true commodity will be given.

2 Standardizing model of delivery

Delivering IT services via networks has a much longer history than any efforts to describe service levels or prices for the usage of a service. Therefore, standards defining the means of transportation and delivery of IT services are a lot more sophisticated and resilient than quality and price related standards. Namely, the OSI reference model developed for communication networks in the early 1980s has set a structure and standards that still persist. The TCP/IP stack is the dominant transportation model for cloud services. It sets the basis for the transporting of messages between applications (e.g. with FTP, HTTP or SMTP protocols) and the web service stack defining protocols to wrap messages in XML (SOAP), access services (REST), describe services (WSDL) and publish services in a catalogue (UDDI).

It can be said that the delivery of cloud services is a rather mature area of IT. Of course, gradual improvements will always occur, but the right choice of which protocol to use in a certain context or the flexibility to access a service with many protocols seems to have a lot higher effect on the results and success of a cloud service.

3 Standardizing quality

The quality of IT services needs to be described with hard, measurable parameters as well as soft parameters that depend on user perception. Traditionally, hard parameters have been defined, monitored and analyzed since the early days of computing. Availability, response times and utilization are understood by IT experts across the globe. Unfortunately, the definition and measurement of these parameters is not standardized and vary greatly from company to company.

Currently, most cloud users rely on quality parameters defined (and often also monitored) by service providers or define and stipulate SLAs to pledge providers to deliver the desired quality. This is rather simple for parameters as availability, but these SLA are arguable due to unclear measurement or remain a result of individual negotiations. A standard SLA for a specific service across industries describing technical details as well as mode and place of measurement has not been defined as of today. For soft parameters describing the user experience, the situation is even worse. The ease of use, problem handling and user helpdesks based on resilient quality measurements is rarely defined in any service contract.

For cloud services, however, the comparison of the final service quality of competing services is of great importance. This requires a flexible SLA specification and monitoring framework to replace the individual SLAs often described in natural language as part of a service contract. In the past, different XML-based languages to define quality parameters like Web Service Level Agreement [3] or WS-Agreement, standards to describe quality concepts like ITU-T E-800 [4] and ontologies to allow QoS definitions like DAML-QoS have been proposed. They all might be valuable tools to develop and describe machine-readable quality standards, but they don´t suggest any concrete SLA definitions or service parameters for a specific service.

While the requirements towards QoS concepts and SLA definitions are often described in detail, e.g. by [5], the IT industry has so far failed to develop overarching quality standards as SOAP or REST representing communication standards. Organizations and communities like IEEE and the National Institute of Standards and Technology (NIST) are already trying to develop standards based on existing technologies. However, these activities are hardly coordinated and focus mainly on interoperability and portability. It will take some years to come until everyone in IT will have the same understanding of "gold level" for a specific service includes.

4 Standardizing pricing mechanisms

Today, Infrastructure- and Platform-as-a-service (as provided for example by Amazon´s Elastic Cloud or Microsoft Azure) is mostly billed depending on the time a processor is used or the volume of input, output or amount of storage used. Software-as-a-Service (SaaS) is usually offered based on fixed time frames, e.g. per month, year or even just a one-time fee for ongoing usage. These pricing models are simple and transparent, but they are not supported by any formal standard to describe pricing parameters. Usually, the current price is displayed on the respective web page or service catalogue, and at the moment of ordering, a time meter is starting. The final bill is the product of time and the price displayed initially. No standard or ontology to describe whether a service is offered at a fixed price, is available for auctions or another pricing mechanism currently exists. The whole topic seems to be "undiscovered" from a standardization point of view, therefore the underlying pricing models are now analyzed in more detail.

4.1 Overview of Pricing Models

Since the times of traditional bartering in natural economies, many pricing and resource allocation mechanisms have been invented to find the "right price" for a good. The introduction of money allowed people to buy goods independently of what the trade partner was offering. This lead to four common pricing models still prevalent: posted prices and auctions which are very common in commodity markets and tendering as well as bargaining that are rather used for customized products.

4.1.1 Posted Prices

Posted prices are either flat or based on the usage of a certain good. They are defined by the vendor, often depending on production costs, maximal profitability or prices of substitute products. Usage-based fees can relate to the time a certain good was used or the amount of the good that was consumed. In some pricing models, flat and usage-based fees are combined where the flat fee is a means to limit the entry to a market and cover fixed expenses.

Vendors of goods with posted prices often use price differentiation according to certain conditions, e.g. the age of the buyer, the time of the year (or day), the region or the number of buyers, to skim the consumer surplus [6]. Also, they allow consumers to influence the price by choosing product bundles, using volume discounts or buying a certain quality or service level of the good (QoS) which is also called Paris Metro pricing.

4.1.2 Auctions

Auctions were invented due to information asymmetries and the vendors desire to further skim the consumer surplus. They help to determine the customers willingness to pay without the vendor risking to set prices too low or too high. Because of the many different types of goods and markets auctions are applied for, a large number of auction types were invented. The most popular type is an English auction often used for auction of works of art. One seller is offering one specific good to many bidders. The starting price is at the low end, mostly a reserve price is fixed, bidders can bid repeatedly and are overbidding each other until no one is willing to pay more. The winner is paying the whole amount of his last bid.

Other important auction types are the Dutch auction with a high starting price which is constantly lowered until a bidder is willing to pay the price, the First-price-sealed-bid auction where bidders are only submitting one hidden bid and the highest wins, and the Vickry auction which is similar to the English auction except that the winning bidder is paying the price of the second highest bid. The stock market represents a double auction where multiple sellers and bidders are constantly "bidding" to buy or sell multiple, identical goods. Further auction types as Calcutta, American, Walrasian, Smart Market [7], Reverse and Progressive Second Price Auction [8] will be described later on according to their usability in cloud computing.

4.2 Tendering and bargaining

Tendering is a customer-driven process mostly used for rather complex, non-standard products. Public institutions and large companies are obliged to tender large infrastructure projects to fulfill legal requirements. Generally, a consumer describes his demand, e.g. for a machine or a building to be constructed, very detailed and publishes it via respective channels. Potential vendors get the chance to pose clarifying questions and make their offer in the end. Since offers are rarely identical (due to the complexity of the product), the customer needs to decide which is most suitable with regards to time, quality and price.

A hybrid between tendering and auctions is the reverse auction. The customer publishes his demand and takes the role of an auctioneer whereas the bidders are underbidding each other in order to get the contract. Since this model does not include a large time frame to ask questions and concretize the demand, it is not used for complex products.

Bargaining is the oldest form of price fixing. It is applied in almost all settings but auctions, mostly if no price or pricing mechanisms are defined, if a customer does not want to pay a posted price or if a price already agreed on needs to be adapted due to a new, unexpected situation.

4.3 Application of pricing models in IT and other industries

Cloud services have been around for quite some time already; Amazon´s Elastic Cloud and Salesforce´ CRM software are just two prominent examples. This chapter will give an overview of pricing models already used in this field. In addition, pricing models that are used for other network based products and further industries are described to lay the foundation for the following outlook on future pricing models in the field of cloud computing.

4.3.1 Pricing in Cloud Computing

Pricing models in cloud computing generally fulfill three conditions: 1) the QoS is clearly defined, therefore quality for a specific service is not changing over time and does only influence cost calculation of the vendor, not the pricing model itself; 2) the price is based on the actual usage of the service; 3) price discrimination can only be achieved with the help of different QoS levels, via pricing depending on time or amount of usage and indirectly via market entry barriers.

The most common pricing schemes in cloud computing today are by far posted prices as shown in Table II. While Platform- and Infrastructure-as-a-Service offerings are generally charged by the hour, all software offered as "cloud software" is billed via a monthly or even annual flat fee. Although that is a lot more flexible than buying software licenses for unlimited usage, true usage-based pricing should be based on shorter periods, e. charging one-time usage to look at or print a file.

The only price discrimination besides different levels of service quality could be found based on region (e.g. cheapest Salesforce.com license at identical QoS in Japan almost 80% more expensive than in the US), volume (e.g. discounts at Amazon´s Elastic Computing Cloud (E2C) for renting an instance for one or three years and getting lower hourly rates in return), or time of usage (e.g. Windows Azure charges inbound data transfer only in peak times from April 2011 on).

The currently most innovative pricing model is used for the spot instances of Amazon´s E2C. Customers can name a maximum hourly fee they are willing to pay, and depending on the current workload of the infrastructure, the hourly fee is increased or decreased by Amazon every 30 minutes to reach an optimum of utilization. Therefore, consumers that don´t want to buy reserved instances for a year (and are flexible with regards to time) can lower their costs compared to on-demand instances that are more than 100% more expensive. Although the price is publicly posted, scientifically speaking this model is a recurring auction on many identical goods with a hidden reserve price (which is constantly changing), hidden bidding and all successful bidders paying the amount of their bid.

This pricing mechanism for Amazon´s E2C is strikingly similar to the pricing method that [9] proposed almost 15 years ago for network based products with guaranteed QoS in a monopoly. Based on an estimation of the probable user demand, they suggest to define the monopoly price (E2C reserved instances) and set up a spot market for remaining capacities (E2C spot instances) while still offering the option to order guaranteed capacities on demand (E2C on-demand instances). This does not only give an impression of Amazon´s current self-conception as a monopolist, it also gives a good foresight of how certain cloud offerings may be priced in the future.

One fact however attracts attention: all offerings only address spot markets with their pricing schemes, i.e. the contract between provider and consumer is put in force at the time the usage is starting and the price at that specific moment. They don´t allow consumers to reserve usage volumes for a later point in time.

4.3.2 Pricing in Other Industries

The four main models to price products – posted prices, auctions, tendering and bargaining – have been adopted and modified in manifold ways in mature, non-IT industries. Compared to network based products, customers in other industries are not facing the problem of congestion. They are not competing for an infinitely divisible, abstract good, i.e. bandwidth, but for real and often non-divisible products carrying different valuations of each consumer. Therefore, consumer goods can be offered at guaranteed QoS and allow for posted prices (e.g. groceries), bargaining (e.g. to lower a posted price) and auctions. The costly pricing mechanism of tendering is used rather for capital-intensive goods as large machines, infrastructure or buildings.

The pricing mechanisms of bargaining and tendering are straightforward and have experienced limited innovations over time. Bargaining is still wide-spread in developing countries and itself often a much valued part of a transaction. In most other cases, it is a means of adapting posted prices for commodity products in case the customer has a lower valuation. However, the rapid growth of the internet has made bargaining also very popular online by using reverse pricing (e.g. on priceline.com). It describes process where a customer releases a bid for a certain product on an internet platform, giving different competing vendors the option to accept or refuse. While the customer has the chance to get a good deal, vendors can optimize the utilization of their resources without risking that the (hidden) reserve price is jeopardizing the posted prices in other sales channels.

Tendering is also a pricing mechanism mostly found offline. Usually applied for very capital-intensive goods, it is a time-consuming way of specifying the final product by detailing the initial tender. Thus, it allows the tenderer to offer a price based on realistic estimations of costs to be expected and gives the client a chance to receive comparable offers. The reversed auction mechanism explained above (e.g. used by my-hammer.com) is an online tendering mechanism. It has become popular for simple customized products due to the immense acceleration of communication speed.

Posted prices are very common for consumer goods. While the internet has become a very important channel to interact with customers and sell products, it has also given vendors the opportunity to significantly refine their pricing mechanisms. Collecting customer data has become so cheap that vendors can easily compare and correlate customer profiles to estimate their willingness to pay for certain products. This allows for very detailed price discrimination and almost personal prices (e.g. based on age, job or even address) which are posted only to the respective individual.

But customers also have to chance to profit from the internet by forming virtual buying syndicates (e.g. on letsbuyit.com). It enables many individuals to aggregate their demands on an internet platform and therefore use volume discounts.

Especially tour operators have been very creative using the internet to optimize pricing mechanisms. They try to maximize profitability by bundling products or discriminating prices via time and quality. Bundling reduces comparability to competing offers and helps to maximize utilization by offering not-so-popular products with popular ones. They use dynamic pricing to increase or decrease posted online prices depending on remaining capacity and oversubscription of available capacities knowing that a certain percentage of guests will cancel or not show up. Even offering goods at a posted price without exactly defining the product is increasingly used, e.g. by airlines selling tickets to “a destination at the beach” or “a European capital”, telling the customer only when and where the flight begins.

Auctions are undoubtedly the area with most innovations in recent times, even leading to the formation of auction theory as independent field of research. Traditional auction mechanisms are English auctions used by auction houses for works of art, Dutch auctions to sell many identical products (e.g. flowers) and double auctions applied on stock markets.

The internet has also made auctions a widely used instrument of pricing for consumer goods. They are often auctioneered with Second-Price Sealed-Bid auctions which have proven to be efficient and giving bidders incentives to bid at their true valuation for the auctioned good [10]. Ebay.com for example is joining characteristics of the English (multiple bids possible per bidder) and the Vickrey auction (second highest bid is final price plus a small increment). In a business context, reversed auctions have become a viable instrument of procurement with the rise of eCommerce, e.g. for basic chemicals and other commodities.

All pricing mechanisms are not only applied on spot markets, but also for future demand. Options to buy (call option) or sell (put option) a certain good at a specific price are even traded on stock markets, thus combining posted prices with auctions.

5 Outlook

5.1 Required standards for cloud services

Cloud services will only be as successful as many people believe today if overarching standards are defined. Communication standards for the interaction of providers, users and among services have a solid basis with the TCP/IP, the web service stack defining XML-based standards as SOAP, WSDL or UDDI and further protocols as REST. Standards describing quality and pricing information on the other hand are either not existent or lacking overarching application.

To define service quality, one of the existing WS languages describing technical elements of quality needs to be agreed on and enhanced to fulfill all requirements. Also, a single framework incorporating all necessary parameters that influence service quality needs to be defined. In addition to response time or mere computing power, this may also include soft facts as usability. It will also be of great importance to define these parameters with customer-oriented view. A user is not interested in the clock rate of certain CPU, all that matters are the FLOPs that a certain service can offer in a specific context.

A number of standards to describe quality concepts like ITU-T E-800, XML-based languages to define quality parameters like WSLA and ontologies to allow QoS definitions like DAML-QoS have been proposed by different vendors and organizations. However, they can only be the basis for a comprehensive catalogue of quality parameters. For cloud services to be comparable, overarching service classes and optional functionalities need to be defined by an overarching initiative. This will result in a service construction kit similar to car configurators with generally accepted standards.

The most promising initiative so far is the “Standards Acceleration to Jumpstart the Adoption of Cloud Computing” project by the NIST. Started in 2010 and designated by the US Federal CIO Vivek Kundra, it´s goal is “to drive the formation of high-quality cloud computing standards” [11] in order to. Although this venture is focusing on portability, security and interoperability at first, standards to describe service quality and actual suggestions for overarching SLAs are likely to be developed soon. For the time being, vendors will try to push their technical configurations and sets of standards into the market, leading to lock-in effects for almost any cloud service.

Concerning prices, the standardization has not even started yet. So far, no protocols or any kind of standard within existing protocols exist to describe which pricing mechanism a service is supporting, if upper and lower price boundaries exist and which types of contracts are supported. This is still an area that requires a lot of research. The next chapter describes on more detail which pricing mechanisms are likely to be used in a world of cloud commodities.

5.2 Future pricing of cloud services

As for any product, the future cloud pricing mechanisms depend strongly on the ratio of vendors and buyers for a specific cloud service and the availability and costs of alternative solutions. If a vendor has a monopoly for his service, he can dictate the resource allocation and choose between posted prices (e.g. as Amazon Web Services today), auctions or bargaining with every individual customer.

However, monopolies will be rare or not persist for long. Especially for IT infrastructure, overarching technical and performance standards are already evolving, making services comparable and eventually leading to converging markets with interchangeable products. This even holds true for the SaaS market which is a lot more diversified compared to Infrastructure- (IaaS) and Platform-as-a-Service (PaaS) due to the manifold application areas and specific requests by end users. While computing services could be made comparable via a virtual measurement unit (e.g. input/output performance or million instructions per second), software is much more dependent on functionality and usability.

Thus, the SaaS market is in fact composed of many small SaaS markets with rather few competitors who are able to shoulder the large initial investments to develop software. Nonetheless, SaaS delivered as true cloud services via networks is as dependent on and limited by the underlying IT infrastructure as any cloud service and calls for differentiated service levels and pricing models. Due to a growing standardization of process landscapes and thus the supporting software, e.g. based on ITIL or eTOM, the software market is also experiencing decreasing fragmentation and is more and more comparable to IaaS and PaaS markets.

Therefore, the level of standardization of a certain service category will be an important driver of the corresponding pricing mechanism in all three categories of cloud services. While ERP systems or stock trading platforms will be subject to customization and an opportunity to gain competitive advantage for some time to come, other services as hosting or communication will be highly standardized. At the same time, customized products will less likely become true cloud services while services with strong positive network effects will be typical cloud commodities.

From a vendors´ perspective, Paris Metro Pricing based on different service levels will be the method of choice to maximize utilization of hardware and revenues while still offering transparent pricing. Customers will define which available services fulfill their required quality levels and leverage pricing and quality to choose the adequate service.

For services with high degree of standardization, usage-sensitive fees based on very short time intervals (e.g. one-time access or per hour) or small volumes (e.g. storage per day per MB) are thinkable since the costs for searching and putting a service into operation are marginal. Vendors can flexibly adjust posted prices according to the current utilization of their resources or auction remaining capacities via double auctions similar to stock markets.

Consequently, customers relying on spot prices of services will try to manage the risk of price volatilities by requesting predicted service volumes for future points in time. As other non-storable goods, cloud services might have much higher volumes in future contracts than in spot contracts someday [12]. Similar to electricity markets, this may even lead to a “cloud stock exchange” with derivatives like options. Also, companies operating their own IT infrastructure might offer unused capacities in times of low utilization via this platform just like private households are offering electricity. In forward markets, pricing via bargaining and tendering could also be applied as long as the expected contract volumes justify higher costs for the customization of products and an extensive pricing process.

Providers working with posted prices will continue to discriminate prices not only based on quality levels or utilization (i.e. time), but also use volume discounts, blur pricing transparency by bundling unpopular products with popular ones or sell capacities to virtual 3rd party cloud providers. Also, for less standardized services that don´t generate enough supply and demand to justify double auctions, business-to-business internet platforms with Vickrey or reverse auctions might help vendors to further optimize utilization.

6 Summary

The commoditization of cloud services today is still in its beginning, applying only few methods of defining comparable service levels or assigning prices to resources based on their valuation. The markets are fragmented and standards to compare performance of similar services are rare, leading to monopolistic structures and therefore pricing mechanisms dominated by the vendors.

However, it seems clear that the market will mature in the next years as companies are continuously outsourcing commoditized IT services which are not supporting their core competencies. As large power plants transformed the electricity industry a century ago, the world of IT will be transformed by huge data centers currently built by providers like Amazon or Google. The underlying economies of scale lead to cost reductions [13], and more and more IT departments will decide to use cloud services once new investments into infrastructure are required.

Vendors as well as governments have understood that a comprehensive set of standards will be the basis for a continuing success of cloud computing. Most efforts currently focus on interoperability, portability and security. This will be followed by standards to describe service quality, allowing cloud users to compare different offerings more easily.

The pricing mechanism used for each type of cloud service depends on the market structure and possible alternatives. Markets of capital-intensive services as software or services addressing a limited clientele will be rather vendor-driven which prefer long-term contracts. Products that can be highly standardized with regards to service quality will become more and more comparable, bringing forward typical pricing mechanisms of commodities as stock and forward markets.

End users however prefer simple pricing schemes and don´t want price meters constantly showing their usage-based costs increasing [14]. Thus, IT departments will have to manage the identification and purchase of various cloud services while translating the diverse, complex and usage-based pricing mechanisms into transparent and user-friendly internal pricing.

Depending on regulatory efforts of developed countries, the “cloud industry” will experience a certain degree of liberalization to avoid large IT companies becoming too powerful. This may lead to a rise of virtual cloud service providers operating platforms to sell and buy remaining capacities with different posted price and auction models, comparable to virtual mobile network operators (MVNOs) in the telecom industry.

Automation of pricing and allocation of resources will go alongside with this development, requiring providers and consumers of IT services to agree on overarching standards to define service levels, underlying service catalogues and price-related information, e.g. with XML. The current research of the authors of this paper is focusing on these standards, platforms, requirements of customers and providers as well as procedural and organizational implications. Results are to be expected within the next year.

7 References

[1] L. Dignan, “ Cloud computing market: $241 billion in 2020”, URL: http://www.zdnet.com/blog/btl/cloud-computing-market-241-billion-in-2020/47702, ZDNet, download 05/2011).

[2] P. Mell, T. Grance, “The NIST Definition of Cloud Computing v15”, URL: http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc, National Institute for Standards and Technology 2009 (download December 2010).

[3] A. Keller, H. Ludwig, "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services", IBM Research, May 2002.

[4] ITU-T E.800, "Terms and definitions related to quality of service and network performance including dependability", September 2008.

[5] G. Dobson, A. Sánchez-Macián, "Towards unified QoS/SLA ontologies", Proc. of 3rd International Workshop on Semantic and Dynamic Web Processes, 2006.

[6] S. Voigt, "Wertbasierte Pricingstrategien", URL: http://www.wiwi.tu-bs.de/marketing/service/download/art/SKP_Voigt_Wertbasiertes%20Pricing.pdf, Technical University Braunschweig, 2008, p. 34 (download January 2011).

[7] J.K. MacKie-Mason, H.R.Varian, "Some Economics of the Internet", 10th Michigan Public Utility Conference at Western Michigan University, 1993, p. 20.

[8] N. Semret, R. R.-F. Liao, A. T. Campbell, A. A. Lazar, "Market Pricing of Differentiated Internet Services", Proc. of the 7th InternationalWorkshop on Quality of Service (IEEE/IFIP IWQOS'99), June 1999, pp. 1-10.

[9] Q. Wang, J. M. Peha, M. A. Sirbu, "Optimal pricing for integrated-services networks with guaranteed quality of service", 1996. In: L. W. McKnight, J. P. Bailey, "Internet economics", MIT Press 1997, pp. 353-378.

[10] W. Vickrey, "Counterspeculation, auctions, and competitive sealed traders", Journal of Finance, vol. 16, March 1961, pp. 8-37.

[11] NIST Cloud Computing collaboration site, URL: http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/About (download May 2011)

[12] G. Cheliotis, C. Kenyon, R. Buyya, „Grid economics: 10 lessons from finance“, Joint Technical Report, GRIDS-TR-2003-3, IBM Research Zurich, April 2003, p. 4.

[13] P. Harms, T. Yamartino, "The economics of cloud", URL: http://www.microsoft.com/presspass/presskits/cloud/docs/The-Economics-of-the-Cloud.pdf, Microsoft 2010 (download February 2011).

[14] J. Altmann, K. Chu, "A Proposal for a flexible service plan that is attractive to users and internet service providers", 1999, p. 6.

Developing Service-Oriented Applications: a method engineering based approach

Alfredo Garro, Wilma Russo, and Andrea Tundis
Department of Electronics, Computer and System Sciences (DEIS), University of Calabria, via P. Bucci 41C,
87036 Rende (CS), Italy
{alfredo.garro, wrusso, andrea.tundis}@unical.it

Abstract - *The Service-Oriented paradigm, which conceives software resources as discoverable services available on a network, is proving an effective approach for providing business solutions in distributed and heterogeneous computing environments. However, due to the different and numerous issues to face, it is witnessing a growing interest in the use of methodologies suitable for supporting the development of service-oriented applications. The paper proposes an approach, centered on the Method Engineering paradigm, which enables the definition of new methodologies tailored to address specific issues arising in developing of service-oriented applications through the exploitation of fragments of methodologies existing and experimented. In particular, it is shown how to obtain, through composition of method fragments, a complete process which covers from requirements specification to testing of service-oriented applications. The complete definition of a method fragment (MF-Web Services Builder) and a related CASE tool are also presented along with a case study showing their exploitation for building a real service.*

Keywords: Service-Oriented Applications; Web Services; Method Engineering; Service Component Architecture; Component-Based Development

1 Introduction

The service-oriented approach promotes the reuse of existing assets in the development of new services and represents an effective interoperability solution in distributed and heterogeneous environments [6, 20]. In particular, the availability of wide adopted Web Services standards (WSDL, SOAP, UDDI, etc.) [20] for the description, communication, and discovery of services enables the jointly exploitation of services independently from the specific implementation technologies. Despite these advantages, the development of a service-oriented application faces several challenges concerning: the requirement specification, the application definition, the discovery, deployment, composition, and integration of services, and, finally, the application testing. To address these topics, several software engineering methodologies and related tools have been proposed in the service-oriented domain [18], some of these cover the whole application lifecycle (from requirements to testing) [16, 2], whereas others address specific aspects [9, 11, 3, 13]. However, existing methodologies often cannot be used “as they are” because of the specific characteristics of the application to develop. In these cases, significant efforts are required which are focused or on customization of existing methodologies or on definition of new ones without any fruitful reuse of those existing [18].

A solution, which makes it possible defining methodologies that fit specific necessities without losing the advantages coming from the exploitation of existing and experimented ones, can be represented by the adoption of the Method Engineering paradigm [4, 5] which has already proved its effectiveness in both the object-oriented and agent-oriented software engineering communities [12]. According to this paradigm, a methodology is obtained by assembling pieces of methodologies (method fragments), *ex-novo* defined or obtained from those existing and available in a repository (Method Base) [7].

In this context, this paper aims to bring the benefits arising from the Method Engineering paradigm in the service-oriented domain. In particular, it is shown how a complete development process, which covers from requirements specification to application testing, can be obtained by composing method fragments addressing some specific and recurrent aspects in the service-oriented domain. The complete definition of a method fragment (MF-Web Services Builder) which addresses a specific aspect concerning the development of Web Services is also provided according to the IEEE FIPA Specifications [8]. Moreover, a CASE tool related to MF-Web Services Builder is presented along with a case study showing its exploitation for building a real service.

The paper is organized as follows: Section 2 introduces fundamentals of the Method Engineering paradigm and exemplifies its exploitation in the development of service-oriented applications; a method fragment (MF-Web Services Builder) is completely defined in Section 3, and the related CASE tool along with a case study showing its exploitation is shown in Section 4; finally, conclusions are drawn and future works delineated.

2 Method Engineering and development of Service-Oriented applications

In this Section, an approach centered on the Method Engineering paradigm which enables, as well as in others software engineering domains, the exploitation of fragments of existing and experimented methodologies in defining new ones, is presented. In particular, Section 2.1 introduces fundamentals of the Method Engineering (ME) paradigm whose exploitation in the service-oriented domain is exemplified in Section 2.2 through a process which is obtained by composing method fragments and covers from requirements specification to testing of a service-oriented application.

2.1 The Method Engineering paradigm

ME allows for obtaining Software Engineering Processes (SEPs) by defining and combining method fragments able to support specific phases of a development process and/or to address specific issues or application aspects. Method fragments which can be either defined ex-novo or obtained by fragmentizing existing methodologies, are auto-consistent and reusable methodological chunks stored in a repository, called Method Base, from which they can be retrieved and assembled during the construction of a SEP.

According to the IEEE FIPA Specifications [8], a method fragment defines a process which receives a set of input work-products and produces a set of output work-products by possibly managing intermediate work-products. A fragment is further characterized by application guidelines that illustrate how to use the fragment and the related best practices, a glossary of the exploited terms, composition guidelines which describe the issues addressed by the fragment, and dependency relationships which give information about others possible related fragments. These meta-data constitute the fragment description which can be codified and stored in the Method Base for facilitating fragment retrieval, adaptation and composition [8].

The definition of a SEP according to the Method Engineering paradigm requires the following main steps (see Figure 1):

1. definition of the characteristics of the SEP and the activities to be carried out (SEP Lifecycle Definition);
2. selection of the methodological fragments from the Method Base (if available) on the basis of the activities defined in step 1 (Method Fragments Selection);
3. definition of new fragments to cover process activities that are not supported by fragments available in the Method Base (Method Fragments Definition);
4. adaptation and composition of the (selected and/or ex-novo created) method fragments to obtain the SEP (Method Fragments Adaptation and Composition).

With reference to step 4, the fragments can be easily integrated through a work-product driven approach (input work-products of a fragment should be derived from output work-products of other fragments, possibly adapted) [7, 10, 17].

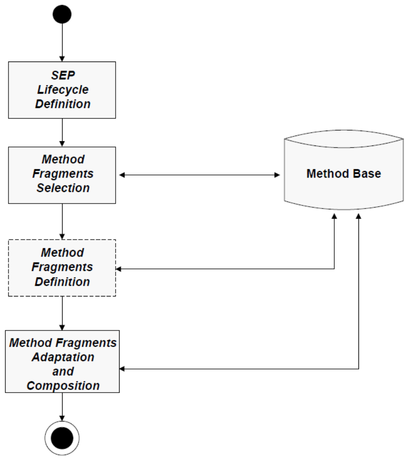

SEP Lifecycle Definition
Method Fragments Selection
Method Base
Method Fragments Definition
Method Fragments Adaptation and Composition

Figure 1. Definition of a SEP according to the ME paradigm

2.2 Exploiting Method Engineering in the service-oriented domain

According to the ME paradigm the development of a specific service-oriented application can be addressed by the composition of method fragments either retrieved from a Method Base and possibly adapted or defined ex-novo. In particular, the following method fragments can be identified:

1. Requirements Specification, which formalizes the application requirements as output work-product.
2. Application Definition, which, starting from application requirements, makes available a choreography of an application.
3. Service Discovery, which discovers, on a given research domain, the set of services able to cover the roles of a given choreography of an application.
4. Service Development, which builds and makes available a service adhering to a specific service contract.
5. Service Composition, which specifies the interactions of a set of services selected to cover the roles of an application choreography.
6. Service Integration, which implements the communication infrastructure among the services constituting an application.
7. Application Testing, which aims at validating and evaluating an application.

An example of composition of method fragments for obtaining a complete process for the development of service-oriented applications, which covers from requirements specification to application testing, is showed in Figure 2.

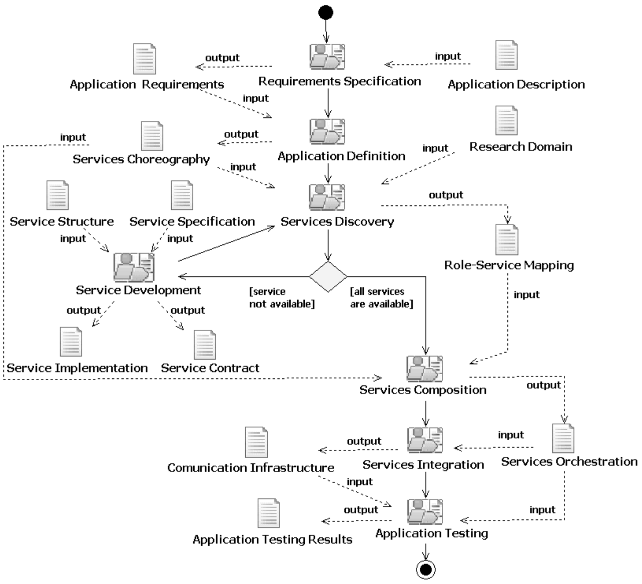


Figure 2. An example of method fragments composition

The process adheres to a typical iterative-incremental lifecycle and the involved fragments are integrated through a work-product driven approach (input and output work-products are reported in Table 1)

TABLE I. Method fragments and related work-products

Method Fragment	Input Work-Products	Output Work-Products
Requirements Specification	- Application Description	- Application Requirements
Application Definition	- Application Requirements	- Services Choreography
Services Discovery	- Services Choreography - Research Domain	- Role-Service Mapping
Service Development	- Service Structure - Service Specification	- Service Contract - Service Implementation
Services Composition	- Services Choreography - Role-Service Mapping	- Services Orchestration
Services Integration	- Services Orchestration	- Communication Infrastructure
Application Testing	- Services Orchestration - Communication Infrastructure	- Application Testing Results

It is worth noting that: (i) starting from a process definition, the availability of method fragments for the service-oriented domain would allow covering the different process phases and/or addressing the different development issues in different and specific ways depending on the chosen fragments; as an example, the availability of different fragments for Service Discovery would allow to address this aspect with different approaches and techniques [11]; (ii) specific SEPs can be defined by composing the method fragments selected on the basis of the desired software development lifecycle; as an example, if the services that constitute the application are all available and their interactions have been already defined, a light process which covers only Service Integration and Testing could be defined and related fragments selected and composed.

3 Component-Based Development of Web Services

In the development of a service-oriented application, a central issue as that concerning the implementation of new services can be addressed through a specific method fragment (see Section 2). In the following, the complete definition of a method fragment for (Web)Services Development (MF-Web Services Builder) is provided. This fragment, once available in a Method Base, could be exploited, as is or after suitable adaptation, in the composition of specific processes that require development of services, processes that can be both complete as that exemplified in Section 2.2 or cover only some phases of the application lifecycle. According to the IEEE FIPA Specifications [8], the description of the process defined by the fragment as well as that of other meta-data concerning the features and the use of the fragment are described in Sections 3.1 and 3.2 respectively.

3.1 Process definition

MF-Web Services Builder conceives and structures a Web Service as a set of interconnected and jointly working components built using the same or different technologies and executed on the same machine or across a network [14].

The concrete model exploited by MF-Web Services Builder for defining Web Services following this component-based approach adheres to SCA Specifications [14], and, in particular to the SCA Assembly Model [14] which models a service-oriented application as a SCA domain consisting of a set of Services called composite which are in turn structured in components. Each component offers a set of business functions (or services) to other components and can have settable properties which influence the execution of business functions; moreover, dependencies of each component on services provided by other components are called references. The configuration of a component requires both to set values for its properties and to wire its references to services provided by other components. In a composite a component is responsible to make available and exploitable the services provided by the composite (the focus component). A composite has settable properties which are related to those of its components and references to other composites. A service contract, which can be codified by a WSDL file, gathers the services provided by a composite.

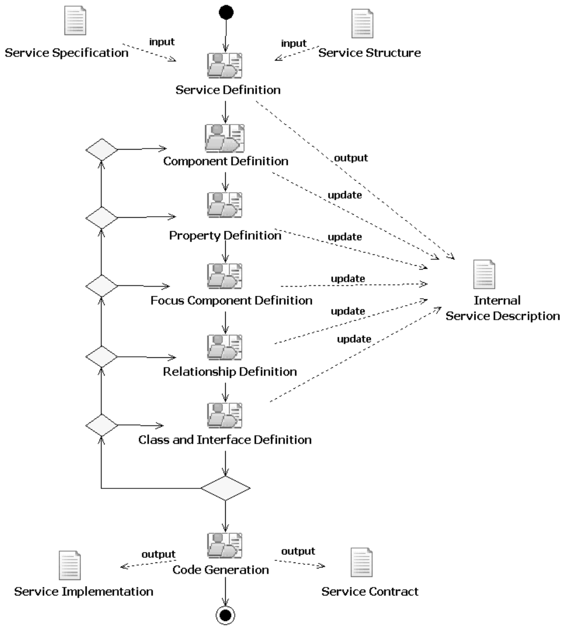


Figure 3. The process provided by MF-Web Services Builder

The adoption of the above described SCA Assembly Model allows defining a complete process for the development of a Web Service conceived as a SCA composite. In Figure 3 the process, which constitutes the core element of the MF-Web Services Builder method fragment, is showed. In particular, the process starts with the Service Definition activity which deals with the analysis of the input work-products (Service Specification and Service Structure) described in Section 3.2. The result of this activity is a clear definition of the Service requirements which is also captured in a refined version of the input work-products. Then, a component-based structure of the composite implementing the Service is obtained and the capabilities and role of each component are specified (Component Definition activity). For each component its properties which capture the component state are introduced in the Property Definition activity. One of the components is selected as an interface with the environment (Focus Component Definition activity). After selecting the Focus Component, the URI and the contract of the Service are defined. At this point of the process, the references (and wires) among the Service components (Component Relationship Definition activity) and the interfaces and classes which constitute each component are defined. The concrete implementation of Service components if not available should be also provided. During these activities, an Internal Service Description which contains a description of the components, their relationships and properties, is constantly updated (see the support work-products in Section 3.2).

The described process can be iterated until the service requirements are met. Finally, the Service Contract is generated along with the Service Implementation (Code Generation activity) which can be refined by adding functions and methods on the basis of both the Service business logic and the chosen implementation (see the output work-products in Section 3.2).

3.2 Meta-data

The IEEE FIPA meta-data related to MF-Web Services Builder are the following:

- Fragment Prerequisites: the specifications of the Service to be developed must be available and formalized in the input work-products (Service Specification and Service Structure).
- Input work-products: a text document (Service Specification) with an informal description of the service to be developed and an UML class diagram (Service Structure) representing a high-level view of the service structure should be provided.
- Output work-products: the fragment produces an implementation of the Service (Service Implementation) and a WSDL file (Service Contract) that describes the offered services and how to structure messages and data for their exploitation.
- Support work-products: a SCDL (Service Component Description Language) file (Internal Service Description), which describes the Service components, their relationships and properties, is constantly updates during the development process.
- Application Guidelines: a set of best practices, examples and case studies which show when and how to effectively exploit MF-Web Services Builder, are provided with the fragment documentation.
- Glossary of terms: to avoid misunderstanding and an uncorrected use of the fragment, a glossary which reports the definition of the main terms used for the fragment definition is delivered. In particular, a definition of the terms related to the SCA Assembly Model [14] is provided.
- Composition Guidelines: useful information for guiding the work-product based composition of MF-Web Services Builder with other fragments is provided (see Section 2).
- Dependency Relationships: the input work-products for MF-Web Services Builder should be provided by those method fragments addressing the identification of Web Services (e.g. the Application Definition method described in Section 2.2); the output work-products of MF-Web Services Builder can represent input for others method fragments as the Services Composition fragment (see Section 2.2).

4 MF-Web Services Builder CASE Tool

To allow a concrete exploitation of the methodological approach to Web Services development provided by MF-Web

Services Builder, a CASE tool (MF-Web Services Builder CASE Tool) has been implemented; its architecture and provided functionalities are presented in Section 4.1, whereas a case study showing its effectiveness for building a real service is reported in Section 4.2. The fragment documentation along with the source code of the CASE Tool is available on the OpenKnowTech Project web site [15].

4.1 An overview of the CASE tool

MF-Web Services Builder CASE Tool, which supports the execution of the process defined by MF-Web Services Builder (see Figure 3), has been designed and implemented as a plugin for the Eclipse platform and according to the SCA specifications by exploiting, in particular, the SCA Composite Designer which is part of the Eclipse STP/SCA project [19] and allows a Model-Driven and component-based service development. The CASE Tool is able to:

- support the execution of the process activities through a wizard-based visual interface;
- create and constantly update the SCDL XML file containing the description of the service components, their relationships and properties;
- automatically generate the Service Implementation (currently in Java) and the Service Contract as a WSDL file;
- deploy the developed service in a SCA runtime environment for making it available at a specified URI.

An overview of the MF-Web Services Builder CASE Tool architecture is reported in Figure 4.

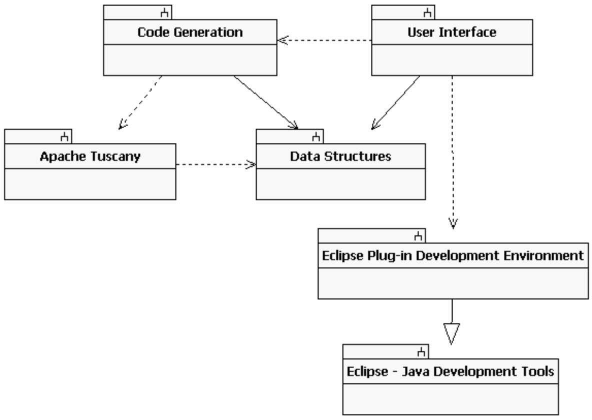
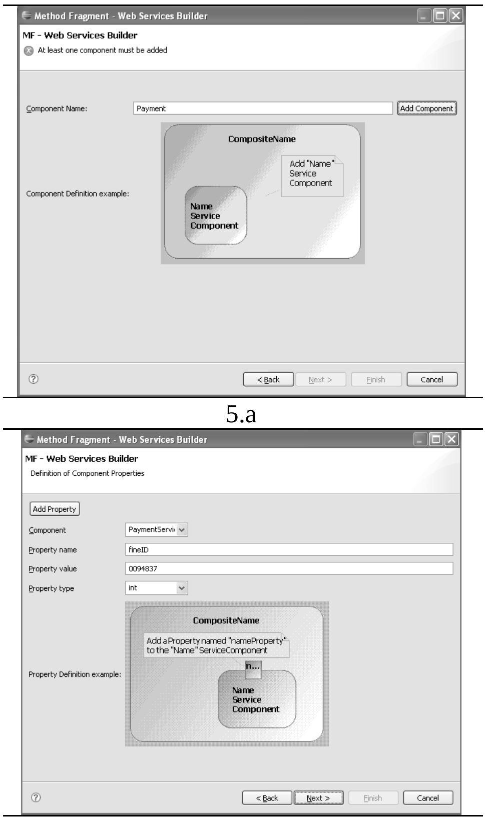

Code Generation
User Interface
Apache Tuscany
Data Structures
Eclipse Plug-in Development Environment
Eclipse - Java Development Tools

Figure 4. The architecture of the MF-Web Services Builder CASE Tool

In particular: (i) internal data structures (Data Structures) map the basic concepts of the SCA Assembly Model; (ii) the visual user interface has been obtained by defining a set of wizards (User Interface) through the Eclipse PDE (Plug-in Development Environment); (iii) the generation of the WSDL service contract and the service deployment are obtained by using the Tuscany Apache framework [1] (Code Generation) and the Tuscany SCA runtime environment (Apache Tuscany) respectively, choosing Tuscany among the implementations of the SCA Specifications [14] as it is complete, well-documented and widely adopted.


Method Fragment - Web Services Builder

MF - Web Services Builder

At least one component must be added

Component Name: Payment

Component Definition example:

5.a

Method Fragment - Web Services Builder

MF - Web Services Builder

Definition of Component Properties

Component: PaymentServi

Property name: fineID

Property value: 0094837

Property type: int

Property Definition example:

5.b

Figure 5. Component (5.a) and Property Definition (5.b) wizards

4.2 Exploiting the CASE Tool

The feasibility of MF-Web Services Builder and its supporting CASE tool in the development of Web Services, and, in particular, the significant reduction of programming and implementation efforts are demonstrated through a simple but real case study concerning the development of the FinePayment Service, a Web Service for the online payment of fines by the Local Police.

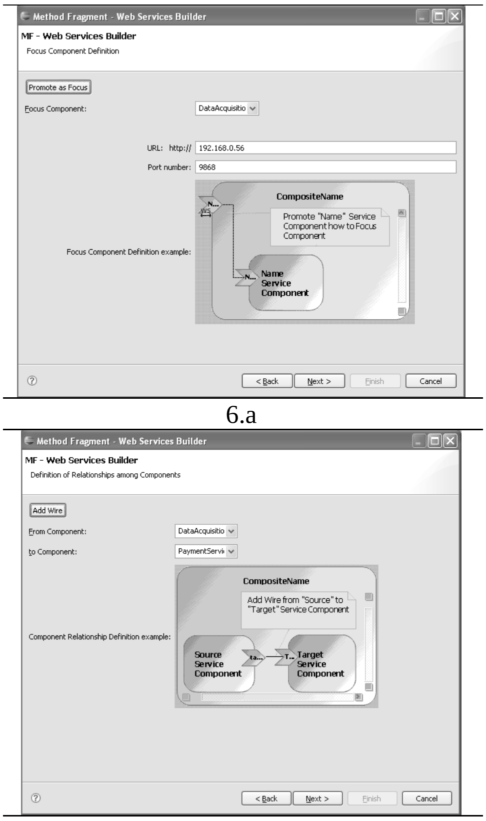
According to the process provided by MF-Web Services Builder (see Figure 3), on the basis on the analysis and definition of the service requirements (Service Definition activity) the Component Definition activity has identified the following three components:

- Payment, which manages data to perform fine payment;
- Data Access, which allows retrieving fine data and storing payments;
- Data Acquisition, which temporarily stores the fine data and exploits the functionalities provided by the other service components.

In the Property Definition activity the following component properties were defined: (i) amount of fine, interest on late payment, and total amount (Payment properties); (ii) data base connection settings (Data Access properties); (iii) fine ID, vehicle plate number, driver name, and bank account for the payment (Data Acquisition).

In Figure 5 the definition of the Payment component and its properties through the wizards provided by MF-Web Services Builder CASE Tool are reported.

In the Focus Component Definition activity, the Data Acquisition component was promoted as Focus and the URL and the port number to identify the Service on the network was specified (see Figure 6.a). In the Component Relationship Definition activity the references among components were introduced (see Figure 6.b).


Method Fragment - Web Services Builder
MF - Web Services Builder
Focus Component Definition
Promote as Focus
Focus Component Definition example:
CompositeName
N...
WS
Promote "Name" Service Component how to Focus Component
N...
Name Service Component
< Back
Next >
Finish
Cancel

Focus Component: DataAcquisitio

URL: http:// 192.168.0.56

Port number: 9868

6.a

Method Fragment - Web Services Builder
MF - Web Services Builder
Definition of Relationships among Components
Add Wire
Component Relationship Definition example:
CompositeName
Add Wire from "Source" to "Target" Service Component
Source Service Component
ta...
T...
Target Service Component
< Back
Next >
Finish
Cancel

From Component: DataAcquisitio

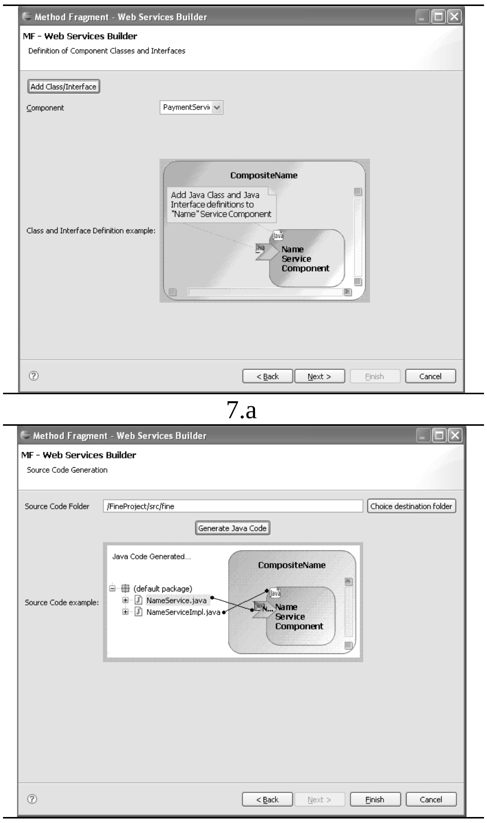
to Component: PaymentServi

6.b

Figure 6. Focus Component Definition (6.a) and Relationship Defintion (6.b) wizards

In the Class and Interface Definition activity classes and interfaces (available or defined *ex-novo*) were associated to each component for implementing its functionalities, and, finally, the Service Implementation and its Contract were automatically generated (see Figures 7.a and 7.b).

Figure 8 shows the high level architecture of the real system for the management of fines in which the *FinePayment* service has been deployed. This system is currently distributed by a European software vendor and used by some public administrations.


Method Fragment - Web Services Builder
MF - Web Services Builder
Definition of Component Classes and Interfaces
Add Class/Interface
Class and Interface Definition example:
CompositeName
Add Java Class and Java Interface definitions to "Name" Service Component
Java
Java
Name Service Component
< Back
Next >
Finish
Cancel

Component PaymentServi

7.a

Method Fragment - Web Services Builder
MF - Web Services Builder
Source Code Generation
Choice destination folder
Generate Java Code
Source Code example:
Java Code Generated...
(default package)
NameService.java
NameServiceImpl.java
CompositeName
Java
Java
N...
Name Service Component
< Back
Next >
Finish
Cancel

Source Code Folder /FineProject/src/fine

7.b

Figure 7. Class and Interface Definition (7.a) and Code Generation (7.b) wizards

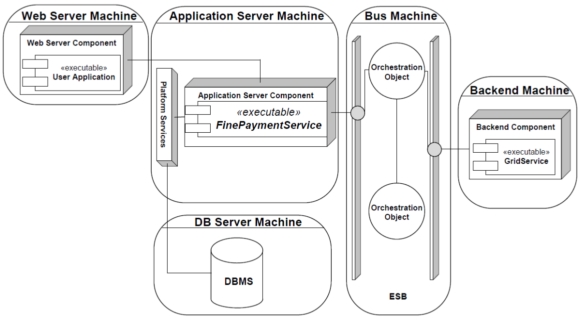


Figure 8. Execution environment of the FinePayment service

5 Conclusions and future work

The variety of contexts in which to develop service-oriented applications actually makes it rather difficult to use a unique methodology and flexible enough to effectively support the development of any application regardless of the specific context in which the application relates. Therefore, due to the heavy and time-consuming efforts required to adapt an existing methodology, when it is necessary to address specific issues arising in developing a specific service-oriented application, often make it more profitable to define a new methodology without any fruitful reuse of the existing ones. This paper has proposed the exploitation of the Method Engineering (ME) paradigm in the service-oriented domain which allows defining methodologies which fit specific needs through the composition of method fragments, extracted from existing methodologies or defined ex-novo. In particular, a set of method fragments able to address recurrent challenges in the development of service-oriented applications has been individuated and composed through a work-product driven composition so obtaining a complete development process. An example of the complete definition of a method fragment (MF-Web Services Builder) concerning the central issue of service development along with the implementation and experimentation of a related CASE tool have been provided. MF-Web Services Builder adopts as the reference logical and architectural service model that defined by the SCA Specifications; moreover, its definition conforms to the IEEE FIPA Specifications so enabling fragment storage in any FIPA compliant Method Base and composition with FIPA compliant method fragments in a specific service-oriented development process.

The ongoing experimentation aims at evaluating the benefits that could arise from the exploitation of the ME paradigm in the service-oriented domain and, in particular, from the availability of method fragments and related CASE tools to exploit or possibly adapt in the definition of development processes of service-oriented applications. Currently, main efforts are geared to: (i) define method fragments covering central issues in the development of service-oriented applications; currently a fragment for supporting the composition of Web Services is under definition; (ii) obtain new and more complete development processes through the composition of the defined method fragments; currently the integration of MF-Web Services Builder with a method fragment for testing the quality of Web Services is under consideration; (iii) define design patterns able to easily drive the execution of the component-based developed process provided by MF-Web Services Builder.

6 References

[1] Apache Tuscany Framework, documentations and software, http://tuscany.apache.org

[2] Arsanjani, A., Ghosh, S., Allam, A., Abdollah, T., Ganapathy, S., Holley, K.: SOMA: A method for developing service-oriented solutions. In: IBM Systems Journal, vol. 47, n. 3, pp. 377--296, IBM Corp. Riverton, NJ, USA (2008).

[3] Baldoni, M., Baroglio, C., Martelli, A., Patti, V.: Reasoning about interaction protocols for customizing web service selection and composition. In: The Journal of Logic and Algebraic Programming, vol. 70, n. 1, pp. 53--73, Elsevier B.V., Amsterdam, The Netherlands (2007).

[4] Brinkkemper, S.: Method Engineering: engineering of information systems development methods and tools. Information and Software Technology, vol. 38, n. 4, pp. 275--280, Elsevier B. V., Amsterdam, The Netherlands (1996).

[5] Brinkkemper, S., Lyytinen, K., Welke, R.: Method engineering: principles of method construction and tool support. Springer-Verlag, Berlin Heidelberg, Germany (1996).

[6] Bruni, R., Lluch Lafuente, A., Montanari, U., Tuosto, E.: Service Oriented Architectural Design. In: Proceedings of the 3rd International Symposium on Trustworthy Global Computing (TGC’07). LNCS, vol. 4912, pp. 186--203, Springer, Heidelberg (2008).

[7] Cossentino, M., Fortino, G., Garro, A., Mascillaro, S., Russo, W.: PASSIM: a simulation-based process for the development of multi-agent systems. Int. J. of Agent-Oriented Software Engineering, vol. 2, n.2, pp. 132--170, Inderscience Enterprises Ltd., United Kingdom (2008).

[8] Cossentino, M., Gaglio, S., Garro, A., Seidita, V.: Method fragments for agent design methodologies: from standardisation to research. Int. J. of Agent-Oriented Software Engineering, vol. 1, n. 1, pp. 91--121, Inderscience Enterprises Ltd (2007).

[9] Di Penta, M., Canfora, G., Esposito, G., Mazza, V., Bruno, M.: Search-based testing of service level agreements. In: Proceedings of GECCO 2007, the Genetic and Evolutionary Computation Conference, London, England, UK (2007).

[10] Fortino, G., Garro, A., Russo, W.: An integrated approach for the development and validation of multi-agent systems. In: Int. J. of Computer Systems Science & Engineering, vol. 20, n. 4, pp. 259--271, CRL Publishing Ltd, Leicester, United Kingdom (2005).

[11] Giallonardo, E., Damiano, G., Zimeo, E.: onQoS-QL: A Query Language for QoS-Based Service Selection and Ranking. LNCS, vol. 4907, pp. 115--127, Springer, Heidelberg (2009).

[12] Henderson-Sellers, B.: Method engineering for OO systems development. Communications of the ACM, vol. 46, n. 10, pp. 73--78, ACM press (2003).

[13] Koehler, J., Srivastava, B.: Web service composition: current solutions and open problems, In: Proceeding of ICAPS 2003 Workshop on Planning for Web Services, pp. 28--35, 2003.

[14] Marino, J., Rowley, M.: Understanding SCA (Service Component Architecture). Addison-Wesley Professional (2009).

[15] OpenKnowTech Project, documentations and software available at http://www.openknowtech.it

[16] Papazoglou, M.P., Van Den Heuvel, W.J.: Service-oriented design and development methodology. In: International Journal of Web Engineering and Technology, vol. 2, n. 4, pp. 412--442, Inderscience Publishers, Geneva, Switzerland (2006).

[17] Ralyté, J., Rolland, C.: An assembly process model for method engineering. In: Proceedings of CAISE01, the 13th Conference on Advanced Information Systems Engineering, Interlaken, Switzerland (2001).

[18] Ramollari, E., Dranidis, D., Simons, A.J.H.: A survey of service-oriented development methodologies. In: Proceedings of the 2nd Young Researchers' Workshop on Service Oriented Computing, Leicester, UK, pp. 75--80 (2007).

[19] STP/SCA Tools project, components and tools, http://wiki.eclipse.org/SCA

[20] Sweeney, R.: Achieving Service-Oriented Architecture: Applying an Enterprise Architecture Approach. Wiley & Sons, Inc., New Jersey, USA (2010).

A Novel Approach for Realizing Business Agility through Temporally Planned Automatic Web Service Composition using Network Analysis

A. P. Sandhya[1], and B. Dr. M. Lakshmi[2]
[1]Research Scholar, Sathyabama University, Department of Information Technology, Jeppiaar Nagar, Rajiv Gandhi Road, Chennai - 600119, Tamil Nadu, India.
[2]Professor and Head of the Department, Sathyabama University, Department of Computer Science and Engineering, Jeppiaar Nagar, Rajiv Gandhi Road, Chennai – 600119, Tamil Nadu, India.

Abstract – *The building block of Service Oriented Architecture is web services. Web services are self contained, self describing, modular applications that can be published, located and invoked across the web. Composite services are built by aggregating elementary services by a process called web service composition. Web service composition can be classified into manual and automatic types. The first type involves design time composition when the architecture and design of the software system is planned. The second type automatic service composition involves composing web services automatically to facilitate ease of composition process for engineers. Automatic composition is generally guided by dynamic composition to handle non determinism at run time. Automatic composition can be materialized as an artificial intelligence planning problem. Planning represents a part of the world as states and changes in those states. Business agility is the ability of business to adapt efficiently and cost effectively in response to the changes in the business environment. The changes in business environment are temporal. In this paper we have devised a new algorithm called Opus deviser which targets to achieve business agility through web service composition that is temporally planned using PERT Network analysis.*

Keywords: *Automatic Web Service Composition, Temporal planning, Network analysis, Programme Evaluation and Review Technique (PERT), Opus Deviser Algorithm*

1 Introduction

Any innovation requires change. In business changes are imperative. Business changes can be incremental, business led and collaborative. Service Oriented Architecture makes changes easier. Service Oriented Architecture is an Information Technology architectural style that supports integrating business as linked services. Service Oriented Architecture increases customer access to company resources and information, and improve customer retention through faster sales. It enhances business decision making and employee productivity. Business agility means having the capability to react to incoming request and changes in a timely manner. Any business becomes agile when it is quickly run to take advantage of the best opportunities. For example when the client requests for a service which suits our service range then the agile business should react to this request and develop an adapted version that meets the client's needs. Many changes in a business environment are based on time. For example costing of a product can be based on time or delivery of products can be based on time, reordering raw materials can be based on time, etc. Business agility is therefore quasi temporally driven. A recent survey by Microsoft suggests that 87% of CIO's suggest Service Oriented Architecture as the next significant enterprise software to gain business agility [1]. The building blocks of Service Oriented Architecture are Web Services. Service Oriented Architecture composite applications are built by aggregation of elementary services. As the business environment is dynamic to time factor web service composition has to be dynamic and automated to offer business agility. Business agility can thus be rendered through temporal planning for service composition to build dynamic and automatic composite applications.

2 SOA and Business Agility

Business does not function in vacuum. Business is sensitive to external factors or influences. The external factors affect the internal functions, objectives and strategies of business. Business environment is influenced by degree of competition; social, legal, economical, political, technological and ethical factors. The external environment changes temporally. Temporal changes include customer developing new needs, new competitors enter the market, new technologies are introduced, world - wide event occurs, government introduces new policies, and so on. Changes pertain to cultural factors as well. Business should react to these temporal changes or lose business. Business reactions could include reducing prices, improving quality, spend more on promotion or cut cost. Organizations must be tight in executing today's activities and loose in adapting to future survival. Service Oriented composite applications are loosely coupled. To cope up with the changes in business environment business agility is the need of the hour. We aim to achieve business agility through Service Oriented Architecture where composite applications are built by temporal planning to adapt to changes in business environment. The composition process is done in an optimal and reliable manner using PERT Network analysis.

3 Web Service Composition

Web service composition refers to the process of collaboration of heterogeneous web services. B2B composite applications can be built from composition of services offered by multiple business partners based on business processes. Web service composition is aggregation of elementary or composite web services.

Web service composition refers to the integration of more than one web service to realize business functionality.

Web service composition can be basically classified in two ways. The first type of classification is as static and dynamic composition. The second type of classification is as manual and automatic composition. Most of the web service composition is a weaving of dynamic and automatic web service composition. Dynamicity deals more with the non determinism of the availability of a web service while automatic composition is composing of services done automatically by an agent for ease of development to the engineers. *Automatic Composition* is a composition that is done automatically by agents. Some artificial Intelligence planning algorithm is harnessed to perform automatic web service composition. A typical example OWLS XPlan2.0 where Xplan is a planner based on Hierarchical Task Network planning.

4 Literature Survey

Some of the familiar planners used for web service composition are listed below. FF [2], FF-Metric and HSP 2.0 are state space based planners. IPP, DPPLAN, LPG, PropPlan are graph plan based planners [3]. STAN4 is based on both graph plan and state space plan. UCP OP is a partial order planning based planner [4]. BLACKBOX is a planner based on SAT plan and graph plan. LGP is based on SAT plan. SHOP2 planner is based on Hierarchical Task Network planning [5]. MIPS is based on planning as model checking [6]. The popular web service composition tools are SeGSec, E-flow (HP), Aurora, STONE, ICARIS, Self-Serv, Composer, Ninja, SWORD, SHOP-2, Theseus, Argos, Proteas, Fusion, Astro, Synthy (IBM), etc [7].

RESCUERS: A framework for reliable web service composition developed in July 2009 is based on Critical Path Method [8].

The drawback of this approach is that the activity duration must known in prior which does not occur in reality. Moreover the time deadline for goals and activities in a business environment is dynamic where CPM becomes unsuitable. The probability of completion is not estimated in CPM. In this paper we put forth to use PERT for planning of web service composition that is temporally driven. Our approach generates an optimal and reliable solution enabling to achieve business agility.

5 Corporate Planning – A Time Driven Concept

Corporate planning is the process of deciding the overall objectives of the company as a whole and selecting the ways and means for achieving these objectives. Corporate planning is needed to achieve business agility.

“Corporate planning is a formal, systematic, managerial process organized by responsibility, time and information to ensure that operational planning, project planning and strategic planning is carried out regularly to enable to management to direct and control the future of the enterprise”. – Basil W. Denning [9]

Corporate planning therefore is a time driven concept.

“Corporate planning is a systematic and disciplined study designed to help identify the objective of any organization, determine an appropriate target, decide upon suitable constraints and devise a practical plan by which the objective may be achieved”. – John Argenti [9]

Corporate planning therefore needs to devise a plan with time factor. There are several planning algorithms but as corporate planning and business agility is time driven we choose temporal planning as our approach for web service composition. Web service composition is therefore thriving in a dynamic environment where we use temporal planning to generate automatic composition plan. Service Oriented Architecture is a best way to render business agility.

6 Temporal Planning

Temporal planning means planning with time based and concurrent actions. According to Dana Nau in reality actions and events occur over a time span. Preconditions do not occur only in the beginning. Actions will require maintaining partial states. Goals can be time bound [10].

7 Temporal Planning Aspects

Classical planning does not annex temporal elements. In a real world environment actions and effects are time dependent. An ideal planner must design a plan that augments the temporal features and execute actions and generate effects in proper duration. Plans can change at some period of time. Plan schemes have to be altered to external events based on time. Classical planning cannot plan concurrent actions. Temporal planning needs to generate plan that executes several actions executed concurrently. Goals of a plan are also temporally driven. A plan for instance needs to meet deadlines. A corporate environment is totally driven temporally as it has time related actions, effects and goals. Some business action need to be performed concurrently and relies on external events that are time allied.

8 Composition Planning Using PERT Network Analysis

Network analysis is used for planning projects by analyzing project activities. While planning a project, Network analysis meets the temporal planning aspects. PERT Network analysis breaks down a project into individual tasks or activities and arranges them into logical sequence. The activities can be performed concurrently or in other sequence. The time required to complete activities in certain projects is generally not known in priori. PERT Network analysis incorporates uncertainty in activity times during analysis. PERT is used to determine the probabilities of completing the various stages of the project with specific deadlines. PERT is also used to calculate the expected time to completion of a project. Web service composition is also a planning problem. In this paper we use PERT Network analysis as a novel approach to plan the temporal aspects of temporal planning to achieve business agility. We use PERT Network analysis to determine which web services should be sequentially executed during composition, schedule the execution of the composed web services and plan the completion of the entire composition within the specified deadline.

9 Overview of Our Approach

Business environment is dynamic. The dynamicity is highly driven by time. Composing web services to create a composite application achieves business agility when the dynamic and time driven factors are taken into account. Web service composition now is a planning problem. Network analysis is used for planning projects by analyzing its activities. Therefore we use PERT Network analysis which assumes that the activity times are not known in prior in a dynamic business environment to plan the composite application unlike CPM. All possible combinations of web service compositions are calculated with reference to goal deadline and verified if activity deadline is met. The matching compositions are chosen and the probability of project completion is compared and the most suitable composition is selected by selection module and executed. We choose PERT Network analysis as it considers the uncertainty of time which is a typical business environment. CPM on the other hand assumes that the activity time is known in prior which is not a typical business agile environment.

10 Opus Deviser – Algorithm

Algorithm - Opus Deviser

Input: Set of Web Services

$$W = \begin{Bmatrix} w11 & \cdots & w1n \\ \vdots & \ddots & \vdots \\ wm1 & \cdots & wmn \end{Bmatrix} \text{ where}$$

m – Number of web services in each category

n – total number of service categories that make up a composite application,

Goal deadline (g),

Activity deadline set T = $\{t_1,t_2,\ldots,t_{n-1}\}$

Categorical sequence CSeq = $\{C_1 \rightarrow C_2 \rightarrow \ldots C_n\}$

Output: Composite service S

```
Begin
01  Generate all possible combinations of web service compositions from set W such that there is one service from each category in the sequential order specified by CSeq. The number of combinations $nc = (mC_1)^n$   (1)
02  For each combination perform a PERT network analysis
      a.  Assume optimistic time (a), most likely time (l) and pessimistic time (b) or each activity based on web service execution environment
          a – Shortest possible time for each activity to be completed by the web services
          l – The time the activity will most likely take to complete the activity by the web services
          b – Longest possible time for the activity to be completed by the web services
      b.  Compute the expected duration set $TC = \{tc_1,tc_2,\ldots,tc_{nc}\}$ for all combinations of web service execution for each activity from a, l and b such that
          $tc_c = \{te_1,te_2,\ldots,te_{n-1}\}$
          $te_d = (a+4l+b) / 6$   (2)
          where c is a combination and d is an activity
      c.  Create a directed graphical representation for each combination such that each node is an event and arrow line is an activity. Each web service of a combination on execution performs an activity and consumes time. The computed expected duration set is represented in the generated graphs of every combination.
      d.  Compute the variance set VC for each activity in each combination.
          $VC = \{vc_1,vc_2,\ldots,vc_{nc}\}$
          $vc_c = \{v_1,v_2,\ldots,v_{n-1}\}$
          $v_d = ((b-a)/6)^2$   (3)
      e.  Generate the critical path set CP for every combination. The critical path is the expected project completion length. Critical path is the maximum of the sum of all the paths from start to finish in each graph.
          $CP = \{cp_1,cp_2,\ldots,cp_{nc}\}$
      f.  Compute the set of variance $TV = \{tv_1,tv_2,\ldots,tv_{nc}\}$ of critical path for each combination as the sum of variance of each activity in the critical path.
          $tv_c = \sum_{d=1}^{n-1} v_d$   (4)
      g.  Compute the set of Standard Deviation $SD = \{sd_1,sd_2,\ldots,sd_{nc}\}$ of the project length for each combination.
          $sd_c = \sqrt{tv_c}$   (5)
```

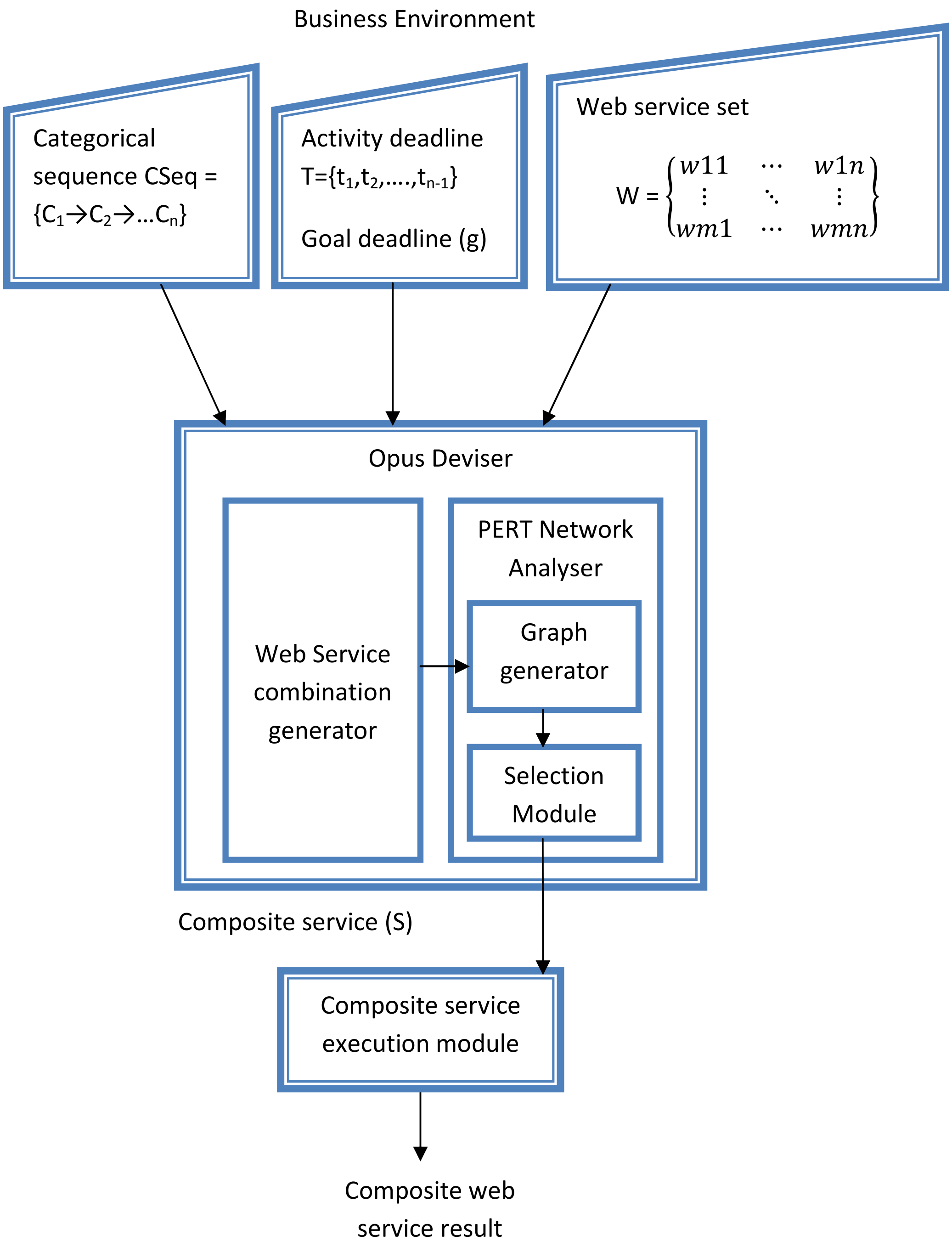

Business Environment
Categorical sequence CSeq = {C1→C2→...Cn}
Activity deadline T={t1,t2,….,tn-1}
Goal deadline (g)
Web service set
W = {w11 … w1n; ⋮ ⋱ ⋮; wm1 … wmn}
Opus Deviser
Web Service combination generator
PERT Network Analyser
Graph generator
Selection Module
Composite service (S)
Composite service execution module
Composite web service result

Fig. 1: Opus Deviser – Block Diagram

h. To plan with the goal deadline assign time under consideration X to goal deadline. X=g

i. Compute the set of standard normal variate
$Z = \{z_1, z_2, \ldots, z_{nc}\}$ for all combinations.
$z_c = (X - \bar{x}_c)/sd_c$ (6) X – Time under consideration (goal deadline)
$\bar{X}_c$ - Mean (Length of critical path)
sd_c – Standard deviation of critical path of a combination.

j. Find the probability that the project will be completed within the completion time from the table of probability values for normal distribution.
$A = \{a_1, a_2, \ldots a_{nc}\}$When $Z \geq 4.09$ the probability of completion becomes 1.

03 Check the set of probability of completion of the project A from step (02 j) and choose the best combination S such that it has high a_c so that the composition is reliable and in case more than one combination has same a_c value check for its Critical path and select the combination with minimum cp_c such that the composition is optimal.

04 Execute the combination S.

05 End

11 Experimentation and Results

Inputs:

CSeq = $\{C_1 \rightarrow C_2 \rightarrow \ldots C_n\}$

$$W = \begin{Bmatrix} C1 & C2 & C3 \\ w11 & w12 & w13 \\ w21 & w22 & w23 \end{Bmatrix}$$

T = { 40, 50 } (min) g = 90 min

Opus Deviser – Web service combination generator

m = 2 and n = 3 nc = $(2C_1)^3 = 8$

Generate 8 possible combinations

Table 1: Table of a, l and b for calculating TC

Activity	a	l	b	Expected Duration
A1	30	50	60	48.33
B1	30	50	60	48.33
A2	30	50	60	48.33
B2	40	60	70	58.33
A3	20	40	40	36.67
B3	20	40	50	38.33
A4	20	40	40	36.67
B4	30	40	40	38.33
A5	30	40	40	38.33
B5	20	40	50	38.33
A6	30	40	40	38.33
B6	30	40	40	38.33
A7	40	55	50	51.67
B7	40	60	70	58.33
A8	40	55	50	51.67
B8	30	50	60	48.33

Calculate expected duration set TC based on a, l and b in minutes.

TC =
{{48.33,48.33},{48.33,58.33},{36.67,38.33},{36.6,38.33},
{38.33,38.33},{38.33,38.33},{51.67,58.33},{51.67,48.33}
}

Opus Deviser - Graph Generator:

Generates the graph and assigns the expected duration set. The graph should not have loops, cross over and recursion. The graphs should be sequential.

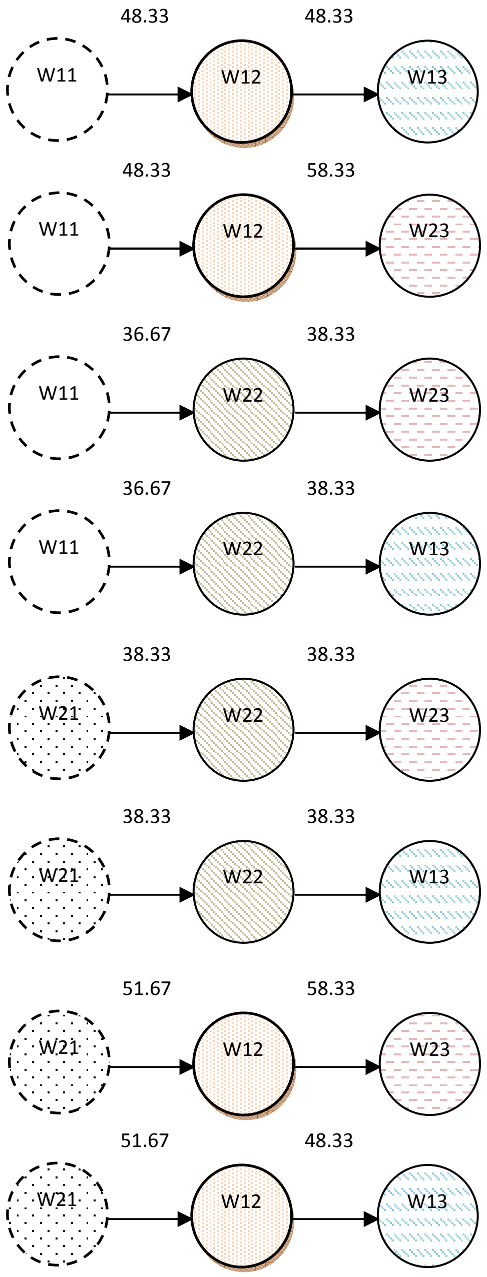


Fig. 2: Graphs for each combination

Calculate the critical path set CP

CP = {96.66,106.66,75,75,76.66,76.66,110,100}

Calculate the variance for each activity in each combination.

VC = {{25,25},{25,25},{11.11,25},{11.11,2.78},{2.78,25},{2.78,2.78},{2.78,25},{2.78,25}}

Total Variance set TV = {50,50,36.11,13.89,27.78,5.56,27.78,27.78}

Assign goal deadline to time under consideration X.

X = 90 min

Calculate standard deviation set SD = {7.07,7.07,6,3.73,5.27,2.36,5.27,5.27}

Calculate the set of Standard normal variate Z

Table 2: Calculation of Z for each combination

X	Length of Critical path	SD	Z
90	96.66	7.07	-0.94
90	106.66	7.07	-2.36
90	75	6	2.50
90	75	3.73	4.02
90	76.66	5.27	2.53
90	76.66	2.36	5.65
90	110	5.27	-3.80
90	100	5.27	-1.90

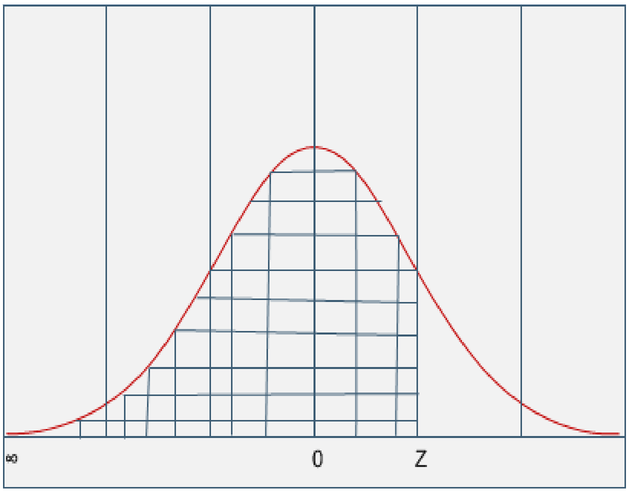


Fig. 3: Area under the curve from ∞ to Z

Calculate the probability of completion of the project from the Z - table of probability values for normal distribution.

area3=0.9938, area4=1, area5=0.9943, area6=1

The selection module finds that combination 4 and 6 are highly reliable as its probability of completion is 1. The module needs to select any one composition for execution. So it looks at the length of the critical path for combination 4 and 6 which is 75 and 76.66 respectively. Composition 4 takes only 75 minutes to complete the composition and therefore is optimal to choose. The composition w11→w22→w13 is selected and executed. In case the probability of completion and the length of the critical path also the same for more than one composition then any one composition is randomly chosen and executed. The compositions with negative Z values are taking more time than the goal deadline and are therefore discarded.

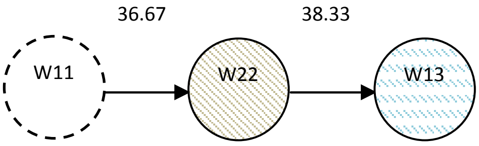


Fig. 4: Selected combination for execution

12 Discussion

In real world the time for completing an activity is not deterministic. The completion time of most of the activities is highly uncertain. PERT treats activity time as a random variable. PERT considers project completion in a probabilistic approach. PERT estimates expected project completion time; probability of completion of project before a goal deadline; the critical path and activity deadlines. Therefore Opus deviser based on PERT is more advantageous than using CPM based service composition planning.

13 Conclusion and Future Enhancement

We have proposed Opus deviser as an automatic, quality driven web service composition algorithm that is in synchronization with business issues that are time driven. We have developed the planner to choose the best composition based on PERT probability predictions which more realistic to CPM based approach. Thus the Opus deviser generates an optimal and a reliable plan. Anyhow PERT is comparatively subjective. PERT does not consider that if some activities are delayed then other paths can become critical. We can enhance the algorithm as a future work using Monte Carlo simulations to handle this issue. Monte Carlo simulation is a method where we analyze uncertainty propagation by determining how random variation, lack of knowledge or errors affect the sensitivity, performance and reliability of the system being modeled [11].

14 References

[1] "Gaining business agility through Service Oriented Architecture", Microsoft Corporation, 2007

[2] Jorg Hoffman, "FF : The Fast Forward planning system", American Association for Artificial Intelligence, 2001

[3] "Graph based planners", University of Huddersfield

[4] Reid Simmons, "Planning, execution and learning: Partial order", Fall 2001

[5] Dana Nau, Tsz-Chiu Au, Okhtay Ilghamy, Ugur Kuter, J. William Murdock, Dan Wu, Fusun Yaman, "SHOP2 : An HTN planning system", Journal of Artificial Intelligence research, 2003

[6] Stefan Edelkamp, Malte Helmerte, "The Model checking Integrated Planning System (MIPS)", www.informatik.unifreiburg.de/~helmert/publications/ AIMag01_**Mips**.pdf, 2001

[7] Keita Fujii, Tatsuya Suda, "Dynamic service composition using semantic information", ICSOC'04, ACM, 2004

[8] LIU An, "RESCUERS : A framework for reliable web service composition", City University Hong Kong, July 2009

[9] "Management control and information", Directorate of studies and research, The institute of company secretaries of India, 1999

[10] Dana S Nau, "Automated planning", Chapter 14 – Temporal planning, Fall 2009

[11] Wittwer J.W., "Monte Carlo simulation basics", Vertex42.com, 2004

Modeling and Designing a Service Oriented Framework for a Comprehensive Emergency System

Samir El-Masri, Hamdan Al-Sabri, Ahmed Ghoneim

College of Computer and Information Sciences, King Saud University, Riyadh, Kingdom of Saudi Arabia

selmasri, halsabri, ghoneim@ksu.edu.sa

Abstract−**Service oriented architecture introduces an effective way for connecting software components and sharing data. Most of the emergency systems lack the methodology to completely integrate and smoothly transfer data within its components. In this paper, we deal with this challenge by modeling and designing a service oriented framework for comprehensive emergency system. The proposed framework is composed of the following distributed subsystems: emergency application for mobile devices, GPS, Ambulances, Main central system, Health record and Hospitals. These subsystems are built using different technologies and completely integrated to handle any emergency event may occur. The framework behavior is illustrated within the internal flow of the collaboration model, which classifies the framework subsystems into provider or consumer. Moreover, the integration between subsystems is shown by handling the business process of emergency events. A real case study presenting the work of the CES framework is discussed.**

Keyword−**Service-Oriented Architecture (SOA), Service Oriented architecture Modeling Language (SoaML), Comprehensive Emergency System (CES), Mobile Web Services.**

1. Introduction

Service Oriented Architecture (SOA) is a pattern or approach that guides all aspects of creating and using the system functions as services throughout their lifecycle. It defines and enhances the integration between different distributed applications to exchange data and functions in business processes regardless of the platforms or programming languages underlying those applications [1]. Web services technology is a standardized way of integrating web-based applications. Web services communicate using open protocols, and based on new standards eXtensible Markup Language (XML), Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Universal Description Discovery and Integration (UDDI). Web services can be used by other applications regardless of the computing environment, in which they are hosted [2], [3]. The need for accessing more services and applications that reside on the web and mobile devices has led to the introduction of Mobile Web Services technology by El-Masri (2005) [4], and El-Masri and Solumein (2005) [5]. In this paper, we present the framework for a Comprehensive Emergency System (CES). The initial concept of CES was proposed in 2005 by El-Masri [17]. The system may be considered more comprehensive than other emergency systems in terms of the number of parties involved, and it is very advanced in terms of the technologies used. The system is also intelligent when it comes to find the right ambulance, hospital and doctor that are suitable for the conditions and location of the accidents. The main advantage and strength of the system comes from the Mobile Web Services technology used in the system. This technology can overcome any problems of interoperability between systems running different applications. The system is flexible enough to receive request for an ambulance from a human being or through a mobile application.

The remainder of this paper is organized as follows; Section 2 will present background and related work and overview SOA and model techniques. In section 3, the proposed framework for Comprehensive Emergency System (CES) and their components will be explained. The case study with an architecture overview and business message flow will be illustrated in section 4. In section 5, conclusions and future research work will be discussed.

2. Background and Related Work

The SoaML metamodel extends the Unified Modeling Language 2.0 (UML) in order to support SOA. This section only presents the concepts of SoaML. The concepts used are based on the revised UPMS submission presented in the literature [6], [18].

[7] presents an approach called AmbientSoaML, which introduces ambients in service oriented architecture modeling language (SoaML). It explains the simple mobility and demonstrates the use of SoaML for modeling SOA of a mobile application. A new model-driven approach is introduced for the generic integration of service-oriented architectures (SOA) and multi-agent systems (MAS) [8]. In another paper, MINERVA was proposed for automating transformations from BPMN to SoaML models in order to automatically generate services from business processes [9]. In [10], the authors analyzed common and widespread service characteristics, derive evaluable design attributes that refer to elements of service designs based on SoaML, and demonstrate the formalization of an exemplarily design attribute using OCL. Another approach was presented with an example of service identification from the Norwegian national Health ICT architecture by using SoaML [11]. [12] shows how SOA modeling and design based on the concept of service component and standard UML modeling constructs and defines service components of different types, scope and granularity. It puts them in the context of a model-driven design approach to provide bidirectional traceability between business requirements and software artifacts. Paper [13] showed the use of shared data models of emergency incidents to support the exchange of data between heterogeneous systems. Summarizes found in [14] on how to use service oriented architecture to lightweight mobile devices [14]. In [15] and [16], the authors focused on investigating the importance of data exchange and message passing on SOA from the security and privacy point of view. Thereafter, they designed a gateway for passing messages in the SOA healthcare platform. Subsequently, they pointed out the interface utilities on the SOA healthcare platform. Healthcare information integration and shared platform based on Service-Oriented Architecture (SOA) was proposed. The platform supports the integration, development, and operation of a full spectrum of healthcare applications.

3. The proposed Framework

A Comprehensive Emergency System (CES) framework is a comprehensive platform to link hospitals, ambulances and computerized operator by transferring patient data and electronic health record in addition to Geographical Positioning System (GPS) location of ambulances, accidents and hospitals. The accident reporter (Emergency requester) device and ambulance systems play the role of mobile web service providers. The abstract view of CES framework illustrates the CES as a black box with inputs and outputs as shown in Figure 1. The comprehensive Emergency System has the following subsystems that they interact together: Mobile Device Application, Main Central System, Ambulance Systems, Electronic Health Record System and Hospital Emergency Systems, as shown in Figure 2. We will summarize the roles of the subsystems as follows:

1) The Mobile Device Application used when there is an accident. Accident reporter (emergency requester) sends an emergency message to MCS.

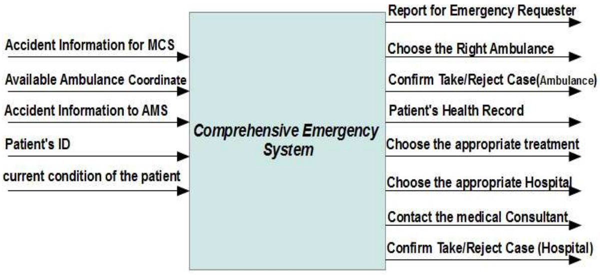

Accident Information for MCS
Available Ambulance Coordinate
Accident Information to AMS
Patient's ID
current condition of the patient
Comprehensive Emergency System
Report for Emergency Requester
Choose the Right Ambulance
Confirm Take/Reject Case(Ambulance)
Patient's Health Record
Choose the appropriate treatment
Choose the appropriate Hospital
Contact the medical Consultant
Confirm Take/Reject Case (Hospital)

Figure 1. The abstract view of the CES architecture

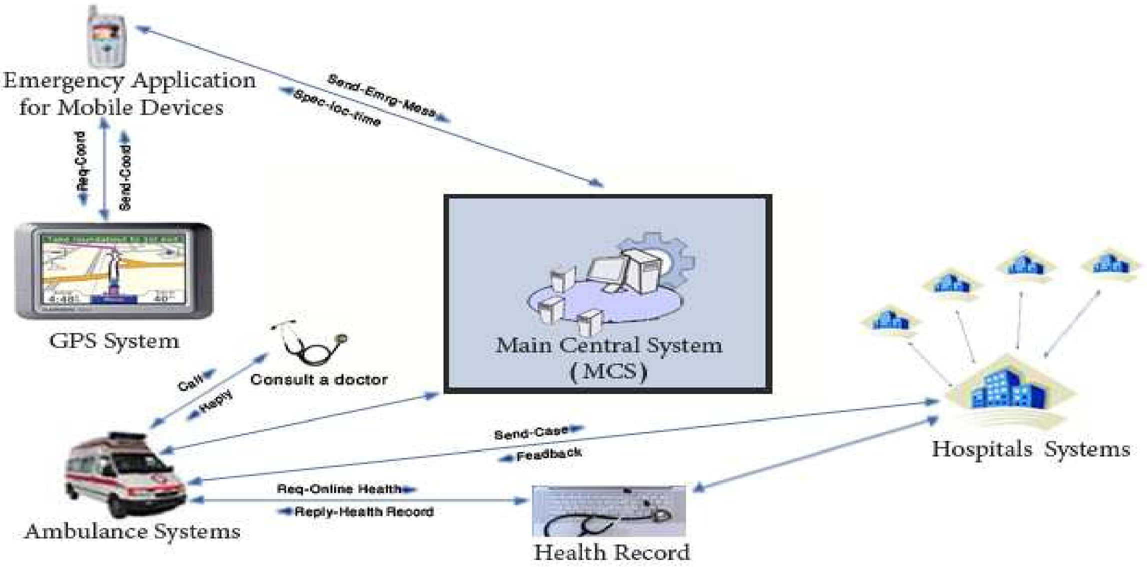


Figure 2. The detailed view of the architecture for CES

2) The Main Central System (MCS): Through the processes in Table 1, will:

- Requests the location for available ambulances in the area that is closed to the accident place. Each ambulance in the system has one of the following states: available, non available, in-mission.
- Calculates the distances using a navigation system between ambulance and accident locations then sends request to the nearest ambulance to deal with the emergency event.
- Sends a report to the initial emergency requester in a matter of a few seconds quoting the approximate time and distance the ambulance needs to reach the accident.

Table 1. MCS processes

```
Input:
    •   Accident Information for MCS (Accident location,
        Number of cars, Number of injured people, date)
    •   Available ambulance coordinates(x, y)
Output:
    •   Report to Emergency Requester (Time, distance),
    •   choose the Right Ambulance
BEGIN
Receive accident information;
For i=1 to i=n {
 // Retrieve ambulances coordinates
If ambulance state ='available' than{
  Get ambulance coordinates ( );
 // insert in to array X;
   }
For i=1 to i=m{
// choose the nearest ambulance to accident
     Calculate distance ( );
     Compare distance ( );
    X= Min(compare distance);
       }
// choose the right ambulance
Send accident information ( ) to X;
//send accident information to the chosen ambulance
  If ambulance-confirm = yes than {
     Send report Informer ( );
Else
   Go to choose the second nearest ambulance. X=X+1;
//select the second nearest ambulance to accident
  Send accident information ( ) to X;
}
END
```

3) The Ambulance System has the capability of using GPS to detect its location, and then send ambulance location to the MCS. When ambulance system receives the MCS request to go to the accident location, it will accept or reject the job. In case of rejection the MCS selects the second nearest ambulance to

accident. In case of acceptance, the ambulance system starts its processes as follows:

- It requests the health record of the patient form the health record center.
- It proposes the appropriate treatment of the patient through a decision support system.
- It finds the specialist, available and the nearest hospital and book a bed in the Emergency Room.

4) The Emergency requester sends accident Information, and then receives a report by MCS.

The Main Central System behavior steps can be summarizes as follows:

1) It consumes available ambulance locations
2) It confirms “take job” from ambulance system,
3) It provides report to emergency requester.
4) It sends accident information to ambulance system. The Health Record Center consumes patient ID and provides patient details.

The CES subsystems are classified into consumer, provider or both roles as shown in Figure 3. The subsystems manipulate two databases: Health Record System and doctor’s information. The main four subsystems play their roles with the life cycle execution of the following web services: emergency event service, handling emergency service, ambulance role service, accessing health record service, hospital availability service, choose consultant service, hospital in action service, and best choose hospital service. In Figure 4, the design simulation illustrates the communications of CES subsystems as web-based behaviors. All these subsystems cooperate and share data using the standard WSDL template for emergency event as shown in Table 2.

Table 2. Structure of the connectivity template

```
<?xml version="1.0" encoding="utf-8"?>
   <Accident-Information>
     <Accident-location>
       <Street-no> 12 </Street-no>
       <Area-no> 1 </Area-no>
       <Coordinates> 335.221, 3225.200
       </Coordinates>
     </ Accident-location>
     <Cars-no>   2 </Cars-no>
     <injured-no> 3 </injured-no>

       </Accident-Information>
  <Report-to-emergency-requester>

      <Time-to-reach> 10 minutes </Time-to-
reach>
      <Distance> 3.5 KM </Distance>

  </Report-to-emergency-requester >
</xml>
```

4. Real Case Study

The case study has been obtained from Riyadh city, the capital of Kingdom of Saudi Arabia. The city has been divided into four areas as shown in Figure 5. The detailed information about each area such as area domain, number of hospitals and others are shown in Table 3. Figure 6 shows the interactions and communications between all the subsystems of the CES in three different scenarios (cases):

1) Case 1: if accident happened in area1 (Al-Nasiriyah) and during the accident there was available ambulance in area1 and patient has broken bones (Available hospital in Area1)
2) Case 2: if accident happened in area1 (Al-Nasiriyah) and during the accident there was not any available ambulance in area1. MCS requested an available ambulance from Area2 and patient has broken bones (Available hospital in Area1).
3) Case 3: if accident happened in area1 (Al-Nasiriyah) and during the accident there was not any available ambulance in area1 and MCS requested an available ambulance from Area2 and patient has broken bones (But there was not any vacancy at Hospital emergency room in Area1).

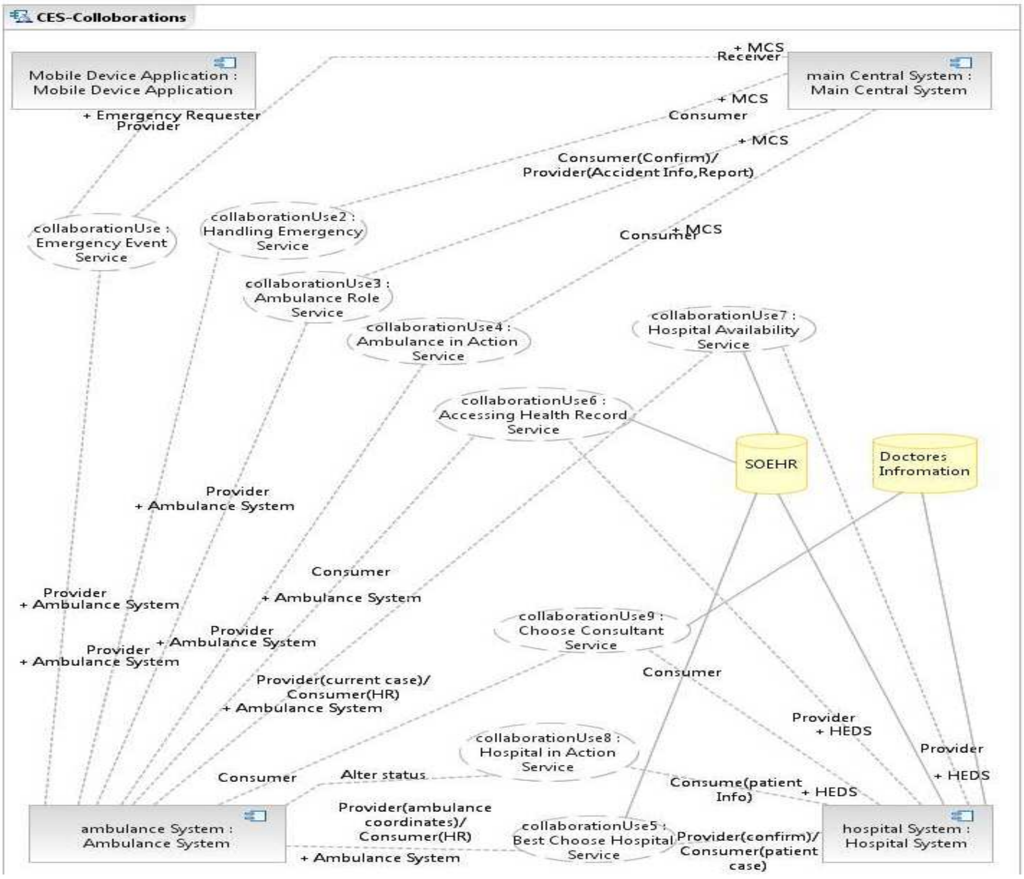

CES-Colloborations
Mobile Device Application : Mobile Device Application
+ Emergency Requester Provider
main Central System : Main Central System
+ MCS Receiver
+ MCS Consumer
+ MCS Consumer(Confirm)/ Provider(Accident Info,Report)
+ MCS Consumer
collaborationUse : Emergency Event Service
collaborationUse2 : Handling Emergency Service
collaborationUse3 : Ambulance Role Service
collaborationUse4 : Ambulance in Action Service
collaborationUse7 : Hospital Availability Service
collaborationUse6 : Accessing Health Record Service
SOEHR
Doctores Infromation
Provider + Ambulance System
Consumer + Ambulance System
Provider + Ambulance System
Provider + Ambulance System
Provider + Ambulance System
collaborationUse9 : Choose Consultant Service
Consumer
Provider(current case)/ Consumer(HR) + Ambulance System
Provider + HEDS
Provider + HEDS
collaborationUse8 : Hospital in Action Service
Consumer
Alter status
Consume(patient Info) + HEDS
Provider(ambulance coordinates)/ Consumer(HR)
+ Ambulance System
collaborationUse5 : Best Choose Hospital Service
Provider(confirm)/ Consumer(patient case)
ambulance System : Ambulance System
hospital System : Hospital System

Figure 3. Service Collaborations of CES

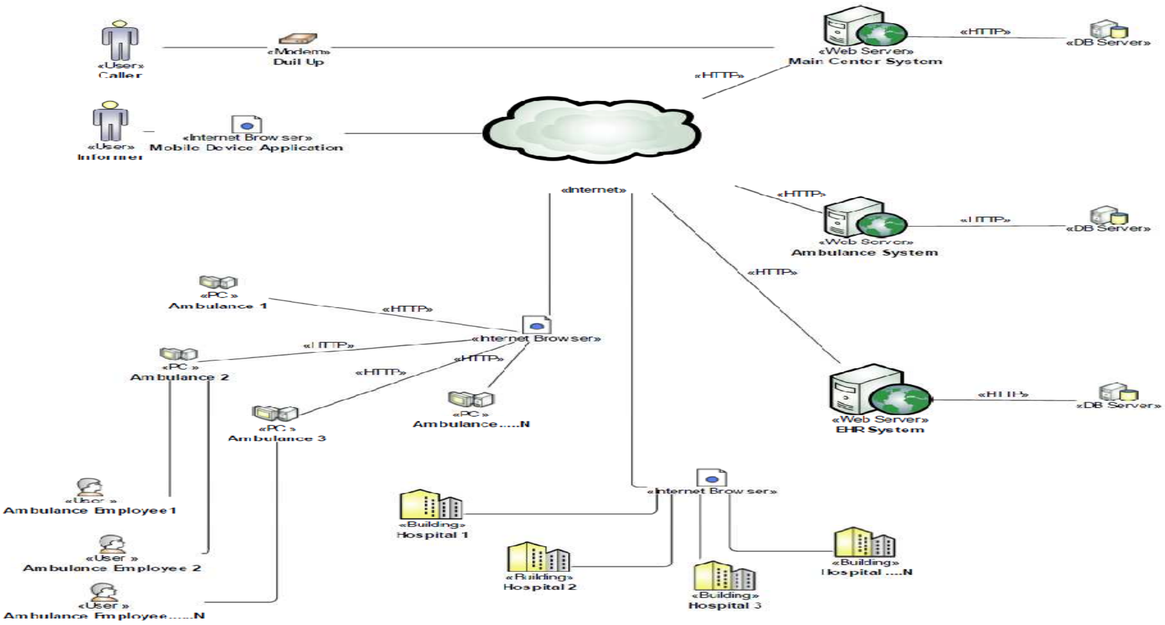


Figure 4. CES subsystems connectivity simulation

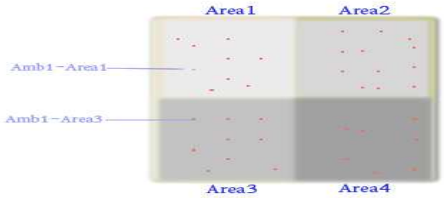


Figure 5. Riyadh four areas

Table 3. Details of Riyadh four areas

Areas	Domain	No. Hospitals	Disciplines
Area1	South West Al-Nasiriyah	3	-General -Eyes -Bones
Area2	North West	4	-Bones -Teeth - Heart - Cancer
Area3	North East	5	-General -Bones -Eyes -Teeth
Area4	South East	3	-General - Heart - Chest

5. Conclusions and further work

In this paper, a new framework that handles an unexpected emergency accident has been proposed. The CES subsystems and their roles have been identified and simulated. The integrated work flow between the CES subsystems and the way to share data has been presented. The used case study obtained from real emergency system services is used to illustrate the applicability of proposed framework.

As for a future work, we plan to use the quality of services (QoS) to test the performance of all subsystems with the CES framework. Also, we plan to apply how to separately control the unexpected events and the subsystems rules from the separated engines. Moreover, we plan to use scripting to model the rules within these engines.

Acknowledgment

This work is part of two year research project which is fully funded by a grant through KACST/ National Plan for Science and Technology in the Kingdom of Saudi Arabia.

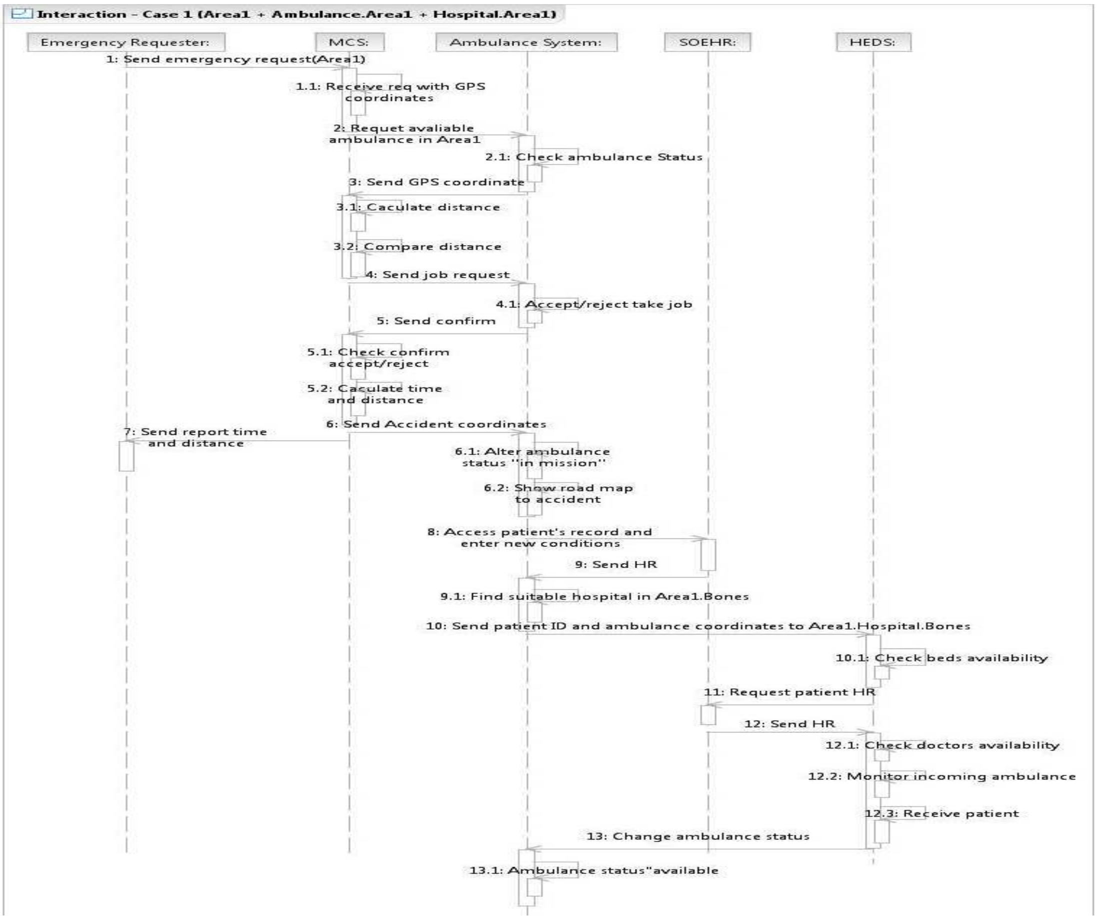

Interaction - Case 1 (Area1 + Ambulance.Area1 + Hospital.Area1)
Emergency Requester:
MCS:
Ambulance System:
SOEHR:
HEDS:
1: Send emergency request(Area1)
1.1: Receive req with GPS coordinates
2: Requet avaliable ambulance in Area1
2.1: Check ambulance Status
3: Send GPS coordinate
3.1: Caculate distance
3.2: Compare distance
4: Send job request
4.1: Accept/reject take job
5: Send confirm
5.1: Check confirm accept/reject
5.2: Caculate time and distance
6: Send Accident coordinates
7: Send report time and distance
6.1: Alter ambulance status "in mission"
6.2: Show road map to accident
8: Access patient's record and enter new conditions
9: Send HR
9.1: Find suitable hospital in Area1.Bones
10: Send patient ID and ambulance coordinates to Area1.Hospital.Bones
10.1: Check beds availability
11: Request patient HR
12: Send HR
12.1: Check doctors availability
12.2: Monitor incoming ambulance
12.3: Receive patient
13: Change ambulance status
13.1: Ambulance status"available

Figure 6. Riyadh decomposition areas and Subsystems interactions

References

[1] Gurpreet Singh Modi, "Seminar Report on Service Oriented Architecture & Web 2.0,"2004.

[2] The World Wide Web consortium W3C, available at: http://www.w3.org/. [Online]. Access date: March 30, 2005.

[3] Web Services Interoperability, available at: http://www.ws-i.org/. [Online]. Access date: March 30, 2005.

[4] Samir El-Masri, "A Framework For Providing Mobile Web Services," School of Computing and IT, University of Western Sydney, Australia.

[5] Samir El-Masri and Basem Suleiman, "A Framework for Providing Mobile Web Services", The second International Conference on Innovations in Information Technology (IIT'05) Dubai, UAE – September 26 – 28, 2005.

[6] Service oriented architecture Modeling Language (SOAML) MDA OMG - Object Modeling Group, 2008.

[7] Ali, N.; Babar, M.A., "Modeling Service Oriented Architectures of Mobile Applications by Extending SoaML with Ambients," 35th Euromicro Conference on Software Engineering and Advanced Applications, Petras, Greece August 27-29, 2009.

[8] Hahn, C.; Jacobi, S.; Raber, D., "Enhancing the Interoperability between Multiagent Systems and Service-Oriented Architectures through a Model-Driven Approach", IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010.

[9] Delgado, A.; García-Rodríguez de Guzman, I.; Ruiz, F.; Piattini, M., "From BPMN business process models to SoaML service models: A transformation-driven approach,", 2nd International Conference on Software Technology and Engineering (ICSTE), 2010.

[10] Gebhart, M.; Baumgartner, M.; Oehlert, S.; Blersch, M.; Abeck, S.; , "Evaluation of Service Designs Based on SoaML," Fifth International Conference on Software Engineering Advances, Nice, France, August 22-27, 2010

[11] Fenglin Han; Moller, E.; Berre, A.J, "Organizational Interoperability Supported through Goal Alignment with BMM and Service Collaboration with SoaML", International Conference on Interoperability for Enterprise Software and Applications, Beijing China, April 21-21, 2009.

[12] Stojanovic, Z.; Dahanayake, A.; Sol, H;, "Modeling and design of service-oriented architecture,", IEEE International Conference on Systems, Man and Cybernetics, 2004.

[13] Hauenstein, L.; Tia Gao; Tsz Wo Sze; Crawford, D.; Alm, A.; White, D.; , "A Cross-Functional Service-Oriented Architecture to Support Real-Time Information Exchange in Emergency Medical Response," *Engineering in Medicine and Biology Society, 28th Annual International Conference of the IEEE*, vol.Supplement, no., pp.6478-6481, Aug. 30 – Sept 3 2006.

[14] Tergujeff, R.; Haajanen, J.; Leppanen, J.; Toivonen, S.; , "Mobile SOA: Service Orientation on Lightweight Mobile Devices", *IEEE International Conference on Web Services*, vol., no., pp.1224-1225, 9-13 July 2007

[15] Chi-Lu Yang; Yeim-Kuan Chang; Chih-Ping Chu; , "A Gateway Design for Message Passing on the SOA Healthcare Platform", *IEEE International Symposium on Service-Oriented System Engineering*, vol., no., pp.178-183, 18-19 Dec. 2008.

[16] Yong-Gang Gong; Xin Chen;, "Healthcare information integration and shared platform based on service-oriented architectures", *2nd International Conference on Signal Processing Systems (ICSPS)*, vol.2, no., pp.V2-523-V2-527, 5-7 July 2010.

[17] Samir El-Masri, "mobile comprehensive emergency system using mobile web services". The second International Conference on Innovations in Information Technology (IIT'05) Dubai, UAE, September 26 – 28, 2005.

[18] A Berre, **"**Service oriented architecture modeling language (SoaML)-specification for the UML profile and metamodel for services (UPMS)" Object Management Group (OMG), 2008.

Towards an UML Based Modeling Language to Design Adaptive Web Services

Chiraz EL Hog, Raoudha Ben Djemaa, and Ikram Amous
MIRACL, ISIMS, Cité El Ons, Route de Tunis Km 10,
Sakiet Ezziet 3021, Sfax, Tunisia

Abstract— *The diversity of Internet users together with the explosive growth of the Web Services, has raised the need for Web Services adaptation. However, existing Web Services are not adapted to the final user profile i.e offered services do not take into account the users diversity and mobility. Therefore, profile adaptation must be suitably managed on the Web Service life cycle. In this paper, we propose a solution for profile adaptation at the design step and introduce an UML profile for Adaptive Web Service that we called AWS-UML (Adaptive Web Service UML). It increases the expressivity of UML by adding labels, graphic, stereotypes and constraints which make it possible the model Adaptive Web Services. We also present a case study to exemplify the application of our UML profile.*

Keywords: Adaptive Web Service, profile, design, meta model, AWS-UML.

1. Introduction

Web Services have emerged as a major technology for deploying automated interactions between heterogeneous systems. They are autonomous software components widely used in various service oriented applications according to their platform-independent nature (e.g., stock quotes, search engine queries, auction monitoring). The Web Services technology allows different applications to be exposed as services via the network and interact with each other through standardized XML-based techniques. These techniques are structured around three major standards: SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), and UDDI (Universal Description, Discovery, and Integration). These standards provide the building blocks on the Web Service life cycle such as description, publication, discovery, and interaction.

The increasing interest on Web Service technology, the growing number of published Web Services and of users profiles have raised new issues in service use. For instance, a Web Service should be able to deliver to the user an adequate service that fulfills each specific user's needs and take into consideration his context. In fact, users can access to these Web Services from various and heterogeneous profiles due to their different interests and preferences. However, a Web Service can be accessed from different locations, through a diversity of devices (laptops, mobile devices, PDA, etc) and network characteristics (Wi-Fi, bandwidth, ...). Users, also want to be able to satisfy their preferences (desired content, layout,...) and interests. According to these heterogeneous, mobile and changing profiles, adaptation is becoming a major requirement which must be taken into account earlier in the Web Service life cycle. This leads to the fact that there is a higher need to automate, at least partially, the design process of Web Services.

Our idea is to provide a generic solution for modeling Adaptive Web Service based on the Unified Modeling Language, UML. In fact, UML is considered as the industry actual standard for modeling software systems. In UML, the structural aspects of software systems are defined as classes, each one formalizes a set of objects with common methods, properties, and behavior. UML can also serve as a foundation for building domain specific languages by specifying stereotypes, which introduce new language primitives by sub typing UML core types, and tagged values, that represent new properties of these primitives. Model elements are assigned to such types by labeling them with the corresponding stereotypes. In addition, UML can also be used as meta modeling language, where UML diagrams are used to formalize the abstract syntax of another modeling language as the work presented in [5] to design adaptive Web Application. Using this opportunity, we aim to define our modeling solution named Adaptive Web Service Unified Modeling Language (AWS-UML).

The rest of the paper is organized as follows. Section 2 reviews literature on adaptive Web Services. Section 3 presents the Use Case diagram of AWS-UML. Section 4 describes the class diagram. Section 5 defines sequence diagram. Section 6 concludes the article and gives some directions for future works.

2. Related works

In this section, we take a look at some research works interested in the possibilities of applying the context adaptation on Web Services life cycle. We provide an overview of some of these works.

El Asri and al.[3] propose a model driven approach for the modeling of user-aware Web Services on the basis of the multiview component concept. The multiview component is a class modeling entity that allows the capture of the various needs of service clients by separating their functional concerns. This work takes into account the profile of the user and his right access to the Web Service functionality. Despite that, the user preferences, device capacity, network characteristic, localization ... are not taken into account.

Sheng and Benatallah[6] present a modeling language for the model-driven development of context-aware Web Services based on the Unified Modeling Language (UML). Although, this work propose a meta model for modeling service context, it don’t care about the actors neither about modeling diagrams.

A number of research efforts have studied Web Services discovery and selection adaptation. Sellami and al.[2] are interested in Web Services discovery in a distributed registry environment. They propose a semantic model to describe Web Services registries (WSRD). This semantic description is functionality driven and is benefit to discover the appropriate service that best fits requester needs. However, it doesn’t take into account the user’s context.

Benaboud and al.[1] have developed a framework for Web Services discovery and selection based on intelligent software agents and ontologies. Ontologies are used to describe Web Services, QOS, customer’s preferences and experiences. But, the proposed framework did not take into account mobile devices with limited capabilities neither network characteristic. Also they have not discussed the impact on Web Services composition.

Balke and Wagner[10] have presented an algorithm for the subsequent selection of appropriate services. This algorithm features an expansion of the service request by user-specific demands and wishes. Services not matching a certain profile are discarded on the fly and equally useful results of alternative services can be compared with respect to user provided strategies. They have not deal with different client devices, using several types of networks (wireless, local, etc.) various networks characteristics, user location...

Soukkarieh and Sedes[4] have proposed a new architecture of Web Services, supporting the adaptation process to the user context and returning to the user a list of Web Services adapted to his context. They extend the architecture of AHA combining it with the classical architecture of Web Service and adding to these architectures an adaptation layer containing various components dedicated to context adaptation.

Other researches have studied Web Services interaction adaptation.

Pashtan and al.[8] present an information system for tourism (CATIS) which enables the adaptation of mobile devices in terms of content and presentation. The application takes into account the user preferences, his localization and the type of his terminal.

Keidl and al.[9], present a generic framework to support the development of context-aware adaptive Web Services. The transfer of context information is performed through SOAP message header. Context information can be explicitly and directly processed by clients or Web Services or be automatically handled by the context framework. However, contexts are limited to the information of service requesters.

Most of works previously studied deal separately different steps of Web Services life cycle. However, some researchers concern on one particular form of adaptation, like semantic enrichment of the client request by his context Benaboud and al.[1]. This request enrichment is not enough to deliver adaptive Web Services. It is needed also to integrate Web Service description the context in which it is adapted. In addition, contextual information partially covers the user general context (El Asri and al.[3], Sheng and Benatallah[6]). Other works have presented specific solutions to a range of use or type equipment used (proposed platform for the field of tourism and adaptation only affects mobile devices in the work of Pashtan and al.[8]). Moreover, proposed works when adapting execution services, do not bring solutions to all general types of media used, but trying to provide solutions to needs very specific.

As we emphasized in the preceding paragraph, in order to provide the user with the most relevant services to his context, we must take user’s profile into account earlier on the Web Service life cycle, essentially on the modeling step.

3. Use Case diagram of AWS-UML

A Use Case diagram is used to describe functionalities provided by a system in terms of actors, their goals represented as Use Cases, and any dependencies among those Use Cases. The UML Use Case diagram meta model, defines one class to model actors and one for Use Cases. This definition doesn’t fit well our need to design Adaptive Web Service. Therefore we propose enriching this meta model by specifying three kinds of actors that could interact with our Web Service and three kinds of Use Cases that describe functionalities according to the variety of actors.

3.1 Actors of AWS-UML

An actor specifies a role played by an external entity that interacts with the system. Those actors can be humans, other computers, pieces of hardware, or even other software systems. The only criterion is that they must be external to the part of the system being partitioned into Use Cases. They must supply stimuli to that part of the system, and the must receive outputs from it.

However, Web Service can be accessed through different ways and by a variety of profiles: the service supplier, the service human client and application client. To model this distinction, we propose three categories of actors:

1) Application Consumer: This actor is used to model a software that interacts with the Web Service. It could be:
   a) A Composite Web Service: This actor is used in the case of services composition. It concerns

requests of users that cannot be satisfied by any available Web Service, whereas a composite service obtained by combining a set of available Web Services might be used. In that case, a Web Service can play the role of a client to another one.

b) A Web Application: This actor is used to model a web application that uses a Web Service to accomplish its functionalities. Through a servlet, a Web application can connect to a Web Service using it's URI and access account.

2) Human Consumer: This actor is used to model a human requester using a Web Service by a web URL. He is an Internet user who interacts with service from the web (weather service, prayer times service, currency converter...). He can also, express his preferences and interests to customize service results.

3) Provider: This actor is used to model the Web Service provider. He is a person or an organization that supplies services over service registry. He could create, update, deploy (add the service to Web Service registry) or undeploy the Web Service.

The table 1 shows icons used to design these actors.

Table 1: Actors of AWS-UML

Provider	Human Consumer	Application Consumer	Composite Web Service	Web Application

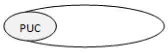
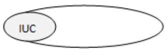
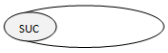
3.2 Use Cases of AWS-UML

The Web Service architecture is based on the interaction between three components: service provider, service registry (the broker) for storing service descriptions, and service requester (the client). These interactions are based on publish, find, and bind operations. According to these operations and by the increasing needs of Web Service consumer, their higher mobility and various interests, we distinguish three classes of Use Cases as follows:

1) Use Case Service Interaction: Used to model interaction with the Web Service. It includes request sent to a Web Service, response received from a Web Service, subscription, specify preferences and interests.
2) Use Case Service Publication: Used to add or remove Web Service on or from the service registry. It includes create, modify, describe, deploy (publish Web Service on the service registry) and undeploy (retrieve service from the service registry).
3) Use Case Service Search: Used to search and select an adequate Web Service.

To distinguish graphically between these three Use Cases we adopt following specific notation on table 2.

Table 2: Use Cases of AWS-UML

Publication Use Case	Interaction Use Case	Search Use Case
		

3.3 Meta model of the Use Case diagram of AWS-UML

The standard UML meta model doesn't allow to model the variety of actors and Use Cases presented above. So, we present a Use Case meta model of AWS-UML depicted in Figure.1.

By analogy to UML, this meta model is represented by the two extended concepts: Actor and Use Cases. Actors (Provider, Human Consumer and Application Consumer) inherit from the *Actor* class in UML meta model. Use Cases (Use Case Service Interaction, Use Case Service Publication and Use Case Service Search) inherit from the *UseCase* class in UML meta model. AWS-UML extensions are presented by classes with font color grey.

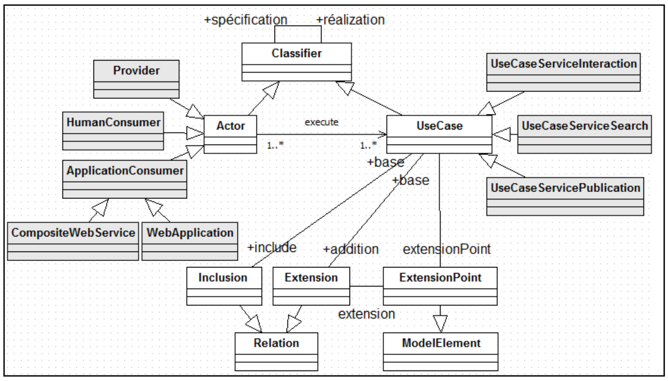


Fig. 1: Meta model of the Use Case diagram of AWS-UML

To validate our model we propose to add the following OCL (Object Constraint Language) constraints:

- An actor *Provider* can have association with a Use Case *UseCaseServiceInteraction*, *UseCaseServicePublication* and *UseCaseServiceSearch* ***Context*** *Provider: self.execute → forAll (a | a.oclIsKindOf (UseCaseServiceInteraction) or a.oclIsKindOf (UseCaseServicePublication) or a.oclIsKindOf (UseCaseServiceSearch))*
- An actor *HumanConsumer* can have association with *UseCaseServiceInteraction* and *UseCaseServiceSearch* ***Context*** *HumanConsumer : self.execute → forAll (a | a.oclIsKindOf (UseCaseServiceInteraction) or a.oclIsKindOf (UseCaseServiceSearch))*
- An actor *ApplicationConsumer* can have association with a Use Case *UseCaseServiceInteraction* ***Context*** *ApplicationConsumer : self.execute → forAll*

(a | a.oclIsKindOf (UseCaseServiceInteraction) or a.oclIsKindOf (UseCaseServiceSearch))

3.4 Examples of Use Case diagram of AWS-UML

To illustrate our proposal, we will give an example of an Adaptive Web Service designed with AWS-UML. The example is a Travel Agency Web Service.
Figure 2 illustrates an application consumer Use Case for the travel agency Web Service.

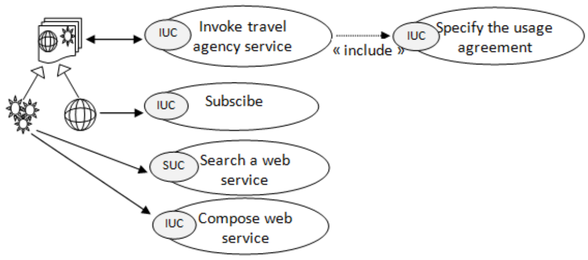


Fig. 2: Example of Use Case diagram of AWS-UML

The web application actor and the composite Web Service actor can invoke the travel agency service specifying usage agreement. The web application can subscribe with the service. The composite Web Service can search or compose with other Web Services.

Figure 3 illustrates a provider Use Case for the travel agency Web Service. The actor provider creates or modifies the travel agency Web Service. He provides to customer the ability to book complete vacation packages: airline service, hotel service are used to query their offerings and perform reservations and credit card service used to guarantee payments made by consumers. The provider actor may also, deploy or undeploy the service. Moreover, he could be a travel agency service consumer.

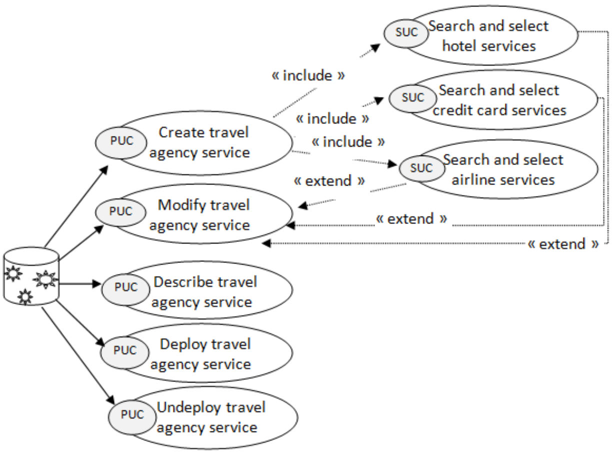

SUC
Search and select hotel services
« include »
SUC
Search and select credit card services
« include »
« include »
PUC
Create travel agency service
SUC
Search and select airline services
« extend »
PUC
Modify travel agency service
« extend »
« extend »
PUC
Describe travel agency service
PUC
Deploy travel agency service
PUC
Undeploy travel agency service

Fig. 3: Example of Use Case diagram of AWS-UML

4. Class Diagram of AWS-UML

Class diagrams are the mainstay of object-oriented analysis and design. They show the classes of the Web Service, their interrelationships (including inheritance, aggregation, and association), and the operations and attributes of the classes. A Web Service may interact with several actors with a variety of profiles. Each profile has specific needs on this service. In order to ensure the flexibility and the adaptability of services, a service must take into account the profile in interaction with it. The problem is how to model these various actors needs when designing class diagram of a Web Service. In order to tackle this problem, we will exploit the concept of VUML (View based Unified Modeling Language) introduced by Nassar [7] and enrich the class diagram meta model with user's profiles. The view concept is largely used in several fields, as a mean of separation of concern, such as Database Management System and Workflow. It helps in writing software that is modularized by concern. For our work, we use views as a means of both assuring functional separation of concern and managing access right. The concept of VUML revolves around two key concepts: Base and View.

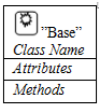
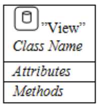
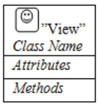
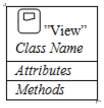
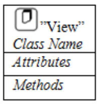
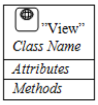
1) Base: Is a core entity which includes specifications that are common to all types of actor.
2) Views: Are used as a means of assuring functional separation of concern and managing access rights.

In our case of study, we define a multiview service as a first class modeling entity that highlights the actors needs and requirements. In order to retrieve the most relevant results with the user's context, we should discover all context elements that influence the result. These elements are enclosed on actor's profiles. The *Base* class allows the representation of the functionalities required by all kinds of users. In contrast, the *View* class allows the representation of the functionalities required by a specific kind of user. These functionalities are accessible only if the specific user is in interaction with the service. We distinguish one Base and three kind of Views:

1) Application Consumer View: Is an abstract view class to model functionalities allowed to an Application. It depends on the Application Profile. This abstract view contains two views: a Composite Web Service View and a Web Application View.
2) Provider View: Is a view class to model functionnalies allowed to a Web Service provider.
3) Human Consumer View: Is a view class to model functionalities allowed to a human consumer. It depends on the Consumer Profile.

Profile characteristics are also represented within the class diagram and they will be traited by the WSDL file. To model the Base and Views on the class diagram of AWS-UML, we define following stereotypes and icons on table 3:

Table 3: Base and Views of AWS-UML

Base	Provider View	Human Consumer View	Application Abstract View	Composite Web Service View	Web Application View
 "Base" Class Name Attributes Methods	 "View" Class Name Attributes Methods	 "View" Class Name Attributes Methods	 "View" Class Name Attributes Methods	 "View" Class Name Attributes Methods	 "View" Class Name Attributes Methods

4.1 Meta model of class diagram of AWS-UML

Modeling elements of the UML class diagram are: the class and the association. In AWS-UML, we need to extend these standard elements to be able to design concern based class diagrams. As extensions we define a *MultiviewsClass* inherited from the *Classifier* and consists of one *Base* and different *Views* and *Abstract Views*. The relation between *Base* and *Views* is achieved through the *ViewExtension* inherited from *Association*. *Views* are related to specific profiles describing actors characteristics.
Figure 4 shows the meta model of this class diagram. Extensions are colored in grey.

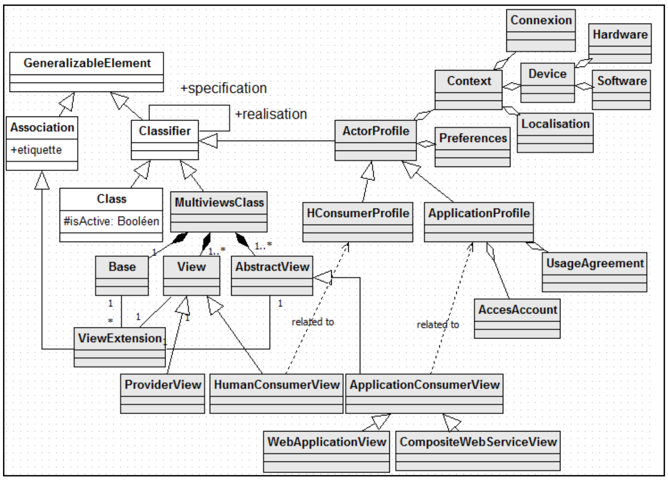


Fig. 4: Meta model of the Class diagram of AWS-UML

Associations in this model obey certain rules that we define using OCL.

- An element *ViewExtension* is an association between a departure element *Base* and a receiver element *View* or *AbstractView*
  context ExtensionView inv : (self.client.oclIsKindOf (View) or self.client.oclIsKindOf (AbstractView)) and (self.supplier.oclIsKindOf (Base))
- An element *View* can inherit from a *View* or an *AbstractView*
  ***context** View inv : self.generalization → forAll (g : Generalization | g.parent.oclIsKindOf(View) or g.parent.oclIsKindOf(AbstractView))*

4.2 Example of class diagram of AWS-UML

In this section, we will design a class diagram of a travel agency service using AWS-UML. The class stereotyped *Base* represents the common behavior. Classes stereotyped *View* represent specific behavior. Views are related to profiles. The Human Consumer View is related to a consumer profile defined by user preferences and the context of use.

1) User preferences specify the display preferences and the content preferences.
2) The context of use contains:
   a) The localisation defined by latitude and longitude,
   b) The connection characteristics defined by the bandwidth, type and the debit,
   c) The device characteristics: software (browser, operating system) and hardware (type, desktop, processor, memory, graphics card).

The Application View is related to an application profile defined by an access account and an usage agreement between the Web Service and the application:

1) Access account defined by login, password and date of validy of the account,
2) Usage agreement defined by the list of input parameters, the list of output parameters and the access url.

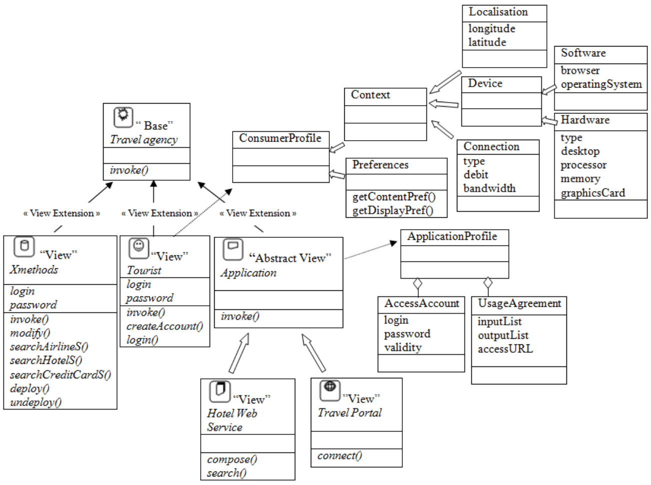


Fig. 5: Example of class diagram of AWS-UML

5. Sequence diagram of AWS-UML

UML sequence diagrams are used to model the flow of messages, events and actions between the objects or components of a system. It presents interactions that can exist between the user and the Web Service functionnalities. This interaction, which is achieved from exchanges of messages, is related to actor's profile. The UML sequence diagram doesn't allow to model objects and instance message for an Adaptive Web Service. We need more precision to model objects and Messages according to their behavior.

5.1 Concepts of the sequence diagram

Objects of the sequence diagram are class instances with specific operations and attributes. We propose in AWS-UML five kinds of objects described by new icons:

- ObjectMultiviews: presenting the Web Service interface,
- ObjectBase: presenting commun behavior of the Web Service,
- ObjectProvider: presenting the supplier activities, the deployment process, etc,
- ObjectHumanConsumer: presenting the human user of the Web Service,
- ObjectApplicationConsumer: presenting a software user of the Web Service witch contains ObjectCompositeWebService and ObjectWebApplication.

These objects interact while exchanging messages that can be simple messages or profile messages (containing profile features) labeled by the corresponding type:

- PublicationMessage: presenting publication activities like deploying, undeploying, creating, updating, etc,
- SearchMessage: presenting searching and selecting Web Service,
- InteractionMessage: presenting all the execution process,
- ProfileSearchMessage: providing search preferences or context parameters to objects,
- ProfileInteractionMessage: providing profile characteristics to objects of the execution process.

5.2 Meta model of the sequence diagram of AWS-UML

Figure 6 illustrates AWS-UML sequence diagram extension to take into account our definition of objects and messages. AWS-UML objects are described by meta classes that inherits from the *Object* UML meta class. And AWS-UML messages are described by meta classes that inherits from the *InstanceMessage* UML meta class. Extensions are colored with font color grey.

Our sequence diagram meta model must obey to the following OCL constraints:

- An *ObjectMultiviews* can only send messages to an *ObjectBase* or another *ObjectMultiviews*
  context *ObjectMultiviews: self.send → forAll (o | o.oclIsKindOf (ObjectBase) or o.oclIsKindOf (ObjectMultiviews))*
- An *ObjectMultiviews* can receive messages from an *ObjectBase* or from ObjectApplicationConsumerView or *ObjectHumanConsumerView* or ObjectProviderView or another *ObjectMultiviews*
  context *ObjectMultiviews: self.receiver → forAll (o | o.oclIsKindOf (ObjectBase) or o.oclIsKindOf (ObjectMultiviews) or o.oclIsKindOf (ObjectApplicationConsumerView) or o.oclIsKindOf (ObjectHumanConsumerView) or o.oclIsKindOf (ObjectProviderView))*

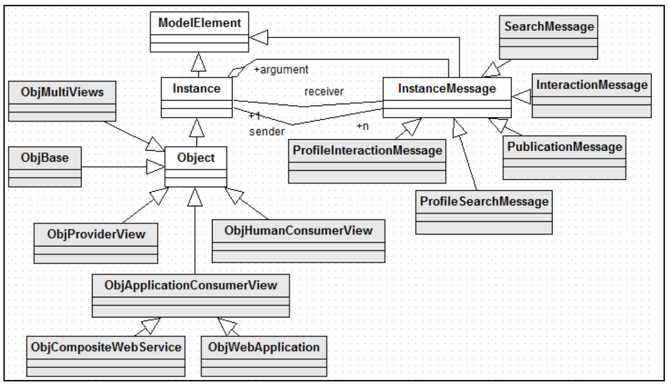


Fig. 6: Meta model of the Sequence diagram of AWS-UML

5.3 Example of sequence diagram

Next, we consider the interactions between the actor Travel Portal Web Application and the Travel Agency Web Service. We focus on the Use Case: *Invoke travel agency service* shown in Figure 2 and describe a sequence diagram example illustrating messages exchanged. Table 4 shows icons used to model objects and Figure 7 depicts the sequence diagram.

Table 4: Icons of the Sequence diagram example

Object Multiviews	WSDL file	Object Base	Object Web Application View
⧉	🗋	⊡	◉

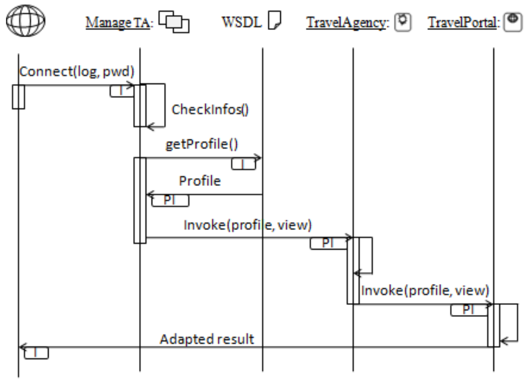


Fig. 7: Example of Sequence diagram of AWS-UML

The Travel Portal component is a Web Application allowing to access to a variety of the travel agency services. It can connect and invoke our Travel Agency Web Service. In order connect to the service, the web application sends a message of type interaction to the object multiviews *Manage Travel Agency*. This latest checks the validity of connection informations and then recovers the profile of the web application from the WSDL file. Then it transfers the profile and the corresponding view to the object base *Travel Agency* within a profile interaction message. Common functionalities are executed by the *Travel Agency* base object witch transfers the execution process to the corresponding view within a profile interaction message. Finally, the web application *Travel Portal View* object executes the adequate operations and sends adapted result to the client.

Messages exchanged are labeled with the corresponding letters as they can be simple messages or profile messages.

6. Conclusion

The adaptability and flexibility are challenging issues in the world of Web Services. In this regard, we have presented in this paper, an effective modeling language for the design of highly adaptive and flexible Web Services named AWS-UML. We have introduced the meta model and notation of the language and illustrated its usage using an example of a Travel Agency Web Service.

AWS-UML is a UML profile for Adaptive Web Services. This extension of UML defines a set of stereotypes, constraints and graphic annotations to allow us to design adaptive Web Services. Along this paper, we have focused on Use Case diagram, Class diagram and Sequence diagram.

Firstly, we have presented actors and Use Cases. We have introduced various classes of actors that can interact with Web Services and are looking for specific needs. We have also defined different Use Cases related to adaptive user needs.

Secondly, we have exploited the multi-view concept which highlights the users needs and requirements by separating their concerns on the class diagram. We have defined the structure and the functionalities of a service according to the actors which are likely to use it. Besides, we presented actors profiles related to each view.

Finally, we have presented our sequence diagram proposal. We have defined more objects and message instances to achieve the goal of designing adaptive interaction.

Diagrams proposed above bring answers to reach insufficiencies of the UML language diagrams to model adaptive Web Services. In the near future, we aim to validate our proposal by proposing an AGL that supports new AWS-UML extensions. It is based on the open source tool: ArgoUML. Then we will focus on the Web Service implementation and deployment process to be able to generate adequate Web Service Descriptor file.

References

[1] R. Benaboud, R. Maamri, Z. Sahnoun, “User’s preferences and experiences based Web Service discovery using ontologies,” in *Proc.ICRCIS’10*, 2010, p 121-126.

[2] M. Sellami, O. Bouchaala, W. Gaaloul, S. Tata, “WSRD: A Web Services Registry Description,” in *Proc.ICNDST’10*, 2010, p 89-96.

[3] B. El Asri, A. Kenzi, M. Nassar, A. Kriouile, A. Barrahmoune, “Multiview Components for User-Aware Web Services,” in *Proc.ICEIS’09*, 2009, p 196-207.

[4] B. Soukkarieh, F. Sedes, “Towards an Adaptive Web Information System Based on Web Services,” in *Proc.ICAS’08*, 2008, p 272-277.

[5] R. B. Djemaa , I. Amous and A. B. Hamadou, “Extending a Conceptual Modeling Language For Adaptive Web Applications,” *International Journal of Intelligent Information Technologies.*, vol. 4, pp. 37-56, 2008.

[6] Q. Z. Sheng, B. Benatallah, “ContextUML: A UML-Based Modeling Language for Model-Driven Development of Context-Aware Web Services,” in *Proc.ICMB’05*, 2005, p 206-212.

[7] M. Nassar, “Analyse conception par points de vue : le profil VUML,” PhD Thesis, National Institute Polytechnique of Toulouse, Toulouse, France, Sep, 2005.

[8] A. Pashtan, A. Heusser, P. Scheuermann, “Personal Service Areas for Mobile Web Applications,” *IEEE Internet Computing archive*, vol. 8, pp. 34-39, 2004.

[9] M. Keidl, A. Kemper, U. Passau, “Towards Context-Aware Adaptable Web Services,” in *Proc.IWWWC’04*, 2004, p 55-65.

[10] W. T. Balke, M. Wagner, “Towards Personalized Selection of Web Services,” in *Proc.IWWWC’03*, 2003.

Designing a Drug-drug Interactions Web Service

Jian-xun Chen[1]
[1] Department of Information Management, Chang Jung Christian University, Tainan, Taiwan, R.O.C.

Abstract - *When two or more drugs are prescribed simultaneously, it may produce drug-drug interactions (DDIs). Current solution to this problem is to download DDIs information from Department of Health and uploaded it to the hospital information system database for on-line usage. This approach will suffer from duplicating investment for large hospitals and lacking such information technology resources for small health units. Since cloud computing advocates software as a service in recent years, we can provide DDIs to all medical institutions by web services. We use the Web Services Business Process Language (WS-BPEL) as our process design tool to reduce the complexity of designing web service. Through web services, the DDIs function can be integrated with heterogeneous computerized order entry system easily, and can solve follow-up maintenance problem. In this way, we can save huge health care resources while guard the drug safety.*

Keywords: Drug-drug interactions, Web services, WS-BPEL

1 Introduction

Based on patient’s conditions, drug prescriptions are the primary treatment to cure diseases. A recent research indicated that an average of 3.6 drugs per out-patient nationwide [1]. When two or more drugs are prescribed at the same time, drugs may change pharmacokinetics or effects of its original design and may lead to adverse reactions. This is so-called drug-drug interactions (DDIs) [2,3]. The more drugs are prescribed, the more DDIs may occur [4]. Because a wide range of drug is available for prescription, in a limit out-patient service time, to rely solely on physicians or pharmacists to examine the complex DDIs are impossible. Therefore, hospitals are downloaded DDIs data from the Department of Health and then uploaded it to their own hospital information system database. After that, they wrote programs to check the DDIs. When prescribing, if the DDIs are found, the system will give appropriate prompts to remind physicians to correct this prescription. This approach improves medication safety considerably.

However, primary health care units, such as clinics, pharmacies or nursing home, are the first line gatekeeper of our health, but they lack of talent and resources to build such DDIs checking mechanism in their computerized Physician Order Entry (CPOE) system. Even they are willing to invest resources to develop such function in their CPOE, but follow-up maintenance issue is arisen to be another problem. In fact, due to various medical institutions have their own practical requirements, CPOE systems are heterogeneous generally, so there is no one single system can solve the all institution’s problem. In such circumstances, our drug safety system appears to have a big loophole. Thus, we need an open access approach to check drug interactions.

In recent years, cloud computing promotes the concept of software as a service (SaaS) [5]. Under the progress of information technology now, this idea has become reality. For example, in October 2007, the largest internet search engine company Google and IBM announced the Blue Cloud to establish a common global data center (http://www.pcworld.com/businesscenter/article/138195/google_ibm_promote_cloud_ computing.html). Microsoft launch a new operating system, Azure (http://www.microsoft.com/ windowsazure/), to provides various kind of Live Services. Therefore, if we can provide DDIs checking service, it will be able to effectively solve the above issue.

2 Literature Reviews

2.1 Drug-drug Interactions (DDIs)

National study showed that potential drug interactions of outpatient prescription rate was 15.99% in 2000 and Digoxin is the most common drug interactions. Shad et al. (2001) estimated that the additional medical expenditures of DDIs were 17,213 U.S. dollars for a single case. Additionally, according to a study from the Health Sciences Center of University of Arizona, the nation one year improper drug usages cases, such as drug side effects, drug overdose, not the right medicine and drug interactions are nearly twenty million people. In 2004, U.S. Congress authorized the Institute of Medicine (IOM) to study the issues of drug safety and quality. The published document of IOM, "Preventing medication errors", reported that about 150 million people in the United States suffered medication error injury each year, and the additional cost caused by it was up to 35 billion U.S. dollars, of which at least a quarter of medication errors can be avoided [6], and these errors can be prevented by information technology in the future.

Because a large amount of drugs is available for prescription in medical institutions, relying on physicians or pharmacists to find out the DDIs during short prescription time is impossible. At present, most hospitals had invested resources in designing DDIs prevention function. Its basic procedure is like Figure 1. The Department of Health (DOH) provided DDI's data for downloading and the information stuffs of hospitals imported this dataset to their HIS database. They then wrote programs to check DDIs on-line against this dataset. Tips or suggestion will be provided when not serious prescription problem is found. For example, duplicate drugs, avoiding combination, monitoring cardiac function and so on. While there is a serious drug interaction found, the system will prohibit this prescription.

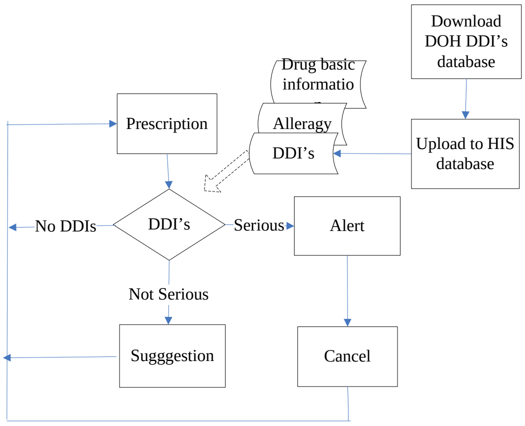
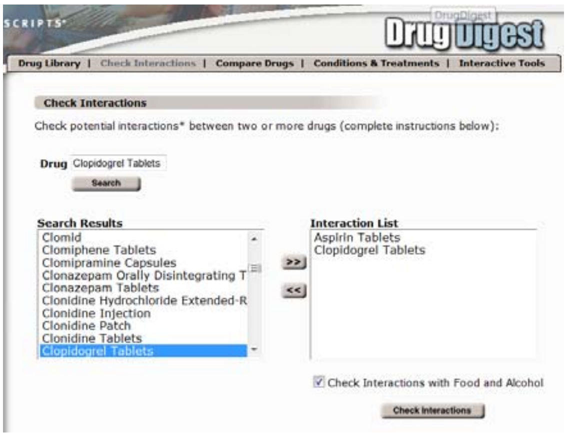


Figure 1. The basic flow of checking DDIs

For primary health care units, what they have is a simple version of CPOE and they have no resources to incorporate the DDIs mechanism into their CPOE system. So currently, if they want to check DDIs, they will go to some public DDIs web to check DDIs (Please refer to Figure 2 and 3, http://www.drugdigest.org/wps/portal). But this approach requires re-entry prescriptions to the web page and thus is inconvenient to use.


SCRIPTS
DrugDigest
Drug Digest
Drug Library | Check Interactions | Compare Drugs | Conditions & Treatments | Interactive Tools
Search
>>
<<
Check Interactions

Check Interactions

Check potential interactions* between two or more drugs (complete instructions below):

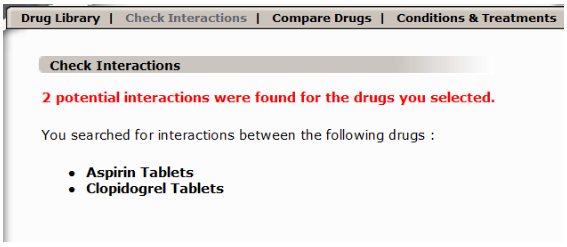
Drug Clopidogrel Tablets

Search Results
Clomid
Clomiphene Tablets
Clomipramine Capsules
Clonazepam Orally Disintegrating T
Clonazepam Tablets
Clonidine Hydrochloride Extended-R
Clonidine Injection
Clonidine Patch
Clonidine Tablets
Clopidogrel Tablets

Interaction List
Aspirin Tablets
Clopidogrel Tablets

☑ Check Interactions with Food and Alcohol

Figure 2. A sample DDIs checking web


Drug Library | Check Interactions | Compare Drugs | Conditions & Treatments

Check Interactions

2 potential interactions were found for the drugs you selected.

You searched for interactions between the following drugs :

- Aspirin Tablets
- Clopidogrel Tablets

Figure 3. The Results of DDIs checking

2.2 Web Services Business Process Execution Language (WS-BPEL)

Web service is a software component and it goes to standard web protocols and data formats (such as HTTP, XML, SOAP, WSDL and UDDI, etc.) to provide service for other applications [7]. Based on these standards and protocol, web service is consists of three main modules, i.e. Service Provider, Service Requester and the Service Registry. The Service Provider provides services and the implementation environment of the services. The Service Requester is a program or an application which search and use web services in internet. The UDDI Broker is a database for service providers to register their public information of a provided service. For enterprises, web service provides a standard interface to execute remote procedures call for local applications, so programmers can use programming language they familiar to access web service [8]

According to above illustration, it seems that web service is an easier way to incorporate heterogeneous COPE system, but it might cost a lot of efforts to familiar with the mechanism of web services. In order to reduce the complexity of designing web service, Web Services Business Process Execution Language (WS-BPEL) is a widely used in commercial business process management system (BPMS) to design web service. It is used to describe high level business processes, and this description is mapping to a set of interactions between web services. In other word, it can be used to integrate a group of loosely coupled web services to a new web service in internet, so that the new service can meet the requirement of business processes [9, 10, 11]. Its predecessor is BPEL for Web Services (BPEL4WS) from the Microsoft and Web Services Flow Language (WSFL) from IBM. Combination of these two standards, they submitted it to OASIS organization and it is released as the WS-BPEL finally.

Important BPEL activities and elements of WS-BPEL include: Invoke, Receive and Reply are responsible for links with the Web Service and data exchange. Assign handles data to variable. Catch, Throw, Waiting and Terminate are used to process errors. Additionally, Flow control elements include: Sequence, Flow, Switch and While to deal with the sequences of execution. By using these elements, we can build up web service more easily.

3 Research Methodology

In the BPEL implementation, there are many development tools, such as Microsoft BizTalk, Oracle BPEL Eclipse, and so on. However, due to Eclipse supports developing BPEL and WSDL document and it is an open source software which we can find more resources in internet, we use the Eclipse and its plug-ins, WS-BPEL Design, as our development platform. Overall development environment is listed in Table 1. In Table 1, Apache ODE is used to execute business processes written by WS-BPEL standard. It talks to web services, sending and receiving messages, handling data manipulation and error recovery as described by our process definition.

Table 1. Development environment

Operating System	Windows XP Professional
HTTP Server	Apache 2.0.59
Orchestration Engine	Apache ODE 1.3.5
Web Service Server	Tomcat 5.5.29
Web Service Development Platform	Eclipse 3.5.0
Web Services Design	Eclipse WS BPEL 0.5.0
WSDL	WSDL 1.1
SOAP	SOAP 1.1

Based on the above technology, our proposed system is like Figure 4. As soon as we integrated Eclipse and web service plug-ins, we are able to mimic web service server at the host where we design our web service. So after we designed and deployed DDIs web service in this platform, client program can use the generated WSDL document from the Eclipse to write web service calling program. This is just like we query UDDI server to obtain web service description. In Figure 4, we can see apache ODE (Orchestration Director Engine) play the role of executing business processes written following the WS-BPEL standard. Basically, it needs to send and receive messages between web services as described by our process definition.

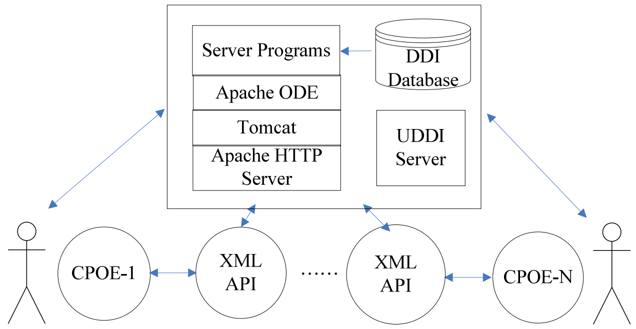


Figure 4. The proposed system environment

4 Results

In reality, the stuffs of medical institutions use brand name of medicines more frequently than generic name. But the generic name of a medicine may have different brand names. For example, the generic name of Aspirn is Aspirin and its brand name may be Aspirin tab. 500mg, Bokey cap. Aspirin 1000mg or Docodon cap.1000mg. So before we can process the request of DDIs checking, we need to transform the brand name to generic name. considering the convenience of using this web service,

Main part of our designing of DDIs is showed below.

(1) BPEL document

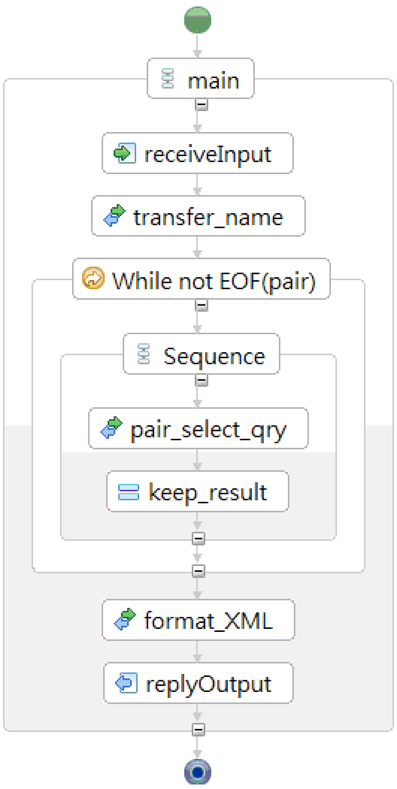


Figure 5. The result of DDI’s BPEL

In the main process (Figure 5), the receiveInput receives client requests and then trigger a sequence of activities to process prescriptions. Fist of all, the “transfer_name” service will translate non-generic name of medicine to generic name for further use. After that, “pair_select_qry” service will select pair of our medicine list in turn and use the paired medicine to query DDIs database to if DDI exists. If DDIs is found, this results is keep temporally in a file. When all medicines are process, then the temporal file is passed to next service, “format_XML” to arrange as a XML format document so that client program can retrieve the meaning of DDIs document. The replyOutput will send the final results back to the client program.

(2) The deployment of the BPEL

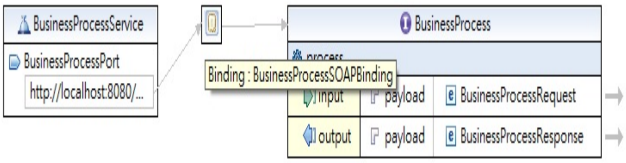


Figure 6. Teh deployment of DDIs web service

Then, we established the Binding links between Service and Process (Figure 6). In order to communicate with the BusinessProcessService, we defined the PortType as BusinessProcess and its protocol is SOAP.

5 Conclusions

The aim of this study is to apply web service and WS-BPEL to construct an open source platform to provide DDIs checking service in internet. The result of this research shows the feasibility of applying web services in drug safety issues. Through the web WS-BPEL, our research also presents a general model depicting the benefits of using WS-BPEL to design a web service. By WS-BPEL, designing web services become more like a traditional process design and can use and integrate existing web service easily. However, in internet, how to ensure the safety of every operation is an ongoing challenge. Another issue needs to be care is the scalability problem caused by a lot of users invoking the DDIs service in the same time. Using commercial web service platforms, such Amazon Web Service (AWS) or Google App Engine (GAE) may be a solution of this problem.

6 Acknowledgment

This research was supported by grants from National Science Council of R.O.C. under contract NSC 99-2410-H-309 -009.

7 References

[1] Huang, Meizhen. "The investigation of high risk drug interaction prescriptions”, research report, Pingtung Branch of National Health Insurance Bureau, Taiwan, 2007.

[2] Tatro, D. S.. “Drug Interaction Facts 2000”, Wolters Kluwer Health, Inc.. August, 1999.

[3] Lafata, J. E., Schultz, L,. Simpkins, J., Chan, K. A., Horn, J. R., Kaatz, S., Long, C., Platt, R., Raebel, M. A., Smith, D. H., Xi, H., Yood, M. U.. “Potential drug-drug interactions in the outpatient setting”; Medical Care, Vol. 44, No. 6, 534--541, 2006,

[4] Weideman, R. A., McKinney, W. P., and Bernstein, I. H.. “Predictors of potential drug interactions”; Hospital Pharmacy, Vol. 33, 835--840, 1998.

[5] Mladen, A. V., “Cloud Computing – Issues, Research and Implementations”. North Carolina State University, USA, 2008.

[6] Institute of Medicine. “Preventing Medication Errors: Quality Chasm Series”. Washington, DC: National Academy Pr, 2006.

[7] Chopra, V., Zoran, Z., Damschen, G., Dix, C. and Cauldwell, P.. “Professional XML Web Services”. Wrox Press, 2001.

[8] Booch, G.. “The Visual Modeling of Software Architecture for the Enterprise”. 1998.

[9] Kloppmann, M., Koenig, D., Leymann, F., Rickayzen, A., Schmidt, P. and Trickovic, I. “WS-BPEL Extension for People(BPEL4People)”. From http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-bpel4people/BPEL4People_v1.pdf. Retrieved May 6, 2011.

[10] Yan, J., Li, Z., Yuan, Y., Sun, W., & Zhang1, J.. “BPEL4WS Unit Testing: Test Case Generation Using a Concurrent Path Analysis Approach”. Poster session presented at the 17th International Symposium on Software Reliability Engineering (ISSRE 06), USA, North Carolina, 2006.

[11] Yaoa,Y., Eversb,P. T., Dresner, M. E. “Supply chain integration in vendor-managed inventory”. Decision Support System, Vol. 43, 663–- 674. 2007.

Metrics for Web Programming Frameworks

Daniel Walker and Ali Orooji
Dept. of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
polyesterhat@gmail.com
orooji@eecs.ucf.edu
Phone: (407)823-2341
Fax: (407)823-5419

Abstract - *Many languages and techniques exist under the umbrella of programming. Web development comprises a small subset of the whole. Without JavaScript, CSS, HTML, Databases like MySQL, and several server side languages such as PHP, the internet would be far less user friendly. Since a vast amount of code is duplicated between projects, web application frameworks were born. Some of these frameworks are basic and others are very feature-rich. This paper provides some metrics for evaluating the MVC (model-view-controller) based frameworks. More specifically, four frameworks are examined: CakePHP, Django, Ruby on Rails, and ASP.NET MVC.*

Keywords: Web Programming Frameworks; Model-View-Controller (MVC); Apache, Python, Ruby, PHP, and MySQL; CakePHP, Django, ASP.NET MVC

1 Why MVC?

The basic concept/idea in MVC is that there are three layers to a dynamic website. The database is the bottom layer which contains records modeling a certain type of data. The layer which the user sees is the view. The controller is the logical glue between the model and view, determining what data to send from the database to the view. The controller can be thought of as a two-way layer, and is divided into actions; each representing an actual page which a user can visit.

Building a modern website can be related to building a cargo ship in the olden days. The ship has many inherent expectations, like the ability to float and store food and supplies. Building the entire ship by hand would be labor intensive; a builder would struggle with even the simplest task like crafting 700,000 nails, or a set of cast iron cannons. However, he could instead hire a company which specializes in this field. This company employs experts with specific tools and materials optimized for the task. The company meets these standards through optimized methodology. Cannons must be molded from cast iron; this is a standard. If a company did not follow this standard, instead using hollowed out palm trees, havoc would ensue. A well-built ship is a vast combination of engineering from multiple sources and industries.

Similarly, a modern website developer needs to employ the right web framework. If the wrong company builds the ship, it may sink. Many web standards exist, and a framework should assist the developer in meeting these standards. A developer will not use “Old Joe’s Framework” if Old Joe did not meet standards in the framework. MVC organizes a web project very nicely; when something better comes along, MVC communities may shift their focus.

2 Installation

In general, a framework aims to assist the developer rather than supply a list of chores; a framework needs to be easy to install. A painless install means project specific code begins right away. In the early days of programing, tasks such as installation required a computer’s command line. Though many developers still prefer a command line, there are now more user-friendly methods. CakePHP, Django, and ASP.NET MVC are very easy to install. For example, to install CakePHP, a developer downloads and moves the folder structure to a virtual host. Django is installed the same way as CakePHP. For ASP.NET, simply, install a Microsoft IDE (integrated development environment). The trickiest part is installing the dependencies. The table below provides the major dependencies of each framework.

Framework	Dependencies
CakePHP	Apache PHP Database
Django	Apache Python Database
Ruby on Rails	Apache with

	Phusion Passenger, etc. Ruby RubyGems MySQL or PostgreSQL
ASP.NET MVC	Visual Studio .NET or Visual Web Developer Express Microsoft SQLServer or MySQL

Table 1: Framework Dependencies

Ruby on Rails requires Ruby, the language, and RubyGems. RubyGems is a module installation system for Ruby that is used to download and install Rails (the web framework). It is important to remember that Ruby and Python are standalone programming languages; PHP and ASP.NET were specifically designed for the web. Mac OS X comes with PHP, Ruby, and Python installed; third party software such as Mono can be used to program ASP.NET on a Unix or Linux environment. An apple computer can use a Bootcamp partition to utilize MS Visual Studio. A windows' user would need to install a server, PHP, Ruby, Python and the ASP.NET IDE manually. Each framework has a website with detailed installation instructions, and each has easy as well as tricky parts. Once the dependencies are installed, steps become simpler.

3 Learning Curve

Unless a developer feels like learning a whole new language, picking up a web framework which uses a language he already knows would be the easiest. If a developer does not know any web languages, then he should learn one before beginning to use a framework. Frameworks make complex things easier, but as soon as something abnormal is required, a good knowledge of general programming is required. There is an extra learning curve when it comes to a web framework.

Frameworks contain a series of shortcuts and methodologies built on top of the language itself. For example, in an MVC, the URL of a webpage is simply the name of a function in the controller object; whereas, without a framework, a file with that name, containing its own HTML, server-side logic, and database queries would be required; there would be a similar file for every page on the entire site. A framework speeds things up, and has many functions developers must become familiar with. Within a few weeks, one would be comfortable with a framework.

The frameworks covered in this paper are the best candidates for each language in terms of their popularity and their meeting of the standard MVC features. However, a good website built in Zend PHP could still be better than a website built in CakePHP. There are many frameworks because, as soon as a developer is unhappy with an existing one, he makes his own. For an expert at CakePHP, it would be easy to learn ASP.NET MVC. Django and ruby are closely related in terms of syntax; as are PHP and ASP.NET. ASP.NET is the strictest when it comes to OOP because it is most like Java. PHP is generally considered a scripting language, not a full blown OO language. So, a developer who does not know OOP may benefit from PHP.

4 Core Library

If someone is looking for a book on a certain topic, they will probably go to the largest library nearby. A larger library is more capable of getting the needed book. Similarly, every web framework has a library containing convenience methods for the developer. Generally, a larger core library is more capable of helping a task. Many of these methods are originally written for use by the developers who made the framework; but included for public use within the framework. For example, CakePHP has a utility library called Set, which adds union and intersection ability for arrays. CakePHP also has a core helper library called HtmlHelper, which assists in generating html code for a view.

A core library on an MVC is built on top of the library functions of the language itself, and all successful web languages have vast libraries. For example, ASP.NET has many built-in functions similar to Java allowing string, array and file manipulations. ASP.NET and Django both have an interesting example of a core library; they auto generate an administrative section for a website project. This section is a web page which allows the managing of users and groups as well as access control for the actions within the project. The Django automatic admin page also allows editing of all content in the database of the project. In other words, it looks at the database schema and generates "scaffolded" controllers and views to edit that data; since it generates the code, a developer can then customize it. This adds a huge jump start to any project in Django. CakePHP does something similar with a "bake script," but is far more basic than Django. This bake script reads the database configuration and builds out "scaffolded" code for the entire website; which can then be edited.

Since there are so many functions already in web languages, a framework only needs to add a higher level functionality such as html helpers, database interaction libraries, and authentication libraries.

5 Object Relational Mapping

Frameworks use a combination of object relational mapping (ORM) and a database abstraction layer (DBAL) to completely eliminate the need for custom SQL. The framework queries a database table for the schema, then dynamically builds objects within the programming language for the developer which model the schema of the database object. The ORM is the set of functions which represent a record in the database. The DBAL is the set of properties given to the ORM object allowing it to perform CRUD operations (create, read, update, delete) on itself. For example, in CakePHP one could perform the following:

```
// this is DBAL
$s1 = $this->Student->findByName('Bob');
// this is ORM
$s1['Student']['name'] = 'Bobby';
$this->Student->save($s1);
// name of Bob is changed and saved
```

Figure 1: ORM / DBAL Example

The process is similar in each of the four frameworks. Without a framework, building a query is a very manual process; not to mention processing through the returned data. As Django's documentation says "A model class represents a database table, and an instance of that class represents a particular record in the database table. To create an object, instantiate it using keyword arguments to the model class, then call save() to save it to the database." A developer can also use plain SQL at times in any of these frameworks if they prefer.

6 Unit Testing

Test driven development means using small asserted statements to test a controller or model for expected output. In fact, using unit testing, a developer could make all the controller actions of a website before making any views. Many developers test simply by refreshing a webpage, but these frameworks allow a more direct approach. ASP.NET programmers may have it the easiest, as unit testing is simply a menu item within Visual Studio .NET. Rails has an easy guide here: http://guides.rubyonrails.org/testing.html. Rails, Django and CakePHP use "Fixtures" which are sample data sets used for the tests which are easy to make. Each of the four frameworks allow third party unit testing libraries, but Django and Ruby may be easier for this than CakePHP and ASP.NET. CakePHP considers a third party class a vendor; and including them into a project can be a hassle. However, CakePHP's built-in unit testing is beyond satisfactory.

7 JavaScript Included

Currently, the most popular JavaScript library online is jQuery. Whether or not a framework comes with some prepackaged JavaScript modules like Ajax loading or effects has no bearing on whether jQuery or another library can be used. Using a JavaScript library is easy in any framework. Each framework has a folder for web resources, CakePHP calls it "webroot", Django calls it "scripts", etc. If a developer does not want to learn JavaScript, he should still stay away from the preconfigured JavaScript modules within web frameworks; custom functionality requires much custom code. In other words, some frameworks come with pre-built support for prototype.js, or Mootools or jQuery, but the point is: no amount of pre-built JavaScript modules will make for a good website. If it needs to be functional on the client-side, much custom JavaScript will be required.

8 Documentation

For good learning, there must be a good teacher. Each of these frameworks have thorough documentation, detailing each aspect of developing with the framework. The documents backing CakePHP, Rails, and Django have similar documentation; all are very high quality. ASP.NET MVC documentation is very different. ASP.NET MVC was built when Microsoft decided to make its already popular web framework ASP.NET into an MVC model; only adding enough code to force it into the MVC pattern. The bulk of the core library documentation is within the documentation for ASP.NET, not ASP.NET MVC.

Since each of these frameworks sits on a popular language, their documentation also sits on the documentation of the language. In other words, something a developer needs in Django, may already be part of Python. Familiarity with the language is a must. Each of these frameworks has two types of docs, the API or api and the tutorials. The api lists every method of each library within the framework while the tutorials show examples of common tasks. Programming languages generally have the api only; but many supporting community websites. CakePHP has a third type of documentation called the Bakery which lists third party tutorials. Also, so many blogs exist online that every aspect of every common task of all four frameworks is thoroughly hammered out. Going purely by aesthetics, the ASP.NET MVC documentation website gets less praises compared to the Rails, CakePHP and Django website.

9 Community

The community is the backbone of a popular framework; the community encourages growth for the framework while maintaining standards. There are major differences between the communities of large frameworks; sports teams have similar fan-bases. The more someone uses a framework, or the less they have used other frameworks, the more their preferred framework is elevated in their mind. With every framework, some people come and go, while others would die before leaving. However, these people are the reason beginners' questions do not go unanswered. The community of supporters for a framework is made of the people who interact and communicate about it on blog websites. It seems that a larger percent of the community of ASP.NET are in foreign countries.

All communities have knowledgeable members. However, it does seem that more people who use Rails and CakePHP like to talk about what they made, more so than people who use Django. People who use ASP.NET seem to only use it for work and therefore only talk about the problems they are having with it. One thing the community loves to do is show off their code. The ASP website is mostly in-house examples. Here are some galleries of framework code:

Framework	Website
CakePHP	http://bakery.cakephp.org/
Django	http://djangopackages.com/
Rails	http://rubygems.org/
ASP.NET MVC	http://www.asp.net/mvc

Table 2: Framework Websites

10 Updates to the Framework

Each of the four frameworks studied in this paper are updated regularly. In fact, regular updates was a required metric for inclusion within our study. It does not matter which framework has the highest version, or even the number of previous versions. The most important aspect is whether or not there has been a steady improvement upon the framework. Also, the language which the framework is built on needs to be regularly maintained and updated. The real reason a developer would want an update is for future projects. Sometimes, updating an old project with a new version of a framework can waste time. CakePHP just released 1.3 which deprecated several old functions and methodologies which would break older projects. Django has a 1.3 release candidate (http://docs.djangoproject.com/en/dev/releases/1.3/). Release candidates are great because developers commit fixes and features before the alpha release.

11 Reusability of Parts (Plugins, Helpers, etc.)

The idea behind web frameworks is that a developer won't have to retype all kinds of common code between projects. What about the custom code that he wants to share between projects or with other developers? The table below provides, for each framework, the different reusable code formats.

Framework	Module Names
CakePHP	Component, Behavior, Helper, Element, Plugin (all encompassing)
Django	Egg, App, Extension, Middleware
Rails	Gems (Sometimes called Plugins)
ASP.NET	Plugins

Table 3: Framework Modularity

CakePHP has many options for breaking up the reusable functionality. Each of the names in the table is essentially a directory within an app. So, sharing with other developers becomes easy. Each framework has different docs outlining how to use and make reusable parts within the framework. Python itself has eggs, similar to how Ruby has gems. Eggs and Gems are not part of the framework, but part of the language. Django has extensions and middleware. There are slight distinctions between all of these items, yet they only become relevant when a programmer finds he is knee deep in a project.

12 Conclusions

There are nearly 100 popular frameworks for about 10 popular web programming languages. The metrics offered in this paper should be a mere starting point for choosing a framework. However, the only true metric will be whether a developer is satisfied with the framework after spending time with it. The MVC methodology should be considered in high regard; frameworks which do not follow it get left behind by communities. When all is said and done, knowing a language is the most important part to using a framework.

13 References

[1] CakePHP http://cakephp.org/
[2] Django http://www.djangoproject.com/
[3] Ruby on Rails http://rubyonrails.org/
[4] ASP.NET MVC http://www.asp.net/mvc/
[5] ASP.NET MVC Overview Tutorial http://www.asp.net/mvc/tutorials/asp-net-mvc-overview-cs
[6] ASP.NET MVC Music Store Tutorial http://www.asp.net/mvc/tutorials/mvc-music-store-part-1
[7] Django Applications http://en.wikipedia.org/wiki/Django_(web_framework)#Applications_built_on_Django

SESSION

SEMANTIC WEB + TOOLS AND APPLICATIONS

Chair(s)

TBA

QAComPS: A Quality-aware Federated Computational Semantic Web Service for Computational Modellers

Peter M. Dew (1) and Shahzad Nizamani (1, 2)
[1] School of Computing University of Leeds LS2 9JT, UK
[2] Mehran University of Engineering & Technology, Pakistan
p.m.dew@leeds.ac.uk, scsan@leeds.ac.uk

Abstract*—This research sets out to help computational modellers to select the most cost effective Cloud service provider. This is when they opt to use cloud computing in preference to in-house HPC facilities. Cloud computing is a pay-per-use model for accessing computing resources from a variety of service providers such as Amazon EC2. Increasingly cloud providers are offering the high performance computing options that are necessary for computational modellers.*

The paper is concerned with a quality-aware computational broker (QABroker). The QABroker service federates across specific service packages offered by a selected set of computational cloud providers that potentially meet the user's computational resource and QoS requirements. These vary during the various stages of the computational modelling cycle. The core of the QABroker is a novel Quality-aware federated computational semantic Web service (QAComPS). This includes an integrated ontology-based system that makes use of OWL2 features. This is used to filter the cloud providers' services into three groups. These are: High, Medium and Low quality of service. This classification is then used by a MatchMaker to automatically select the highest ranked service that meets the user requirements.

A SAWSDL interface was used to transfer semantic annotations to/from the QAComPS service and QABroker. Early evaluation of the QAComPS service was very promising and demonstrates its potential to make cloud computing more accessible and cost effective for computational modellers.

Keywords
Cloud computing, quality aware service, semantic Web, SAWSDL, computational modelling and broker mediation

I. Introduction

The motivation for this research was to help computational modellers to have cost-effective access to computational cloud services. Computational modellers address complex, real-world problems through building computerized models of physical phenomena. Their modelling often requires access to HPC facilities. Typically, their computational and quality of service needs vary during the modelling life cycle. This research therefore focuses on developing a quality-aware computational broker (QABroker) that automates for the user the process of selecting and running a cloud provider service that meets the user's computational and Quality of Service (QoS) requirements throughout the lifetime of the model and development.

Cloud computing [1] provides predictable and flexible on-demand pay-per-use access to a shared pool of computing resources (e.g. networks, servers, storage facilities, applications and services). In this research we were concerned solely with IaaS (Infrastructure as a Service) cloud service providers. Each cloud computing service provider offers the user a choice of different VMs (Virtual Machines). A VM emulates a physical machine.It is performed by hardware virtualization where a physical machine is used for creating VMs. Each VM has processor, memory, storage and other resources. Price of a VM depends on the allocated resources (e.g. the amount of run-time memory and the number of CPU cycles).

The user therefore has to choose between cloud computing service providers and also between the VMs they offer. This research considers the cloud providers: Amazon EC2 [2], Rackspace [3] and FlexiScale [4]. Amazon EC2 now offers a clustered HPC option so it is reasonable to assume other cloud providers will also offer comparable HPC services. An early comparative study of HPC cloud providers is given in [5] and [6] reports their experience of using EC2 for HPC.

This paper is concerned with a computational cloud broker service called the QABroker that mediates across a selected set of computational cloud providers. The broker service includes a cost structure that incorporates QoS metrics such as reliability, user satisfaction (or reputation), cost and security. The role of the QABroker is to automate the selection process of cloud providers and their associated VMs. The selection process takes account of the user's computational resource and QoS requirements at that point in time. The user receives the selected VMs without needing to know about the provider.

A key step in building the QABroker was to design and evaluate the *Quality-aware federated computational semantic Web service (QAComPS).* The main features of QaComPS are:

1. A federated cloud provider's ontology service to integrate the information on the QoS and cloud provider's resources with associated costs;
2. An automatic (agent) selection process to discover the best VM that meets the computational modellers QoS and resources requirements;

3. A semantic annotation for web service description language (SAWSDL) interface between the Broker information and QAComPS service, for example to update the QoS metrics.

A primary evaluation study was performed to demonstrate the feasibly and value of the QAComPS service. The paper is structured in the following way: §2 provide the details on the proposed QABroker service architecture, in which the novel QAComPS service resides. The following section provides details of the QAComPS semantic Web service including the architecture and ontology. This is followed in §4 with the evaluation study and key details of the implementation. §5 covers the relevant background literature and the last section provides the conclusions and recommendations for future work.

II. QA Cloud Broker

a. QA-Aware Cloud Services

A number of papers around 2003/2004 discuss QoS issues in Web Services (e.g. see [7-9] and more recently [10]). Service ontology provides a consistent semantic data model for describing QoS metrics that are non-functional properties. The two types of QoS are “best–efforts” and “guaranteed service”. Here we use “best-efforts” that is referred to as *Quality-awareness* where the service provider can: (1) just drop the service in the case of overload; and (2) provide no guarantees concerning the response time, job throughput etc.). Today most public cloud providers only offer quality aware services. This paper also provides a quality aware service.

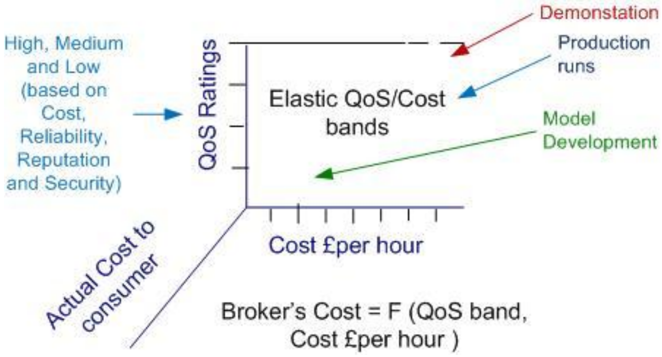

Demonstation
Production runs
High, Medium and Low (based on Cost, Reliability, Reputation and Security)
QoS Ratings
Elastic QoS/Cost bands
Model Development
Cost £per hour
Actual Cost to consumer
Broker's Cost = F (QoS band, Cost £per hour)

Figure 1 Cost Model

Figure 1 shows the use of QoS and the cloud cost. Four non-functional QoS metrics were considered:

Cost *C(p)*, Reputation R_p *(p)*, Reliability $R_l(p)$ and Security *S(p)*

where *p* is a provider and each rating is on a rating scale of 1 – 9 with 1equating to lowest quality and 9 to highest quality. Selection of these illustrative and other QoS metris can be added/deleted as required in the future. The cost QoS is computed from the cost model described below. The reputation and reliability ratings of the VM used can be updated with quantitative data after each computational modelling run. Security ratings are updated qualitatively. A MatchMaker matches the user’s QoS requirements against all the VMs that the different providers offer and subsequently ranks the VMs from best to worst on the closeness of the match. This technique is widely used for selecting Web Services. In our case the goal is to automatically select the VM that meets the user’s computational resource and QoS requirement at the lowest cost. For fuller details on the QoS metrics and the MatchMaker the reader is referred to paper [9, 10]. As it can be difficult for the user to specify the exact QoS they require they are asked to indicate whether they desire a high, medium or low QoS. They are also required to state the relative importance of the each QoS metric for a job. The QABroker can be adapted to incorporate additional levels. Three levels were used in our evaluation study.

b. Case Study

At the University of Leeds, computational modellers have been experiencing frustrations in two areas. Firstly, they feel disadvantaged by the way local HPC facilities schedule jobs with a run-time of twelve hours or more (a feature of much of their work). The turnaround time of these jobs can be unpredictable depending on the size of the HPC job queue and long job runs may be limited to weekends. Secondly, they find the price and reliability of the service. These are inflexible and do not cater very well for their computational service needs as they vary throughout the model development process. For example in the early stages of the modelling, when job runs tend to be short and each one has relatively little importance depending on the outcome, the developer may be happy to accept some reduction in service reliability in order to have lower run time costs. However, the longer the run time for a job, the more important it is that the run is successfully executed at the first attempt. In addition, at certain times, for example when demonstrating to project sponsors, the reliability of the service is crucial, and for these job runs, the modeller is likely to prefer to pay a premium rate for very high reliability.

Figure 1 illustrates the way cost and quality of service are related and indicates the likely preferences of the computational modeller during three phases (model development, production runs and demonstration). The QABroker would discover the user’s computational and QoS preferences at each stage of the life cycle of the model. This information would enable the broker to make a set of VM selections. Each choice would meet the modeller’s computational and QoS requirements at the lowest cost.

QABroker Service Architecture

Figure 2 shows our envisioned QABroker, the environment in which the QAComPS semantic Web service is used. The QABroker cloud service is managed by an external Broker. To illustrate the services figure 2 shows two Web services:

- Broker RUN Web Service provides the infrastructure to enable the computational modeller to run their job on the selected cloud;
- Broker information Web service manages the information for the QAComPS service.

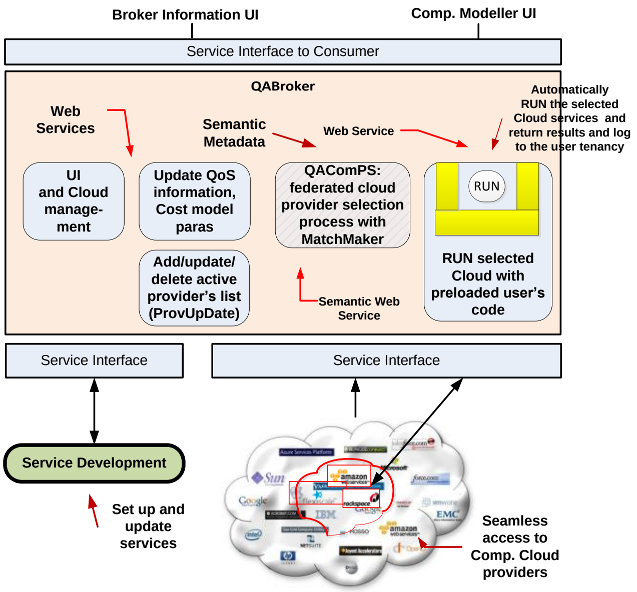

Broker Information UI
Comp. Modeller UI
Service Interface to Consumer
QABroker
Web Services
Semantic Metadata
Web Service
Automatically RUN the selected Cloud services and return results and log to the user tenancy
UI and Cloud manage-ment
Update QoS information, Cost model paras
QAComPS: federated cloud provider selection process with MatchMaker
RUN
RUN selected Cloud with preloaded user's code
Add/update/ delete active provider's list (ProvUpDate)
Semantic Web Service
Service Interface
Service Interface
Service Development
Set up and update services
Seamless access to Comp. Cloud providers

Figure 2 QA Cloud Services

The QAComPS semantic Web service is the main service discussed in this paper. Its purpose is to select the VM that best meets the user's QoS and computational requirements. This required enhancing the capability of the MatchMaker to automatically make VM selections. This would more precisely meet the user's requirements than had previously been possible. This QAComPS service is fully described below. It makes use of external Web services as shown in the diagram.

These Web services are used for two purposes: (1) To access the world of cloud providers, from which the Broker selects, a set of suitable computational cloud providers*;* (2) Where possible the Broker transparently accesses the provider's information from their WSDL document or API. The Broker service monitors each provider for any changes to the information it holds about each provider and updates its records at each Broker service break.

In addition to the QAComPS the QABroker internal services include: (1) the UI and management (not developed further in this paper); (2) a QABroker information service to dynamically manage updates about the QoS of each VM. The information in these updates is periodically passed across to the QAComPS service; (3) a RUN Service that provides the framework to run the selected cloud provider's VM; and (4) The ProvUpDate service maintains a list of available providers. It activates and deactivates providers depending on their availability and performance.

The RUN service has a SAWSDL interface that receives Resource Description Framework (RDF) information from the VM selected by the QAComPS. The RDF information is used to access the selected cloud service and run the user's computational model. The job log is by QABroker and is also uploaded to the user's tenancy.

III. QAComPS Service

This is a novel service that mediates across providers' various VMs. The service contains a logical model that integrates three areas of information: general information on the VM and cloud providers; details of the computational resources the VMs offer and information about their QoS. The model contains four QoS parameters: cost; reliability; reputation and security (each on a scale from one to nine). The precise meaning of a QoS rating is a business decision.

The cost element of the QoS model is derived from a separate Cost model. This maps cost against the computational resources purchased for example RAM (GB), virtual core (integer), Disk space (GB), price per hour, communication and storage services. The cost model can be expressed as:

$$Cost = (\alpha \ x \ C_l \ + \beta \ x \ Dt)/ \ (\alpha + \beta)$$

where C_l is made up of memory (GB), processor (virtual cores), and non-persistent storage. D_t is the data transfer rate and α, β are constants chosen by the broker. The computational resource size is in four bands: Small, Medium, Large and Very Large. Table 1 shows the actual cost values used in our evaluation study. Here the model does not include long term storage costs that may also be important.

Table 1: Broker Cost Model RUN Service

	Cost/ hour	Memory GB	#Cores	Storage GB
Small	0.06	2.00	1	160
Medium	0.13	4.00	4	500
Large	0.22	8.00	8	800
Very Large	0.45	16.00	16	1700

The costs are based on an exchange rate of 0.6 £/$ and it assumes a Linux operating system.

Three cloud providers were considered: Amazon, Rackspace and FlexiScale as a representative sample of computational cloud services. These providers were used to specify a logical cost model and associated resources (see Table 1). This was used by the MatchMaker that forms part of the selection process. This is discussed in the next section.

The cost model was also used to define the cost QoS rating. In our evaluation study we used three cost bands: Low (1 – 4) equates to high cost VMs (the most expensive for the computational resources they provide); medium (5 – 7) represents intermediate value for money and high (8 – 9) represents excellent value for money (the lowest cost). Cost bands are the main linkage between the actual costs (what the user pays) and the Cost QoS ratings.

Amazon is a cloud provider that explicitly offers HPC VMs. These are clustered and support applications using MPI (the message passing interface) and running very computationally intense jobs. As these services are ideal for running HPC applications the broker RUN service can be restricted to considering only computational clouds that offer this type of service.

a. Architecture

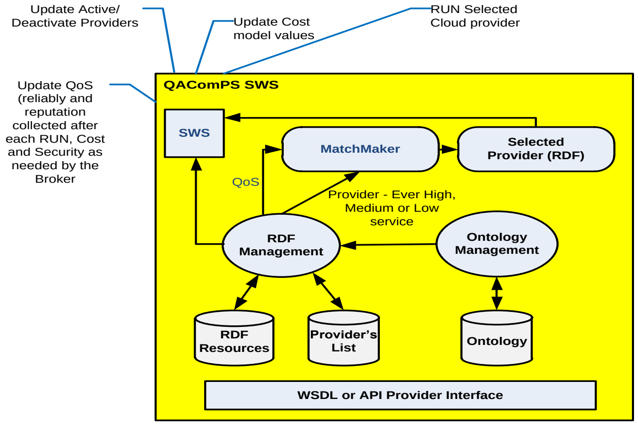

Update Active/ Deactivate Providers
Update Cost model values
RUN Selected Cloud provider
Update QoS (reliably and reputation collected after each RUN, Cost and Security as needed by the Broker
QAComPS SWS
SWS
MatchMaker
Selected Provider (RDF)
QoS
Provider - Ever High, Medium or Low service
RDF Management
Ontology Management
RDF Resources
Provider's List
Ontology
WSDL or API Provider Interface

Figure 3 QAComPS Architecture

The QAComPS architecture has three main functions: (1) The management of the selected providers; (2) The matchmaker; and (3) The selection of the best cloud provider. The service ontology federates across the active cloud providers and logically filters the providers' VMs into High, Medium or Low quality services. This filtration improves the performance of the selection process.

The first stage of the selection process is to apply the widely used Euclidean distance algorithm [4] to rank the active provider's VMs based on the four QoS ratings. The second phase is to take the top *five* ranked VMs and use the *Analytic Hierarchy Process (AHP)* [11]. It is a widely used MCDA (Multi Criteria Decision Analysis) based method that uses a hierarchical approach to decision making. The AHP process matches the VMs against the user requirements using the QoS levels (High, Medium and Low) together with weights. A weight represents the importance of each QoS parameter for the particular job. For example at certain times the cost parameter may be much more important to the user than the parameter 'provider reliability'. QoS levels and weights express the relative priority of each QoS parameter. The performance evaluation results are given below along with the justification for our approach (see §4).

b. Provider's Ontology

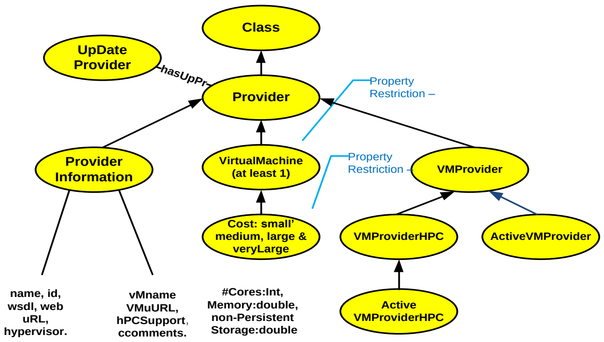

Class
UpDate Provider
hasUpPr
Provider
Property Restriction –
Provider Information
VirtualMachine (at least 1)
Property Restriction –
VMProvider
Cost: small' medium, large & veryLarge
VMProviderHPC
ActiveVMProvider
name, id, wsdl, web uRL, hypervisor.
vMname VMuURL, hPCSupport, ccomments.
#Cores:Int, Memory:double, non-Persistent Storage:double
Active VMProviderHPC

Figure 4 Provider's ontology

The Provider's ontology is shown in Figure 4. The top class is the Provider and a sub-class models the number of Virtual Machines (VM) offered by the provider. An OWL2 property restriction is used to ensure that there is at least one VM. As discussed above the cost model provides four levels varied by VM size. An information class is provided that handles information associated with a provider. The VMProvider class models the type of VM and whether it's active or inactive. A UpDateProvider class enables the information on the selected Provider's VM to be transferred to RUN service using SAWSDL annotations.

c. QoS ontology

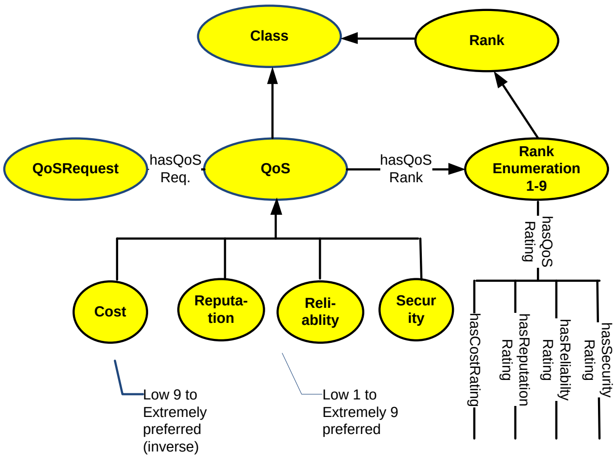


Figure 5 QoS Ontology

The QoS ontology is based on the QoS model as discussed above. It uses an OWL2 Rank Enumeration giving the meaning to the ranking. The data properties enable the user to access each QoS data property. A QoSRequest class enables communication of information between the QAComPS and the broker information service using SAWSDL annotations.

d. Filter Ontology

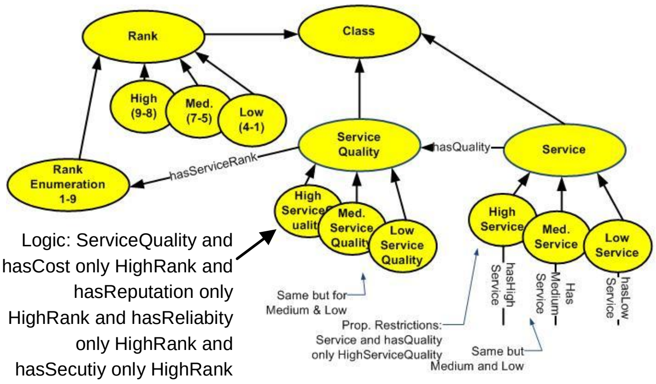


Figure 6 Filter Ontology

The purpose of the filter ontology is to formally classify the providers' VMs into three groups. These are Classes named High, Medium and Low Service. They are used by the MatchMaker. Our evaluation results in §4 show the effectiveness of the filter and this simple classification for the selection of the best VM. The ontology is shown in figure 6. The two main Classes are ServiceQuality and Service. There is an object property linking them. The QoS ratings (1-9) are again used for each QoS metric. The meaning given to the three service quality levels are: High (8-9), Medium (7-5) and Low (4-1). The meaning of each level is assigned by using the OWL2 union property restriction. The logical expression to do this is:

$$UQoS = C \& R_p \& R_l \& S$$

where C, R_p , R_l & S are given in §2.1. The meaning assigned to each service quality class is as follows: HighServiceQuality is UQoS ≥ 8; MediumServiceQuality is UQoS ≥ 5 and < 8; and LowServiceQuality = UQoS < 5. The semantic Web statements are shown in figure 6.

Logic for filtering providers for high QoS is given below:

Logic: ServiceQuality and hasCost only HighRank and hasReputation only HighRank and hasReliabity only HighRank and hasSecutiy only HighRank.

The logic for medium and low can be given in the same manner.

IV. Evaluation

The evaluation identified whether the QAComPS selected the "best" provider while fulfilling all the QoS constraints. It also pointed to any performance lags and other issues with the proposed solution. The proposed solution evaluation was mainly for the selection process against AHP and Quality Matchmaking Process (QMP) [9] that used a combination of Euclidean Distance and AHP algorithms. QAComPS was implemented as a service. In order to undertake this experiment AHP and QMP services were needed to be created. These services were created and tested before undertaking the evaluation. This section is subdivided into a description of the experiment and its results.

a. Experiment

The experiment consisted of the development of two services: QAComPS (responsible for processing user requests) and QABroker (used for updating the provider information).

QABroker provided a light weight service used for creating and updating provider information. Inputs to the service were provider and VM information shown as data properties in Figure 4. This information was processed by creating a model of the ontology with the inputted data values. This was than inferred and reasoned for any errors. If there were no errors than the information was added to a newly created RDF record and an entry was made in the index file. For this experiment public providers such as Amazon were considered but were not used as it would have been too expensive. The results from this research will provide assurance before investing further with this research. Instead twenty five simulated providers were created each offered a different set of QoS metrics while all of them offered the same computational resources. The simulated providers offered small, medium, large VMs while public providers also offered extra large VMs that were not part of this experiment. The resource information associated with small, medium or large came from public cloud providers and is shown in Table 1. QAComPS as described in section 2.3 that used SAWSDL to communicate with other services. SAWSDL annotations included a model reference, and lifting and lowering schema mapping data. The model references represented entities that form part of the ontology while the lifting and lowering schema mapping formed the communication channel between QAComPS and other web services. The lifting schema mapping was used for transferring data from a non-semantic source, such as XML to QAComPS. For the lowering schema mapping QAComPS used a SPARQL query to extract information from RDF and pass it to a non-semantic web service.

QAComPS MatchMaker consisted of a ranking and selection step. It started off by receiving a user request that consisted of resource and QoS requirements. The resource requirements included memory, storage and CPU requirements. The QoS requirements were low, medium and high. The request was passed on to the Euclidean Distance based ranker that ranked the list of available providers. The top five providers were passed on to the AHP-based matchmaker that selected the "best" provider. At the top of the AHP hierarchy the goal was setup to identify the best provider. This was followed by the QoS criteria parameters and their associated weights. These were inputted by the user to reflect their relative priorities. At the bottom of the hierarchy there were alternatives that represented available VM options.

b. Results

The experiment was performed by creating twenty five simulated providers and twenty four user requests. There were eight user requests each for low, medium and high QoS. They were controlled requests whose output was previously calculated beforehand to identify the progress of each service. Each user request was passed to AHP, QMP, QAComPS and QAComPS (with filter).

Figures 5, 6 and 7 showed the results. The horizontal axis shows the user request whilst the vertical axis shows the logical cost. The logical cost model is given in (section III). The logical cost is measured on a scale of one to nine (with nine showing the best option).

The selection was dependent on the user requests and provider QoS. While providers with higher QoS service and lower costs were selected more than once there were other providers which were not selected at all as they offered higher costs and lower QoS. The results shown in figures 5, 6 and 7 show the average cost of the selected providers. At the start of the experiment user requests for high QoS were made and then for medium and lower QoS.

Figure 5 shows the results for high QoS. This means that each of the QoS rating (Cost, Reliability, Reputation and Security), for the selected providers, were high. It can be observed that AHP, QAComPS with and without the filter produced good results. However QAComPS had some inconsistencies and QMP was not very effective.

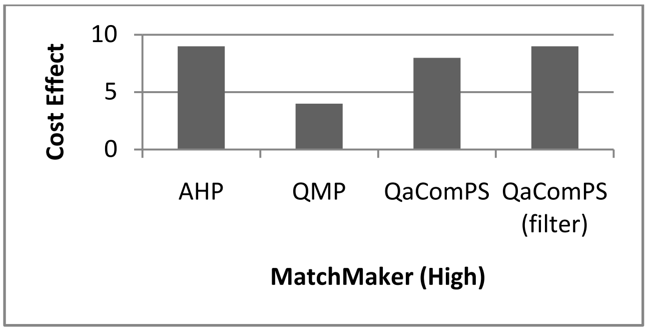


Figure 5 MatchMaker Comparisons (High QoS)

Figure 6 shows the results for medium QoS where the selected providers had a medium level of QoS for all four ratings. It can be observed that QAComPS and QAComPS (with filter) were very effective.

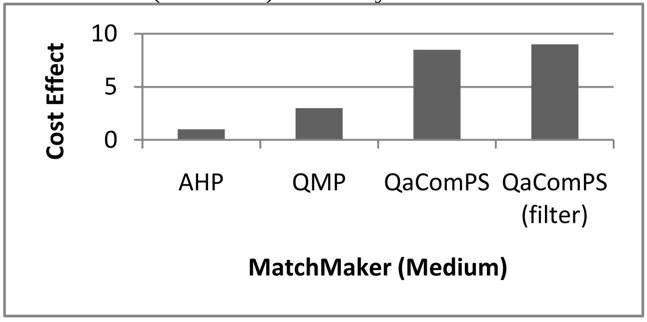


Figure 6 MatchMaker Comparisons (Medium QoS)

Figure 7 shows the results for low QoS, for the selected providers, all have low QoS ratings. It can be observed that QMP was very effective with low QoS while QAComPS (with filter) was consistent for all the user requests.

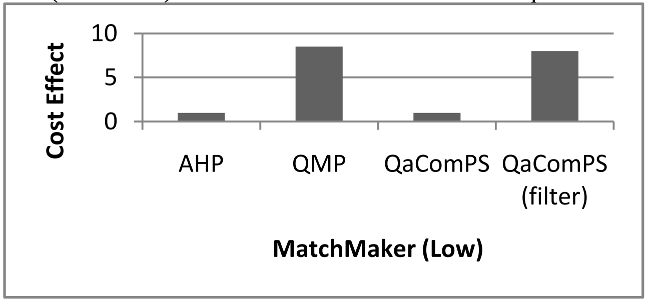


Figure 7 MatchMaker Comparisons (Low QoS)Overall AHP performed well for user requests for higher QoS and did not perform well for medium and low QoS. This may be due to the way AHP selects the best provider as it prioritize high. It behaved in a different way to a fixed set of simulated providers and prioritized selecting the same provider more than once.

Overall the AHP performed well for higher level of QoS while it did not perform well for others. AHP was effective for major changes to user requests however and it was less effective to smaller changes. This may be in part due to the way the algorithm worked as we were using the same data set for every request while it is still being investigated. QAComPS without the filter works well for high and medium with some discrepancies but was not very effective for low while the introduction of filter resolves produces effective results for any QoS. There were no performance issues as the processing time for the twenty five user requests was always under two seconds.

Overall QMP performed well for low but not so well for medium and high. This is due to the way QMP operates as it may prioritize for providers offering lower QoS
Overall QaCompS performed well for high and medium QoS while for low it was not good. This was due to the way QaComPS operates as it always prioritises requests for higher QoS and prefers selecting providers with higher QoS.

Overall the results showed that the QAComPS (with filter) performs best for all the three levels of QoS. Another advantage of the filter was that the performance was enhanced as it reduces the set of relevant providers which results in less processing.

V. Related Research

For the background Semantic Web Ontology the reader is referred to [12]. There are two approaches to creating semantic web services. One is a top down approach using OWL-S the other is a bottom up solution using either SAWSDL [13] or WSMO lite [14]. SAWSDL annotations were used in this paper. This avoids the complexity of using OWL-S [15] while retaining the benefits of a semantic user interface.

There are a number of papers on semantic matchmaking arising from e-commence. The most notable is [16] that presents a prototype matchmaking service using description logic and DAML-S (this was a forerunner to OWL-S). However for performance concerns a non-semantic matchmaker was used in this paper.

There are also papers on QoS Web services that use ontologies. In [17] they provide a novel, rich and extensible ontology for the selection of the requested QoS. Also see [18, 19]. However these ontologies were too comprehensive for this research.

The Grid community [20] has been actively involved in QoS Web services federated by a broker. This brokerage involves managing the negotiation between the service provider and the service consumer and recording any service level agreement (SLA) that is reached. Such brokers often use the standard on-line WS-Agreement SLA document [21] for enforcement (including QoS guarantees). For example, see the ASSESS project that considers risks to fulfilling the SLA (e.g. because of reliability failures) [22]. Buyya and Ranjan discuss a federated resource manager for both the grid and cloud providers [23]. Rochwerger et.al. [24] argue that cloud providers have only recently begun to address the requirements of enterprise solutions, such as support for infrastructure service-level agreements. Their Reservoir project aims to enable providers of cloud infrastructure to dynamically partner with each other to create a seemingly infinite pool of IT resources. A new EU cloud project called OPTIMIS [25] is also aimed at enterprise cloud computing. Its goal is to enable organizations to automatically externalize services to trustworthy and auditable cloud providers in a hybrid cloud model. WS-Agreement are being used in this project.

VI. Conclusion

This paper has presented a novel quality-aware federated computational semantic Web service (QAComPS). This service enabled automatic selection of cloud providers. QaComPS is the key service for the envisioned cloud broker. The evaluation results have potentially shown that QaComPS service can successfully select the best computational cloud provider that meets the user's resource and QoS requirements. The paper has also shown the benefits of using semantic annotations to communicate with external services.

For the future this research needs to ground the simulated results to actual computational cloud providers. The next stage is to fully integrate QaComPS service into to QaBroker service.

QaComPS selects the best cloud provider and passes the RDF file of the selected provider onto the cloud run service. This service invokes the required VM and deploys a VM image on the selected VM. Public providers such as Amazon offer a public DNS key which is used to access VMs remotely. The run service shares this key with the user which grants the user access to the VM. The user can now run his code on the VM and can also install new software. Once the user has completed the job he notifies QaBroker which invokes the cloud run service to stop VM.

This research also has the potential for helping brokers provide a service within a specified timeframe. This service assumes that a user or broker has inserted a number of checkpoints. This would enable the performance of the selected VM to be monitored during a job run. These jobs typically take many hours it would be possible to switch to another VM if the job run fell behind schedule. This would help the broker fulfill service level agreements irrespective of difficulties encountered with a particular VM. This moves us closer to providing a guarant**e**ed service. This assumes that the modeling process is particularly regular to predict the end performance.

VII. Acknowledgments

The authors are extremely grateful to Dr. Benadon Bennett for writing the Filter ontology. Thanks to Professor Peter Jimmack for posing the computational modelling scenario. Thanks also to Dr K Djemame his support for this research.

VIII. References

[1] Armbrust, M., Fox, A., Griffith, R., and Joseph, A.D.: 'A view of cloud computing', Communication of the ACM, 2010, 2010, 53, (4), pp. 50-58.

[2] Amazon Web Services. http://aws.amazon.com/ec2 (09/03/2011).

[3] Rack Space Hosting. www.rackspace.co.uk/cloud-hostin (09/03/2011).

[4] FlexiScale Public Cloud. http://flexiscale.com (09/03/2011).

[5] He, Q., Zhou, S., Kobler, B., Duffy, D., and McGlynn, T.: 'Case Study for Running HPC Applications in Public Clouds'2009.

[6] Evangelinos, C., and Hill, C.N.: 'Cloud Computing for parallel Scientific HPC Applications: Feasibility of running Coupled Atmosphere-Ocean Climate Models on Amazon's EC2'. Proc. CCA-08, Chicago 2008.

[7] Menasce, D.: 'Composing Web Services: A QoS Veiw', IEEE Internet Computing, 2004

[8] Menasce, D.: 'QoS Issues in Web Services', IEEE Internet Computing, 2002.

[9] Eleyan, A., Mikhailov, L., and ., L.Z.: 'Quality-of-Services in Web Services Architecture', Ingnierie des Systmes d'Information, Special Issue on Information Systems Quality, 2004, 9, (5-6), pp. 185-203.

[10] Tran, V., Tsuji, H., and Masuda, R.: 'A new QoS ontology and its QoS-based ranking algorithm for Web services'. Proc. Simulation Modelling Practice and Theory, 2009.

[11] Saaty, T.L.: 'How to make a decision: the analytic hierarchy process', European Journal of Operational Resea rch, 1990, 48, (1), pp. 9-26.

[12] Allemang, D., and Hendler, J.: 'Semantic Web for the Working Ontologist' (Morgan Kaufmann, 2008.

[13] Kopecky, J., Vitvar, T., Bournez, C., and Farrell, J.: 'SAWSDL: Semantic annotations for WSDL and XML schema', IEEE Internet Computing, 2007, 11, (6),

[14] Brooke, J., Fellows, D., Garwood, K., and Goble, C.: 'Semantic matching of Grid resource descriptions', in Editor 'Book Semantic matching of Grid resource descriptions' (Springer-Verlag, 2004,), pp.60-67.

[15] Vitvar, T., Kopecký, J., Viskov, J., and Fensel, D.: 'Wsmo-lite annotations for web services'. Proc. Proceeding ESWC'08 Proceedings of the 5th European semantic web conference, 2008

[16] Martin, D., Burstein, M., Mcdermott, D., and Mcilraith, S.: 'Bringing semantics to web services with owl-s'. Proc. World Wide Web 2007.

[17] Li, I., and I Horrocks: 'A software framework for matchmaking based on semantic web technology - International Journal of Electronic Commerce, Proc. WWW2003, May 20-24 2003.

[18] Wang, F., Vitvar, T., Kerrigan, M., and Toma, I.: 'A QoS-aware selection model for semantic web services', IEEE Transaction Software Engineer, 2006, 30, (5), pp. 311-327.

[19] Kritikos, K., and Plexousakis, D.: 'Semantic QoS metric matching'. Proc. European Conference on Web Services (ECOWS'06) 2006.

[20] Ian Foster, Carl Kesselman, and Steven Tuecke "The Anatomy of the Grid: Enabling Scalable Virtual Organizations" International Journal of High Performance Computing Applications Fall 2001 15

[21] Andrieux, A., Czajkowski, K., Dan, A., K. Keahey, Ludwig, H., Pruyne, J., Tuecke, S., and XU, M.: ' Web services agreement specification (WS-Agreement)', in Editor 'Book Web services agreement specification (WS-Agreement)' (Gridforum.org), 2006.

[22] K. Djemame,J. Padgett, I. Gourlay, and D. Armstrong. 'Brokering of Risk-Aware Service Level Agreements in Grids'. Concurrency and Computation: Practice and Experience, Vol. 23, No. 7, May 2011

[23] Buyya, R., and Ranjan, R.: 'Special section: Federated resource management in grid and cloud computing systems', Future Generation Computer Systems, Elsevier B.V., 2010.

[24] Rochwerger, B.B., D. Levy, E. et al: 'The Reservoir model and architecture for open federated cloud computing', IBM Journal of Research and Development, 2010, 53, (4), pp 1-11.

[25] A.Juan Ferrer, F.Hernandez, J.Tordsson, E.Elmroth, R.M.Badia, K.Djemame 'OPTIMIS: a Holistic Approach to Cloud Service Provisioning'. In Proceedings of the 1st International Conference on Utility and Cloud Computing (UCC 2010), Chennai, India, Dec 2010.

UML MODELING OF SEMANTIC INFORMATION SYSTEM NETWORK

Evan Tsai, Gabriel Nunéz, Aleksander Milshteyn, Jorge Estrada, Adrienne Lam,
Sergio Mendoza, Neil Arellano, John Paul Adigwu, Sergio Beltran, Helen Boussalis, Charles Liu

Structures, Propulsion and Control Engineering (SPACE) Center
College of Engineering
California State University, Los Angeles
5151 State University Drive
Los Angeles, CA 90032 USA
1-323-343-5445
etsai@calstatela.edu

Abstract - *This paper focuses on the modeling of the Semantic Information System (SIS) Network with Ubiquitous Video Conferencing (UVC) multicasting applications in the Unified Modeling Language (UML) format. The UML diagrams present detailed visualizations and the representation of the SIS Network's architecture, component functionality and interdependency. The development of UVC multicasting applications is geared towards users who are in constant need of real-time interactive collaboration. It is best suited for educators, researchers, and team-project members – all of which can be referred to as SIS Network participants. SIS Network participants will objectize contents of their work to their respective Semantic Networks, which will generate a hierarchical tree structure to interrelate those contents based on their semantic meaning and relationship to each other. These user-generated contents can be accessed, updated, and shared with other Network participants.*

Keywords: Unified Modeling Language (UML), Ubiquitous Video Conferencing (UVC), Semantic Information System network, decentralized server, client-server model

1 Introduction

The Advanced Computation and Communication (ACC) team of the NASA-CSULA SPACE Center is focused on design and development of new tools for information dissemination for collaborative education and research[11]. The SPACE Center consists of faculty-led graduate and undergraduate students which are formed into specific teams based on particular areas of research.

The current project objective is to design and implement an integrated framework for ubiquitous computing based on the Tuple Space paradigm [1]. The framework itself consists of two major components: Ubiquitous Video Conferencing which focuses on seamless communication between users across different platforms, while the SIS Network supports dissemination of information to provide a collaborative work environment.

Both components have a wide range of uses, which vary by its audience. The SIS Network is intended for targeting communities with similar interests, whether that community is in industry, education, or recreation. Combined with Ubiquitous Video Conferencing, the SIS framework is designed to be flexible and powerful for a wide range of uses accommodating a broader range of audiences.

UML (Unified Modeling Language) is geared for object-oriented analysis and design. It is used to model, document, and visualize every individual elements of an object-oriented system by displaying the results in different types of detailed diagrams which are different representations or aspect views of the project. UML is used in the industry by system engineers to visually model software-intensive and/or large and complex systems thus formalizing the organization of the project architecture.

The SIS Network is in its initial UML modeling stage. This visual organization is not only beneficial to the engineers but to the project management and newly hired workers as it accelerates the understanding of the inner workings of the project as well as a roadmap to the desired goal. One of the main features of UML is the ability to tie together and model how different engineering aspects interact with one another within the system such as an electromechanical implementation as well as being able to identify, distribute, and organize the workload between teams by the use and placement of the diagrams.

[1] Acknowledgement to NASA University Research Center Program, Grant # NNX08BA44A

The main tools used for this particular project are described as follows: Qt is an open-source software framework with project vital models of Phonon and QtMultimedia, FFMPEG encoder/decoder and OpenCV library for image processing. Qt is also an Integrated Development Environment similar to Microsoft Visual Studio; it is available as an open-source software and able to generate programs that are cross-platform compatible such as being able to run on different operating systems and different classes of computing hardware such as embedded systems and capable Smartphone devices. Phonon is a Qt module that is used for high-level multimedia manipulation; it is used specifically for playback of media files. QtMultimedia is another Qt module that is used for low-level multimedia manipulation; in our case the capture and streaming of real time audio. FFMPEG is an open-source multimedia software that is mainly used in the project to encode/decode audio and video streams.

2 SIS Network Architecture

A Tuple Space parallel programming paradigm was originally applied to an Aerospace Information Server (AIS), which uses a single-server-multiple-worker model to develop high-performance task management schemes and support parallel processing. This technology is the foundation for our current work and is applied to a distributed server scheme to increase the capabilities of the framework. The role of Tuple Space and other components in the framework is generalized in Figure 1. In the figure below, clients (and their compatible hardware) are situated and interfaced with the various applications on the bottom level. These applications connect and interact in the background with their respective servers that are built upon Tuple Space. Lastly, the servers themselves are synchronized with one another via Active Directory.

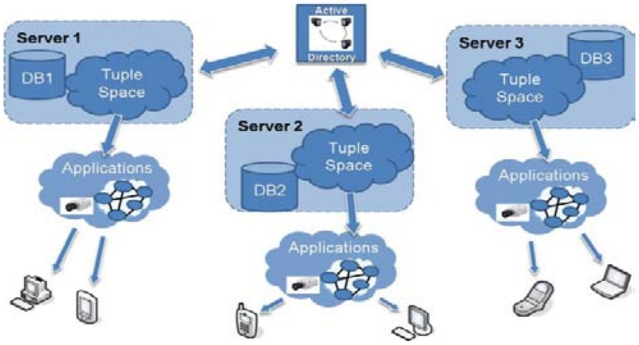


Figure 1: Overview of distributed server scheme and applications

The process model in Figure 2 highlights key areas of work that is necessary to create an integrated and fully functional platform. Only then can real-time network changes and multi-user interactions be supported. The Client-Tuple Space layer handles requests for application events and user authentication protocols. The Tuple Space-Active Directory layer ensures data synchronization across all servers in the distributed network. These include the Tuple Space-Fileserver layer and the Tuple Space-Active Directory Layer.

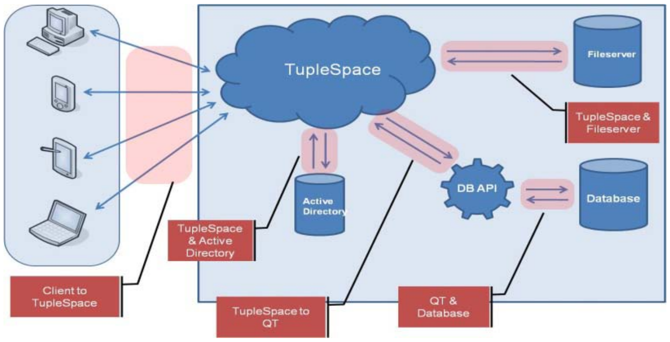


Figure 2: Application process diagram

The diagram shown in Figure 3 is a top-level view of the SIS Network using the Component model of a UML Diagram. The terms <<subsystem>> and <<component>> are parts of the UML standard called stereotypes, which is a variation of an existing system but in a different context. In the figure, each component is a software module for performing the functions of the SIS.

This model highlights the interdependencies between each component by its connections. The combined 'bubble-and-socket' symbol/notation in the figure represents a type of interface in UML. Each 'bubble' connection indicates that the component is providing a service while its associated 'socket' bracket indicates the component requires a service to properly function. In our applications, such interfaces are implemented in software. The small squares in the figure indicate port connections between the client software and server. In our design, such ports indicate physical connections to other physical components.

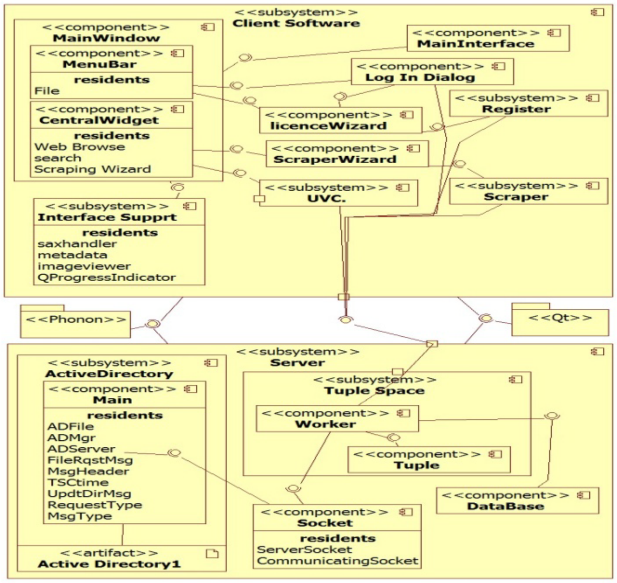


Figure 3: SIS Network Component Diagram

Complimentary to the Figure 3, Figure 4 presents the top-level behavioral view of the system. This model is a Use Case UML diagram which demonstrates the use of the software from the point of view of a user. As seen from in this diagram, the first feature observable to the user is the Main Window interface, which then connects to every other feature in the system such as login and registration, web browsing, the Scraping Tool, and the UVC.

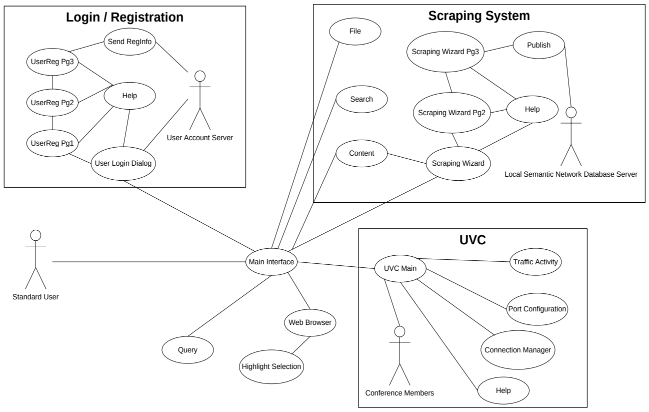
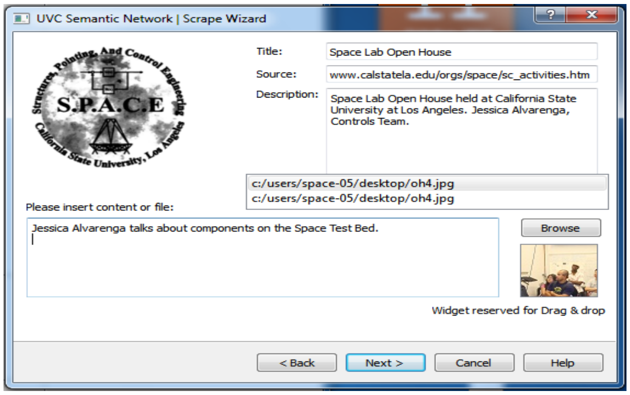


Figure 4: SIS Network Use Case Diagram

2.1 Model of SIS Scraping Tool

The Semantic Network development has focused on the incorporation of the Scraping Tool. The Scraping Tool provides users of the Semantic Network the ability to add new objects into the hierarchical tree structure by featuring drag and drop capability of object content, as shown below in Figure 5. Within this context, object contents can include text, image, video and/or audio files.


UVC Semantic Network | Scrape Wizard

Title: Space Lab Open House

Source: www.calstatela.edu/orgs/space/sc_activities.htm

Description: Space Lab Open House held at California State University at Los Angeles. Jessica Alvarenga, Controls Team.

c:/users/space-05/desktop/oh4.jpg
c:/users/space-05/desktop/oh4.jpg

Please insert content or file:

Jessica Alvarenga talks about components on the Space Test Bed.

Figure 5: Scraping Wizard, Content Input

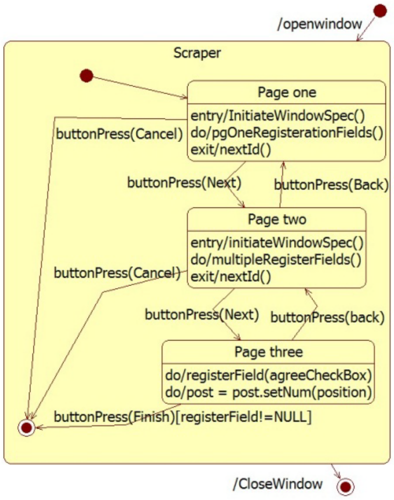

/openwindow
Scraper
Page one
entry/InitiateWindowSpec()
do/pgOneRegisterationFields()
exit/nextId()
buttonPress(Cancel)
buttonPress(Next)
buttonPress(Back)
Page two
entry/initiateWindowSpec()
do/multipleRegisterFields()
exit/nextId()
buttonPress(Cancel)
buttonPress(Next)
buttonPress(back)
Page three
do/registerField(agreeCheckBox)
do/post = post.setNum(position)
buttonPress(Finish)[registerField!=NULL]
/CloseWindow

Figure 6: State Chart Diagram of the Scraping Tool (Top Level)

The Activity chart diagram in Figure 6 is a slightly more detailed behavioral model of a system, which describes the state of the system at any point and the transitions made to change its current state. This diagram is powered by 'states', which are tasks that are done sequentially. For example, the solid circle on the outside of the box denotes the start of a specific sub-system. The smaller solid circle with multiple inputs in the middle of the diagram represents a junction node where it can fork out to single connection based on input conditions, which are known in UML as 'guards' or denoted as square brackets.

Each state can contain action primitives inside such as: 'entry/', 'do/', and 'exit/' in an Activity Chart UML diagram. An 'entry/' action implies that a section of code will only be executed once when it enters the state (e.g. such as initialization routines). A 'do/' action shows that this section of code is to be looped continuously. Lastly, the 'exit/' action is another section of code that will only execute once just before it leaves the state. For our example of the Scraping Tool, the system revolves around four basic states. Each state is dictated by such actions as the click of a button to continue or cancel the publishing process, as well as requesting help for the current task.

For example, the 'Page 2' state in Figure 6 contains the bulk of the information necessary to publish new information into the SIS Network. An important feature to note at this point is the use of sub-state systems and the advanced features of drag and drop, which facilitates the process.

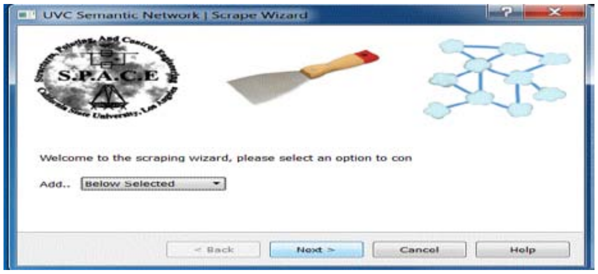
The Scraping Tool is the means to populate the Semantic Network [3]. This component facilitates scraping of information, objectized tagging, and the generation of metadata. The latter two are important for increased relevancy when searching for information within the Network.

The hierarchical tree structure is comprised of the QTreeWidget class, which provides a predefined tree view model [4]. Object IDs are assigned from the PostgreSQL Database upon execution of a SQL insert statement. This returned object ID is used to name the object directory of the newly published content. Similarly, the metadata file created is also saved as ‘objectID.xml’[5].

The Scraping Tool is implemented using several classes within the Qt Nokia cross-platform environment. QWizard provides standard core functions associated with the development of ‘form’ design. These include field, pages, error-checking subclass functions, enabling the creation of a fully functioning registration tutorial/wizard.

The ‘Scraping Wizard’ sequence is displayed (Figure 7) when the ‘add new object’ button is triggered in the main application window. The sequence is such that a series of pages are displayed in specific order to provide proper navigation through the object creation process. Unlike many other classes within Qt, QWizard manages the allocation and temporary storage of user input within the forms. After the Wizard process is completed, the registered QWizard fields are available outside of the QWizard class for further information processing.

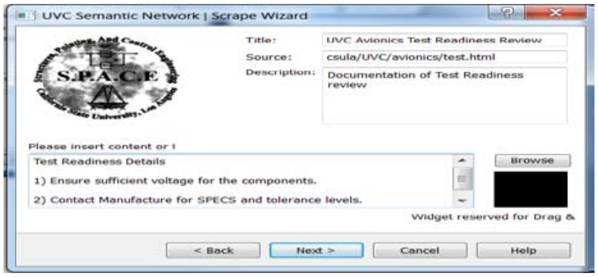
The first layer of the Scraping Wizard gathers object positioning information for new data to be placed in the tree structure. The relative positioning tracking is made available via QTreeWidget, which inherits several core classes from QTreeView. To keep track of object positioning, QTreeWidget::currentItem() is selected as a pointer to QTreeWidgetItem, an item within the tree structure. Depending on the position selected, a function call to QWizard::ItemAbove, QWizard::currentItem, QWizard::ItemBelow, and QWizard::insertTopLevelItem allow for the insertion of new objects into the desired position within the tree structure.



Welcome to the scraping wizard, please select an option to con

Add.. Below Selected

Figure 7a: Scraping Wizard Interface – Object Placement



Title: UVC Avionics Test Readiness Review

Source: csula/UVC/avionics/test.html

Description: Documentation of Test Readiness review

Please insert content or I

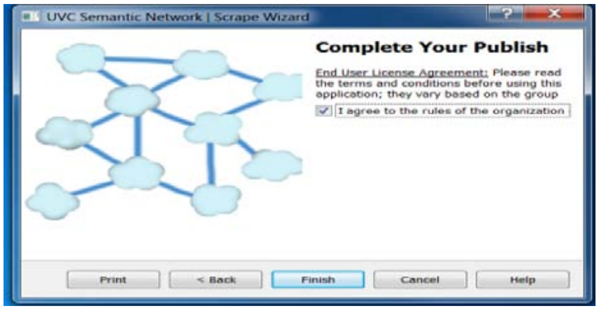
Test Readiness Details

1) Ensure sufficient voltage for the components.

2) Contact Manufacture for SPECS and tolerance levels.

Widget reserved for Drag &

Figure 7b: Scraping Wizard Interface – Content Insertion



Complete Your Publish

End User License Agreement: Please read the terms and conditions before using this application; they vary based on the group

☑ I agree to the rules of the organization

Figure 7c: Scraping Wizard Interface – Publishing Stage

The Scraping Wizard can detect if the network has an Internet connection. In the event of connection error, the addition of new information is still possible locally. Moreover, the newly created information is uploaded to the network when an Internet connection becomes available. This scheme is accomplished through a simple timestamp naming convention to object directory folders.

2.2 Model of SIS Ubiquitous Video Conferencing

The Ubiquitous Video Conferencing application is utilized over the SIS Network. The primary responsibility of the UVC is to provide SIS participants with flexible forms of communication such as visual, audio, text-messaging, and data transmission. Video and Audio transmissions are handled through UDP channels, while text and data are transmitted through TCP/IP.

Figure 8 displays a State Chart Diagram for one of the many features of the Ubiquitous Video Conferencing. When the SIS network participant clicks on the UVC tab in the Main User Interface, it goes into an idle state, doing its first initialization while continuously refreshing the GUI and checking for hardware changes in the user’s audio and/or video input sources. At that idle state, the user can either start to transmit its audio/video and broadcast its existence within the SIS network or it can begin receiving other users’ audio/video from a specified port.

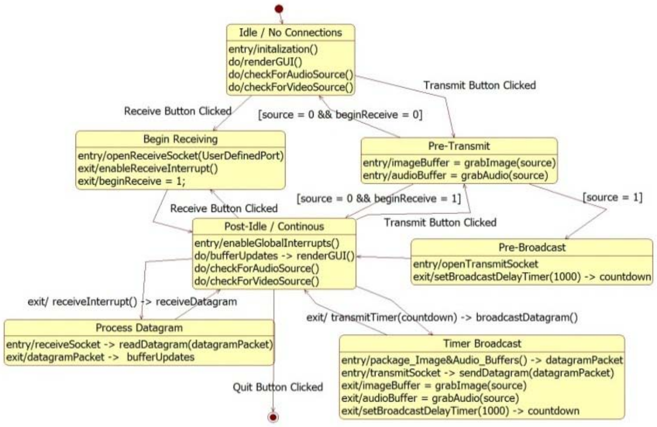

Idle / No Connections
entry/initalization()
do/renderGUI()
do/checkForAudioSource()
do/checkForVideoSource()
Transmit Button Clicked
Receive Button Clicked
[source = 0 && beginReceive = 0]
Begin Receiving
entry/openReceiveSocket(UserDefinedPort)
exit/enableReceiveInterrupt()
exit/beginReceive = 1;
Pre-Transmit
entry/imageBuffer = grabImage(source)
entry/audioBuffer = grabAudio(source)
[source = 0 && beginReceive = 1]
[source = 1]
Receive Button Clicked
Transmit Button Clicked
Post-Idle / Continous
entry/enableGlobalInterrupts()
do/bufferUpdates -> renderGUI()
do/checkForAudioSource()
do/checkForVideoSource()
Pre-Broadcast
entry/openTransmitSocket
exit/setBroadcastDelayTimer(1000) -> countdown
exit/ receiveInterrupt() -> receiveDatagram
exit/ transmitTimer(countdown) -> broadcastDatagram()
Process Datagram
entry/receiveSocket -> readDatagram(datagramPacket)
exit/datagramPacket -> bufferUpdates
Timer Broadcast
entry/package_Image&Audio_Buffers() -> datagramPacket
entry/transmitSocket -> sendDatagram(datagramPacket)
exit/imageBuffer = grabImage(source)
exit/audioBuffer = grabAudio(source)
exit/setBroadcastDelayTimer(1000) -> countdown
Quit Button Clicked

Figure 8: UVC State Chart Diagram

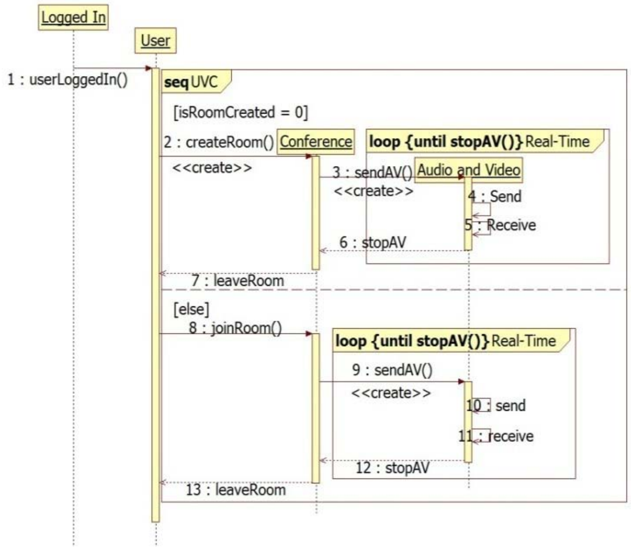


Figure 9: UVC Sequence Diagram

Figure 9 above is a Sequence Diagram that shows another different perspective of a top-level view of the Ubiquitous Video Conferencing, which can be interpreted as the ‘layers’ the program has to go through in order to perform the desired function. In this case, we show the steps the program goes through to create or join an existing Conference Room and chooses to participate by sending real-time audio and video data. The user logs in through the Main Interface, goes to the UVC tab, then if the user wants to create a room, the block statement creates a ‘Conference’ layer; then the user can decide whether to send audio and/or video or just idle along the ‘Conference’ layer. In this diagram, the ‘seq UVC’ frame is called a combined fragment in UML, which is a conditional statement that divides up the program path. For example, [isRoomCreated = 0] and [else] are fragments within ‘seq UVC’ box which perform different functions when being passed into the ‘Conference’ lifeline; either createRoom() or joinRoom().

3 SIS NETWORK DATABASE

PostgreSQL database was selected to perform metadata information storage [6]. Due to the hierarchical structure of objects on the network, a database capable of facilitating keyword searching across “parent” and “child” nodes is the dominant determining factor in appropriate database selection. These features include the ability of PostgreSQL to define data types and fully describe relationships and features, which are not available on traditional databases such as mySQL. In addition, there is a capability for “child” nodes to access the network to inherit “parent” attributes (data types which compose the parent object).

4 Conclusion

Further efforts in regards to Semantic Information System Network implementation will deal with integrating the Database and Tuple Space. Specifically, the addition of hash ID in Tuple Space provides recognition of a SQL defined request from network users and inter-server query tasks. The performance tests in real-time simulation will be conducted in order to provide accurate results on bandwidth and memory consumption. Different scenarios will be simulated in order to optimize SIS Network system.

5 References

[1] S. Balle, D. Palermo, “Enhancing an Open Source Resource Manager with Multi-Core/Multi-threaded Support,” *Hewlett-Packard Company*, 2007.

[2] A. Alegre. “Aerospace Information Server.”
California State University Competition, 2008. Structures, Propulsions, And Control Engineering (SPACE) Center – Research Topic: Ubiquitous Computing and Embedded Architectures
URL:http://www.calstatela.edu/orgs/space/sc_UbiquitousComputing.htm

[3] J. Alvarenga, H. Boussalis, “A Semantic Aerospace Network using Objectized Tags Exchanged through Decentralized Information Servers.” HBCU/OMSI \ Aerospace Collaboration Conference 2009 Cleveland, Ohio, USA, July 2009.

[4] “QT Cross-Platform Application and UI Framework” URL: http://qt.nokia.com/

[5] R. Tolksdorf, F. Liebsch, and D. Minh Nguyen, “XMLSpaces.NET: An Extensible Tuplespace as XML Middleware.”

[6] The PostgreSQL Global Development Group PostgreSQL 9.0.3 Documentation.” URL: http://www.PostgreSQL.org/files/documentation/pdf/9.0/PostgreSQL-9.0-US.pdf

Traversing Documents by Using Semantic Relationships

Bilal Gonen[1]
[1]Computer Science and Engineering Department, University of Nevada, Reno, Reno, Nevada, U.S.A.

Abstract— *Relationships are keys to the semantics and hence the Semantic Web. One interesting way to exploit relationships is to link documents such that terms in the documents are semantically related through well defined relationships. In his 1945 article, Dr. Vannevar Bush posited the idea of creating a device capable of recording all of human knowledge. Dr. Bush suggested that such a device would allow its user to traverse a space of documents by following a "trail of associations" in the user's mind. The assumption however is that links of the Web are meaningful to the user. This assumption certainly does not hold for the eight billion pages on the Web. We consider a scenario where the user has some prior knowledge of the domain. Our approach to building such a system relies on two aspects of Semantic Web: (a) semantic metadata for documents, and (b) populated ontology with relationship instances.*

Keywords: semantic web, semantic browser, data mining

1. Introduction

There have been significant research activities in document categorization. A number of tools have been built that provide users to browse among the classified collection of documents. In order to give the user ability to pick the most relevant documents which are classified under a category, several methods have being used to rank them based on their representative values, i.e., term frequency, inverse document frequency, etc. In most of these tools, the classification of the documents is built as a taxonomy. There is no connection between the different nodes in the taxonomy other than hierarchy-relationships. Consequently such a document organization restricts the users to browse documents under a particular category. To get to a document on a related topic the user has to navigate up the hierarchy and back down to the relevant topic.

For instance, consider a user reading an article about the disease "measles". The user would like to read articles which talk about some drugs that help in curing measles. For instance, there is a "cure" relationship between "egg plant seeds" and "measles". Here is a quotation from the home-remedies web site: "The seeds of the egg plant are a stimulant. Intake of half to one gram of these seeds daily for three days will help develop immunity against measles for one year." [3] Although these two concepts "egg plant seeds" and "measles" are related to each other by an important relationship between them, how likely is it that they may be classified under the same category? Availability of such "related" information depends on whether the author of the text includes such information. Without having the information that a relationship "cures" links the concepts, or having a means to use this information, a document categorization tool would not put documents about these two concepts under the same category, because one of them is a disease, and the other one is a plant. A categorization system arguably should not put such "related" documents in the same category, but the ability to navigate from a document about "measles" to one about a cure is a very useful one to have. The egg plant seeds are not the only things helping in remedy of measles. The same webpage [3] mentions that "application of mud packs" also helps in remedy for measles. Therefore, would not it be nice to suggest the user to read about articles about "egg plant seeds", "mud packs", "barley", "turmeric", and perhaps some others, by also giving the "cures" relationship, thus telling the reason why some articles about these topics are offered? Also, before offering the articles of topics which cures the measles, would not it be better to offer the user the relationships about the measles, first? Perhaps, the user may not be interested in reading topics about curing the measles, but may be interested in reading topics about causes of measles. By choosing this relation, totally different topics would be offered to the user, instead of egg plant seeds, barley, etc. which help cure the measles.

The above discussion demonstrates the need for using some kind of knowledge base which includes the entities along with relationships between them. Here is where the Semantic Web comes in. The Semantic Web has gained much interest during the past few years. Data typically published on the web is human understandable and is meant for human consumption. However, the Semantic Web makes such data machine-understandable. Associating formal semantics with data, and using it in making search, browsing and analysis more intelligent and precise, can also lead to saving time for a user who would otherwise have to peruse more data to get all the information he or she seeks.

When we see a word in the text, for instance "America", human cognitive processes associate meaning with the term. However, a machine does not know that "America" is a country, located in North America, has a border with Canada, etc. As humans, we know that America is (likely or certainly) the USA, which is a country. We need to attach the property or the relationship "country" to the word "America" somehow. Also we know that America, USA, United States, US, etc. all refer to the same thing. Should we also attach such

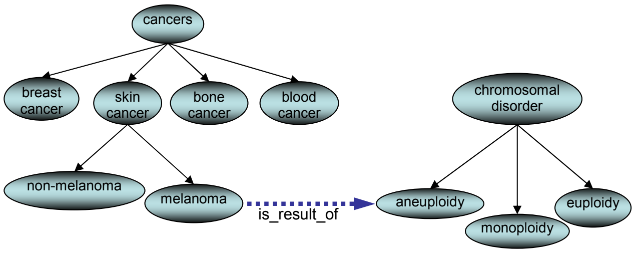


Fig. 1: Relationship in the ontology

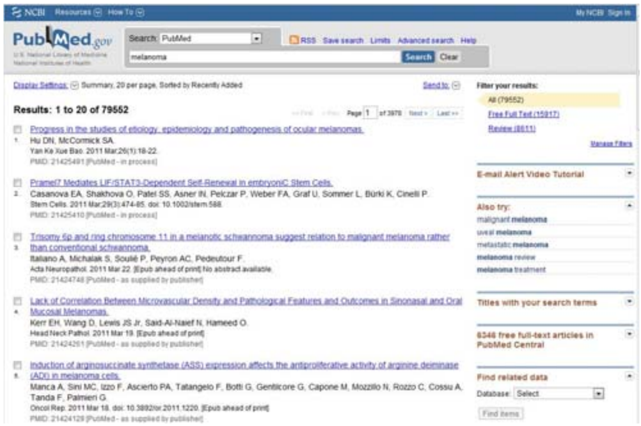
information to the word "America"? How about the concepts that have some relationships with America? Georgia in the relationship of being a state, or George W. Bush, in the relationship of being the President, for example. It would be useful for us to attach this information to the word America in text somehow to have the machine find the relevant information for the user. Instead of attaching all the information we know about a word in the plain text, some external knowledge bases (parts of what is called ontologies) are used in the Semantic Web. In the ontology, we have class-subclass hierarchy, such as apple is subclass of fruit, and fruit is subclass of food, and so on. Taxonomy also has these hierarchical relations. The difference between ontologies and taxonomies is that ontologies have named relationships between the classes, even if they do not have hierarchical relationship. However, taxonomy does not have such relationships besides hierarchical relationships.

Our Semantic Browser tool enables easier navigation with relationships as opposed to hyperlinks. In their paper, Logical Information Modeling of Web-accessible Heterogeneous Digital Assets [5], Amit Sheth, and Kshitij Shah defined hyperlinks as "physical (hard) relationships" and semantic relationships in the Relationship Web as "virtual links".

In our paper, we also show that more relevant documents are returned by using the virtual links as opposed to hyperlinks which do not have semantic relationship between the documents. Processing the documents in our dataset by using our ontology, we built a relationship web in which documents are connected to each other with relationships. The advantage of these virtual connections is that a user can navigate from one document to another using relationships, even if there is no hyperlink between those documents.

Our work is one of the earliest attempts at utilizing the semantics of the relationships to support browsing and navigation of a document space.

In section 2, we describe the concept "Relationship Web". Then, we explain the system architecture in Section 3. In that section, we describe the ontology and dataset used in the project. Section 4 shows the "Semantic Browser" tool [25]. In section 5, we review related work. In section 6, conclusions and future work are explained.



melanoma

Display Settings: Summary, 20 per page, Sorted by Recently Added

Results: 1 to 20 of 79552

Page 1 of 3978

1. Progress in the studies of etiology, epidemiology and pathogenesis of ocular melanomas.
Hu DN, McCormick SA.
Yan Ke Xue Bao. 2011 Mar;26(1):18-22.
PMID: 21425491 [PubMed - in process]

2. Pramel7 Mediates LIF/STAT3-Dependent Self-Renewal in embryoniC Stem Cells.
Casanova EA, Shakhova O, Patel SS, Asner IN, Pelczar P, Weber FA, Graf U, Sommer L, Bürki K, Cinelli P.
Stem Cells. 2011 Mar;29(3):474-85. doi: 10.1002/stem.588.
PMID: 21425410 [PubMed - in process]

3. Trisomy 6p and ring chromosome 11 in a melanotic schwannoma suggest relation to malignant melanoma rather than conventional schwannoma.
Italiano A, Michalak S, Soulié P, Peyron AC, Pedeutour F.
Acta Neuropathol. 2011 Mar 22. [Epub ahead of print] No abstract available.
PMID: 21424748 [PubMed - as supplied by publisher]

4. Lack of Correlation Between Microvascular Density and Pathological Features and Outcomes in Sinonasal and Oral Mucosal Melanomas.
Kerr EH, Wang D, Lewis JS Jr, Said-Al-Naief N, Hameed O.
Head Neck Pathol. 2011 Mar 19. [Epub ahead of print]
PMID: 21424261 [PubMed - as supplied by publisher]

5. Induction of arginosuccinate synthetase (ASS) expression affects the antiproliferative activity of arginine deiminase (ADI) in melanoma cells.
Manca A, Sini MC, Izzo F, Ascierto PA, Tatangelo F, Botti G, Gentilcore G, Capone M, Mozzillo N, Rozzo C, Cossu A, Tanda F, Palmieri G.
Oncol Rep. 2011 Mar 18. doi: 10.3892/or.2011.1220. [Epub ahead of print]
PMID: 21424129 [PubMed - as supplied by publisher]

Filter your results:
All (79552)
Free Full Text (15917)
Review (8611)

Also try:
malignant melanoma
uveal melanoma
metastatic melanoma
melanoma review
melanoma treatment

6346 free full-text articles in PubMed Central

Find related data
Database: Select

Fig. 2: A screenshot from PubMed Website

2. Building Relationship Web

Given a set of documents along with an ontology schema and a set of entity instances, our goal is to create a web of documents linked to each other by named relationships. Any two documents may not be connected to each other by physical links (HREF), but may contain terms which are related to each other based on the ontology used. Without building the Relationship Web, the user does not have a way to navigate semantically connected documents unless there is at least one physical link between those documents. Even with the existence of such a physical link, the relationship between the hyperlinked term and the target page is based on the interpretation of the user. Although the relationship is human understandable, it is not machine understandable.

Some websites have some system to suggest related articles to users. However, the relationships between those related articles are not named. Another issue is that the user may not be interested in reading the articles related to the article she reads at that moment, but may be interested in reading articles related to any particular term in the article she reads.

Let us consider a scenario that a user reads a medical article in the PubMed [6] Web site. She reads a sentence that contains "melanoma". Melanoma is a type of skin cancer. In the page where she reads the article abstract, the web site includes a hyperlink saying "Related Articles". The user may not be interested in the articles related to the main concept of the article she reads at that time, but may be interested in the article related to the particular term "melanoma" that she has just encountered in the article abstract. If the term "melanoma" in the sentence is not hyperlinked, then the user has to type the term "melanoma" in the search box at the top of the page, to read articles about the melanoma. What if she is not interested in articles about the melanoma, but is interested learning about the causes of the "melanoma"? If the user does not know the causes of the melanoma already, there is no way that she can reach to the articles which talk about those causes in the current system. She needs to know the causes beforehand, so that she can type their names in the search box.

Imagine that in order to return related terms to user, a website uses a statistical mechanism. Such statistical mechanisms may bind two terms together as “related” just because they occur in the same sentences frequently. If we consider the recommendation systems on the online stores; they relate some products together, just because they are frequently purchased in the same transactions. According to such statistical systems, the only relationship between those products is statistical proximity. Instead of giving the user related terms only, it is necessary for the user to know the relationships also, because the user may not be interested in spending her time to read articles about “aneuploidy”, unless she knows that aneuploidy is a result of melanoma. By using our approach, however, for the chosen term “melanoma”, the user is offered several relationships, such as “affects”, “co occurs with”, “occurs in” and “is result of”. By choosing the relationship “is result of”, the user can get the terms which are the result of “melanoma”.

As can be seen in the scenario above, no physical link (hyperlink) is needed to navigate to “semantically” related documents. Also, no prior knowledge is needed by the user to know what the results of “melanoma” are.

The critical research issue, however, is to identify which of the prohibitively large number of relationships are more relevant than others. We demonstrate this capability by developing an application that allows users to browse documents by following chains of named relationships much the same way we follow hyperlink today. As a starting point, we test a subset of PubMed [6] abstracts linked to each other by named relationships. The named entities at the instance level are associated with some schema class types at the schema level of the ontology. The instances are usually associated with one, two, or three class types at the schema level. Because we do not yet have relationships between concepts at the instance level, we look at the relationships between the types of those instances. Then we use these relationships to build our relationship web.

3. System Architecture

In this chapter, we explain the ontology and dataset used in the project.

3.1 Ontology Used In The Process

To build the schema layer of the ontology used in our project, we have parsed the UMLS [1] dataset. The UMLS dataset consists of several XML files. Triples (terms and relationships associated between them) were extracted and stored in an RDF file. In the RDF schema file generated, there are 135 classes and 49 relationships.

In order to build the instance layer of the ontology used in our project, we used SAX java parser to parse the MeSH XML file. This XML file is 240 MB in size. It contains 21,945 MeSH terms, but does not contain relationships between these terms. Because MeSH terms are represented as instances in our ontology, we have 21,945 instances in the instance layer of our ontology. From this XML file for each instance, we extracted the UI number which is a unique number for each MeSH term. We extracted the UMLS types of MeSH terms. As mentioned above, these UMLS types are represented as schema classes in the schema layer of our ontology.

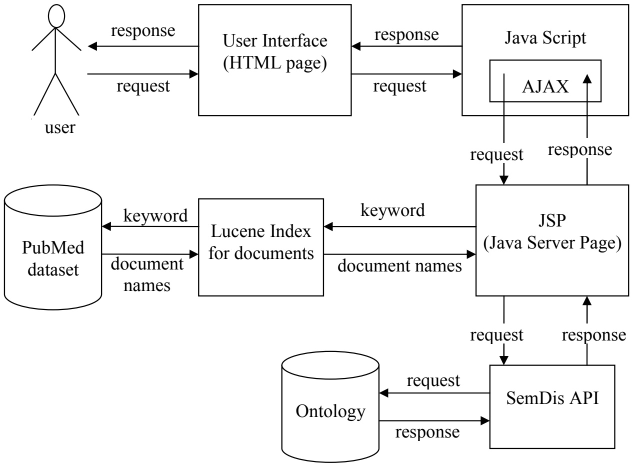
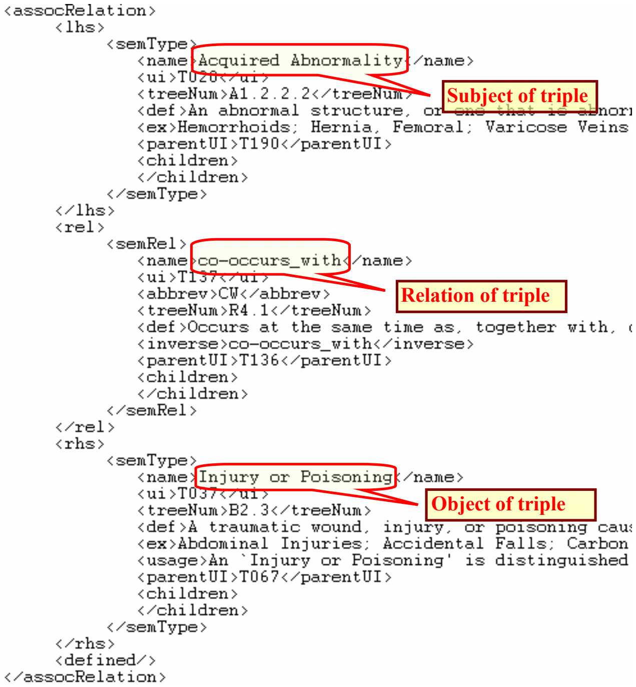


Fig. 3: Architecture of the Semantic Browser Application



```
<assocRelation>
    <lhs>
        <semType>
            <name>Acquired Abnormality</name>
            <ui>T020</ui>
            <treeNum>A1.2.2.2</treeNum>
            <def>An abnormal structure, or one that is abnor
            <ex>Hemorrhoids; Hernia, Femoral; Varicose Veins
            <parentUI>T190</parentUI>
            <children>
            </children>
        </semType>
    </lhs>
    <rel>
        <semRel>
            <name>co-occurs_with</name>
            <ui>T137</ui>
            <abbrev>CW</abbrev>
            <treeNum>R4.1</treeNum>
            <def>Occurs at the same time as, together with, 
            <inverse>co-occurs_with</inverse>
            <parentUI>T136</parentUI>
            <children>
            </children>
        </semRel>
    </rel>
    <rhs>
        <semType>
            <name>Injury or Poisoning</name>
            <ui>T037</ui>
            <treeNum>B2.3</treeNum>
            <def>A traumatic wound, injury, or poisoning caus
            <ex>Abdominal Injuries; Accidental Falls; Carbon
            <usage>An `Injury or Poisoning' is distinguished
            <parentUI>T067</parentUI>
            <children>
            </children>
        </semType>
    </rhs>
    <defined/>
</assocRelation>
```

Fig. 4: A triple in XML format from the UMLS dataset

The main benefit of this ontology is to let the user see the class of an entity instance, and all of the relationships that class has. By selecting any existing relationships, the user

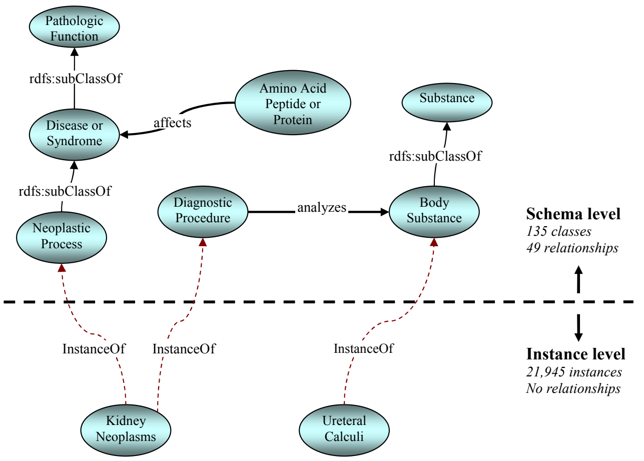


Fig. 5: Schema level and instance level of the ontology used

can traverse to another document which is indexed under the type that relationship is connected to.

Figure 5 illustrates the schema level and instance level of the ontology we used in our implementation. There is no direct relationship between concepts at the instance level. Without using the schema level, we can not say if there is any relation between "Kidney Neoplasm" and "Ureteral Calculi". As can be seen from the Figure 5, Kidney Neoplasm has two types in the ontology schema. Ureteral Calculi, however, has only one type in the ontology schema. Figure 5 demonstrates that there is an "analyzes" relationship between "Diagnostic Procedure" and "Body Substance" at the schema level. Diagnostic Procedure is one of the types of Kidney Neoplasm. Body Substance is the type of Ureteral Calculi. Because we have the "analyzes" relationship between "Diagnostic Procedure" and "Body Substance" at the schema level, we conclude that there is also "analyzes" relationship between "Kidney Neoplasm" and "Ureteral Calculi" instances at the instance level.

3.2 Dataset Used In The Project

In this project, we use PubMed as a dataset [6]. It has 16 million documents with abstracts. We used a fraction of that dataset (48,252 documents). Their format is illustrated in Figure 6.

To distinguish the MeSH terms in an abstract, those MeSH terms needed to be annotated. We have developed an algorithm for this annotation which works in linear time. The algorithm finds the MeSH terms in the sentences, and encloses them between specials tags. Figure 7 below is the annotated form of the abstract at Figure 6.

The algorithm tokenizes the text, and processes it word by word. When the algorithm finds a word in the text and that term exist also in the ontology, it does not annotate that word immediately. Instead, the algorithm continues processing the words in the text and tries to find the term with maximum

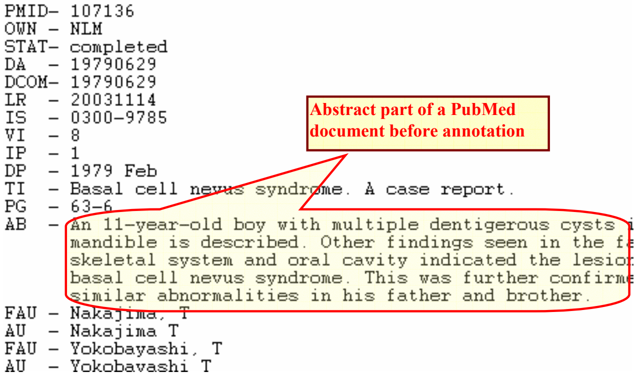

```
PMID- 107136
OWN - NLM
STAT- completed
DA   - 19790629
DCOM- 19790629
LR   - 20031114
IS   - 0300-9785
VI   - 8
IP   - 1
DP   - 1979 Feb
TI   - Basal cell nevus syndrome. A case report.
PG   - 63-6
AB   - An 11-year-old boy with multiple dentigerous cysts i
       mandible is described. Other findings seen in the fa
       skeletal system and oral cavity indicated the lesio
       basal cell nevus syndrome. This was further confirme
       similar abnormalities in his father and brother.
FAU - Nakajima, T
AU   - Nakajima T
FAU - Yokobayashi, T
AU   - Yokobavashi T
```

Fig. 6: File format of PubMed dataset before annotation

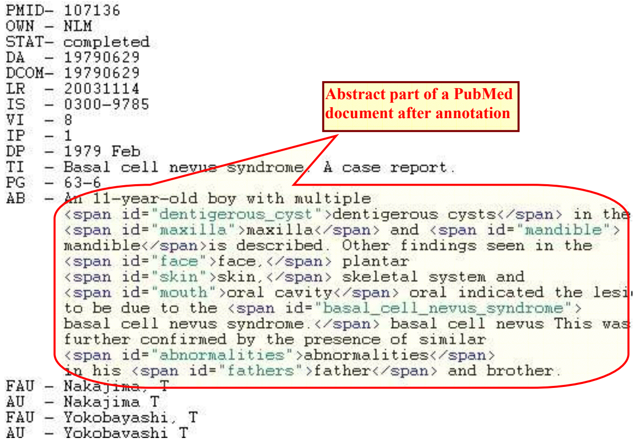

```
PMID- 107136
OWN - NLM
STAT- completed
DA   - 19790629
DCOM- 19790629
LR   - 20031114
IS   - 0300-9785
VI   - 8
IP   - 1
DP   - 1979 Feb
TI   - Basal cell nevus syndrome. A case report.
PG   - 63-6
AB   - An 11-year-old boy with multiple
      <span id="dentigerous_cyst">dentigerous cysts</span> in the
      <span id="maxilla">maxilla</span> and <span id="mandible">
      mandible</span>is described. Other findings seen in the
      <span id="face">face,</span> plantar
      <span id="skin">skin,</span> skeletal system and
      <span id="mouth">oral cavity</span> oral indicated the lesi
      to be due to the <span id="basal_cell_nevus_syndrome">
      basal cell nevus syndrome.</span> basal cell nevus This was
      further confirmed by the presence of similar
      <span id="abnormalities">abnormalities</span>
      in his <span id="fathers">father</span> and brother.
FAU - Nakajima, T
AU   - Nakajima T
FAU - Yokobayashi, T
AU   - Yokobavashi T
```

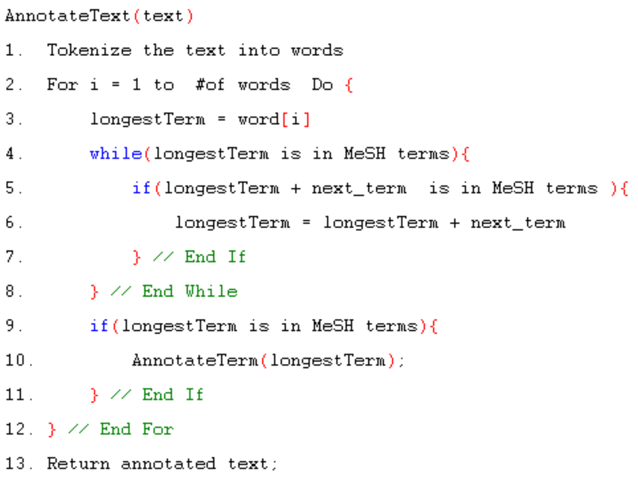
Fig. 7: File format of PubMed dataset after annotation

number of words in text which also exists in the ontology. The word "LongestTerm" in the Algorithm 8 refers to the term with maximum number of words.

3.3 Processing Data And Indexing The Articles

To index the documents in our dataset, we used Lucene [7]. By using the regular expression methods of the Java API (java.util.regex), we picked each annotated term in the documents, and associate each of these annotated terms with the document. As each document is associated with several MeSH terms, also each MeSH term is associated with several documents. In our implementation, we used 21,945 MeSH terms and their synonyms. Thus, we used around 104,000 terms to index our dataset.

The user interface is a simple HTML page using JavaScript and CSS. JavaScript contains the AJAX engine [9]. AJAX engine serves as an intermediary between the JavaScript methods and server. Because of its high performance and quick response time, we have used AJAX engine to send requests and to receive responses from the server. Server-side code is written in Java Server Pages (JSP) codes. After JSP receives the request from AJAX engine, it



```
AnnotateText(text)
Tokenize the text into words
For i = 1 to  #of words  Do {
    longestTerm = word[i]
    while(longestTerm is in MeSH terms){
        if(longestTerm + next_term  is in MeSH terms ){
            longestTerm = longestTerm + next_term
        } // End If
    } // End While
    if(longestTerm is in MeSH terms){
        AnnotateTerm(longestTerm);
    } // End If
} // End For
Return annotated text;
```

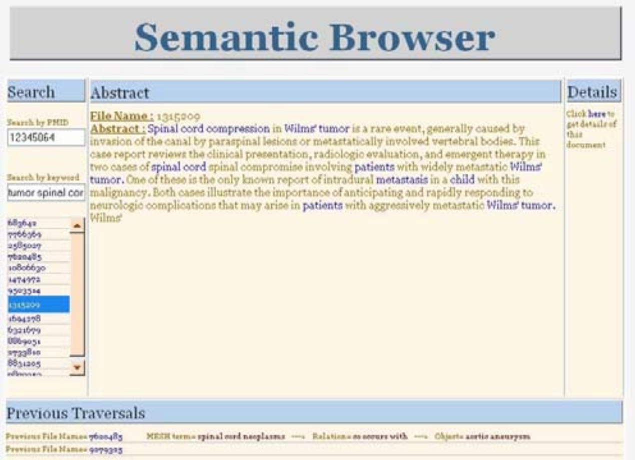
Fig. 8: Pseudocode of AnnotateText Algorithm

parses the Request String to get all of the parameters coming from AJAX engine. Based on the parameters received, a JSP method is called. The method then calls one of the methods in the Java class. Java class includes Lucene [7] and SemDis API [8] methods. Java methods return an array list (java.util.ArrayList) to the JSP. The JSP method returns the response to the AJAX as in XML format. The methods in the JavaScript can easily manipulate the response by using XML DOM methods. We then insert the response received from the AJAX to the HTML file, without having to reload the whole HTML page.

4. Semantic Browser Application

The articles to be shown in the user interface are annotated beforehand. To begin traversing between documents, the user needs a document to begin with. On the left column of the user interface, there are two text boxes, where the user can search documents by either the PMID number of the article or by any MeSH term which may appear in the documents. As the user types in the textbox, just below the textbox, a popup menu appears showing the MeSH terms, which begin with the typed letters.

After selecting any MeSH term from the list, the names of the files in which the selected MeSH term, or its synonym appears. For instance, by entering the keyword “neoplasm” in the search box, the user also gets file names in which “serum” appears, but “neoplasm” does not. Because, serum is a synonym of “neoplasm”, the user does not miss that document which may be very important for the user. Those documents are returned, because when we index the documents with Lucene, we have also used the synonyms of the MeSH terms. While the number of MeSH terms is 21,945, the number of terms used to index documents are around 104,000 including the synonyms of MeSH terms.



12345064

tumor spinal cor

683642
7766369
2585027
7620485
10806630
1474972
9503514
1315209
1694278
6321679
8869051
2733810
8831205

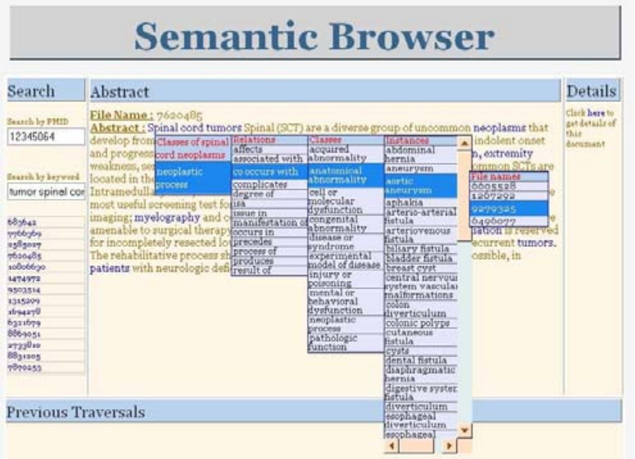
File Name : 1315209

Abstract : Spinal cord compression in Wilms' tumor is a rare event, generally caused by invasion of the canal by paraspinal lesions or metastatically involved vertebral bodies. This case report reviews the clinical presentation, radiologic evaluation, and emergent therapy in two cases of spinal cord spinal compromise involving patients with widely metastatic Wilms' tumor. One of these is the only known report of intradural metastasis in a child with this malignancy. Both cases illustrate the importance of anticipating and rapidly responding to neurologic complications that may arise in patients with aggressively metastatic Wilms' tumor. Wilms'

Previous File Name= 7620485 MESH term= spinal cord neoplasms ---> Relation= co occurs with ---> Object= aortic aneurysm

Previous File Name= 9079325

Fig. 9: A screenshot of Semantic Browser tool



12345064

tumor spinal cor

683642
7766369
2585027
7620485
10806630
1474972
9503514
1315209
1694278
6321679
8869051
2733810
8831205
7870253

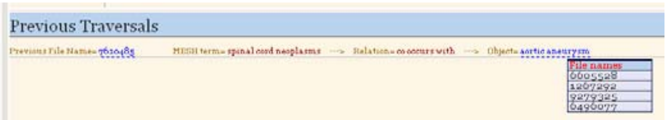
File Name : 7620485

Abstract : Spinal cord tumors Spinal (SCT) are a diverse group of uncommon neoplasms that develop from … indolent onset and progress… n, extremity weakness, se… ommon SCTs are located in the… Intramedulla… most useful screening test fo… imaging; myelography and c… amenable to surgical therapy… ation is reserved for incompletely resected lo… ecurrent tumors. The rehabilitative process sh… ossible, in patients with neurologic defi…

Fig. 10: A screenshot of Semantic Browser tool

The named entities in the text have different colors so that the user can understand that those terms are annotated based on some ontology, and will let the user traverse to other documents. This traversal is based on the semantic relationships between the two documents. By simply hovering the mouse pointer on that named entity, a popup menu appears with a list of types under which that entity instance is placed. By hovering over any of those types, the user also gets the list of relationships of that type. By hovering over any of those relations from the list, the user gets the list of types that the relationship selected is connected to. By hovering over any of the types from the list, user gets a list of MeSH terms which are instance of hovered types. By hovering over any of the MeSH terms from the list, user gets a list of file names in which the MeSH term appears. Then, the user simply clicks on any file name in the menu and the abstract part of the selected file appears on the webpage.

The AJAX technology makes this process is very fast, because the page does not have to be reloaded every time



Previous Traversals

Previous File Name= 7620485 MESH term= spinal cord neoplasms ---> Relation= co occurs with ---> Object= aortic aneurysm

File names
6005528
1267292
9279325
6496077

Fig. 11: Previous Traversals section of Semantic Browser

the content of the popup menus is retrieved from the server, and abstract of the file is retrieved from the PubMed dataset.

By picking a relationship in the popup menu, the user can go to another document. Just after this process, the name of the previous document is written into the "Previous Traversals" section of the Semantic Browser tool. Just next to the name of the file, the triple that the user went through in that document is written. Let us assume that the user reads the abstract of the document named "7620485". By hovering over the term "spinal cords neoplasm", the user picks the relationship "co occurs with". Then, the user picks "aortic aneurysm". Then among the file names appearing in the popup menu, the user selects the document "9279325". Just at the end of this process, the name of the previous file, 7620485, is written into "Previous Traversals" section. The traversal done in that document is written next to the file name. By simply clicking on the file name, the user can go to previous document easily. If the user wants to read other documents that contain the term "aortic aneurysm", without having to go back to previous file, user can hover on the term "aortic aneurysm" in the "Previous Traversals" section. When the popup menu appears, the user can select another file name and go to that document (See Figure 11).

Such a history mechanism is meant to be more than just a browsing aid. These recorded traversals contain entities and named relationships. They can therefore serve as "Semantic" indices in the document corpus. If the user discovered new and interesting information during such a traversal she could choose to remember the trail that she followed to get the information. In the future we plan to build a relationship-based document index that will allow retrieval of documents that were found along such trails. This is an idea very similar to "trailblazing" described by Dr. Vannevar Bush [4].

5. Related Work

In this paper, we presented an implementation that classifies documents, builds a Semantic Web by using the semantics of the relationships, and enables users to navigate among the documents in a meaningful way by using the relationships as opposed to hyperlinks which do not carry any semantics of relationships between the linked documents. In this section, we discuss some of the related work to our approach.

There has been a considerable amount of work done to discover the relationships between terms in unstructured text by using natural language processing techniques. There are several NLP (Natural Language Processing) tools that are used by third parties to parse the sentences in their dataset to extract the relationships. GATE (General Architecture for Text Engineering) [10], CGPARSER [11], OpenNLP [12], and Link Grammar [13] are some of the commonly used NLP tools. In our implementation however, we used a structured text (XML file) to generate triples (two entities and the relationship between them). By using these triples, we generate our ontology.

There are many algorithms for automatic clustering, such as the K-Means algorithm [14], hierarchical clustering [15] and Expectation Maximization [16] to form the clusters. These algorithms do not utilize the relationships that exist between the terms. In their paper "Text Clustering using Semantics" [17], the authors have proven that using relationships between the documents increases the accuracy. In their work, they demonstrate that the text clustering using semantics outperforms other clustering algorithms which do not use semantics of relationships. The advantage of our application is that we have the relationships not from document to document, but from term to term. These terms may appear either in the same document, or also in different documents. Thus, any two documents may be connected to each other via multiple relationships that exist between term pairs.

There are some "Semantic Browser" implementations [18]–[22] that enable users to navigate between the different websites by using semantic relationships. To the best of our knowledge, our Semantic Browser tool is the only existing work that uses AJAX technology [9]. By using AJAX technology, and JavaScript that works on the client-side, users do not have to install anything to their machine. AJAX technology works on almost all of the commonly-used browsers, such as; Microsoft Internet Explorer, Mozilla Firefox, Chrome, SeaMonkey, Camino, Flock, Epiphany, Galeon, Netscape, and Apple Safari.

Anyone who has used Flickr, GMail, Google Suggest, or Google Maps will realize that a new breed of dynamic web applications is emerging. These applications look and act very similar to traditional desktop applications without relying on plug-ins or browser-specific features. Web applications have traditionally been a set of HTML pages that must be reloaded to change any portion of the content. Technologies such as JavaScript programming language and cascading style sheets (CSS) have matured to the point where they can be used effectively to create very dynamic web applications that will work on all of the major browsers [23]. Likewise our implementation, although being a web application, looks and acts like a desktop application.

Semantic Web content and Semantic Web tools depend each other. Without sufficient Semantic Web content, few tools will be written to consume it; without many such tools, there is little appeal to publish Semantic Web content.

In the Piggy Bank project (MIT) [22], they define this problem as “chicken-and-egg problem”, and they provide a web browser extension called Piggy Bank that lets users make use of Semantic Web content within web content as users browse the Web. By using Piggy Bank, a tool integrated into the contemporary web browser, Firefox, web users extract information items from within web pages and save them in RDF format [24]. When the user visits a web site, if there is a structured data of the same information available to retrieve, the web browser shows a “data coin” icon in the status bar for each site. By clicking on that icon, the “pure” information from each web site is collected. The browser Piggy Bank shows the information items it has collected from one of the sites, right inside the same browser window. The user can also tag an item with one or more keywords, to help him/her find it later. The “tag completion” dropdown suggests previously used tags that the user can pick from. The user can also tag or save several items together. To make this effort in collaborative way, with one click on the “Publish” button for each item, the user publishes information to the Semantic Bank. Thus, other user can use this information. Although having some similarity to our project, their approach differs from ours in the usage of semantics of relationships. They cluster documents by using metadata in RDF, but not using relationships.

The project “Magpie” [26] is similar to our project. However, in Magpie, a plug-in is needed to be installed to browser, whereas in our project no plug-in is needed. In the Magpie, they have very few relationships (around 6-7), whereas we have 49 relationships in our project. Because they have also very few classes (around 4 usually), they show the different classes with different colors. Having 135 classes in our project, this was not possible.

6. Conclusions and Future Work

In this paper, we described a tool that allows users to navigate between documents using virtual relationships. However, currently we use relationships that exist between the types of instances, not between the instances. This method works most of the time. However, we can not assume that it holds true all the time. Because it is not guaranteed that when there is a relationship between any two types, the relationship should exists between every instances of these two types. For instance; in our ontology schema, there is a relationship “adjacent_to” between classes “body part organ” and “body location or region”. Some of the instances of the class “body part organ” are breast, prostate. Some of the instances of the class “body location or region” are cheek, chin, elbow, and abdomen. If we try to put “adjacent_to” relationship between some of instances, we would get incorrect statements. If we put “adjacent_to” relationship between prostate and chin, then this would not be a correct statement. In order to always get correct results, we need relationships at the instance level of the ontology, not at the schema level.

Currently, the indexing using Lucene [7] does not take relationships into consideration. We will index our dataset by also using the relationships that are extracted from the documents. In this way, more relevant documents will be returned to the user.

Besides providing the users ease of browsing between the documents, our goal is also to reduce the time that a user has to spend with documents. Among the returned multiple documents, we should rank the documents based on their relevance to the triples, the user may be interested to learn.

Our work is one of the earliest attempts at utilizing the semantic relationships to support browsing and navigation of a document space.

References

[1] Unified Medical Language System. http://umlsinfo.nlm.nih.gov

[2] Medical Subject Headings. http://www.nlm.nih.gov/mesh

[3] Home Remedies and Natural Cures for Common Illnesses http://www.home-remedies-for-you.com/remedy/Measles.html

[4] Vannevar Bush: As We May Think. The Atlantic Monthly 176(1): 101-108 (1945)

[5] Kshitij Shah, Amit P. Sheth: Logical Information Modeling of Web-Accessible Heterogeneous Digital Assets. 266-275

[6] National Center for Biotechnology Information. http://www.ncbi.nlm.nih.gov

[7] Apache Lucene Home Page. http://lucene.apache.org/java/docs

[8] Semantic Discovery: Discovering Complex Relationships in Semantic Web. http://lsdis.cs.uga.edu/projects/semdis/sweto/index.php?page=5

[9] Ajax: A New Approach to Web Applications by Jesse James Garrett. http://www.adaptivepath.com/publications/essays/archives/000385.php

[10] GATE, General Architecture for Text Engineering. http://gate.ac.uk

[11] Contextual Grammars. http://www.uni-koblenz.de/ harbusch/CG-PARSER/welcome-cg.html

[12] OpenNLP Home Page. http://opennlp.sourceforge.net

[13] Link Grammar. Davy Temperley, Daniel Sleator, John Lafferty. http://www.link.cs.cmu.edu/link

[14] J. B. MacQueen (1967): "Some Methods for classification and Analysis of Multivariate Observations, Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press, 1:281-297

[15] A.K. Jain and R.C. Dubes. Algorithms for Clustering Data, Prentice Hall, Englewood Cliffs NJ, U.S.A., 1988.

[16] Arthur Dempster, Nan Laird, and Donald Rubin. "Maximum likelihood from incomplete data via the EM algorithm". Journal of the Royal Statistical Society, Series B, 39(1):1Ű38, 1977

[17] Choudhary, R., Bhattacharyya, P.: Text Clustering Using Semantics. The 11th International World Wide Web Conference, WWW2002, Honolulu, Hawaii, USA (2002)

[18] Haystack Project. http://haystack.lcs.mit.edu

[19] Semantic Browser. Associative Similarity Browsing. http://semanticbrowser.aspasia-systems.de

[20] BigBlogZoo Home Page. http://www.bigblogzoo.com

[21] Amblit Navigator. http://www.amblit.com/products

[22] Piggy Bank Home Page. http://simile.mit.edu/piggy-bank

[23] AJAX. http://java.sun.com/developer/technicalArticles/J2EE/AJAX

[24] Resource Description Framework. http://www.w3.org/RDF

[25] Semantic Browser. http://lsdis.cs.uga.edu/projects/semdis/SemanticBrowser/

[26] Martin Dzbor, John B. Domingue, and Enrico Motta. Magpie - towards a semantic web browser. In Proceedings of the 2nd Intl. Semantic Web Conference,October 2003. Sanibel Island, Florida.

Semantic Clickstream Mining

Mehrdad Jalali[1], **and Norwati Mustapha**[2]
[1] Department of Software Engineering, Mashhad Branch, Islamic Azad University, Mashhad, Iran
[2] Department of Computer Science, Universiti Putra Malaysia, Malaysia

Abstract - *Nowadays Web users are drowned in all kind of available information. However, only a tiny part of it is usually relevant to their preferences. Web usage mining which extracts knowledge for usage and clickstream data has become the subject of exhaustive research, as its potential for Web-based personalized services, prediction of user near future intentions, adaptive Web sites, and customer profiling are recognized. Moreover, semantic web aims to enrich the WWW by machine processable information which supports the user in his tasks. Semantic clickstream mining which integrates semantic in Web usage mining processes aims to improve the quality of the Web usage mining systems. Given the primarily syntactical nature of data Web usage mining operates on, the discovery of meaning is impossible based on these data only. Therefore, formalizations of the semantics of Web resources and navigation behavior are increasingly being used. In this paper, we discuss the interplay of the Semantic Web with Web usage mining and also we give an overview of where the two research areas meet today. Moreover a proposed framework to integrate semantic Web and Web usage mining discuss in the rest of the paper..*

Keywords: Web usage mining, Clickstream Mining, Semantic Web, Semantic Web usage mining

1 Introduction

Semantic Web Usage Mining (SWUM) or Semantic Clickstream Mining aims to integrate two research areas Semantic Web and Web Usage Mining for obtaining more fine-grained and meaningful user behaviours in the Web environment. To better understand user's next intentions from observing him while navigating on a Website, all semantically interaction data needs to be tracked as well as tracking clickstream data. The most important issue facing in the classical Web Usage Mining system is quality of the results. The aim of this paper is to give an overview of where the two areas meet today, and what we can do to improve the results of integrating semantic Web and Web Usage Mining.

The remainder of this paper is organized as follows: Section 2 covers a brief overview of the areas Semantic Web and Web Usage Mining. Section 3 describes some related research about semantic Web usage mining and introduces proposed framework for integrating semantic Web and Web usage mining. Finally, section 4 concludes the current study and sheds light on some directions in the future works.

2 Web Usage Mining and Semantic Web

In the first part of this section, we cover some backgrounds in the WUM systems. In the second part, we recall our understanding of semantic Web. A brief discussion about integrating these two areas will be illustrated in the end of this section.

2.1 Web Usage (Clickstream) Mining

In general, Web mining can be characterized as the application of data mining to the content, structure, and usage of Web resources [1, 2]. The goal of Web mining is to automatically discover local as well as global models and patterns within and between Web pages or other Web resources. However, Web mining tools aim to extract knowledge from the Web, rather than retrieving information. Research on Web mining is classified into three categories, which are Web structure mining that identifies authoritative Web pages, Web content mining that classifies Web documents automatically or constructs a multilayered Web information base, and Web usage mining that discover user access patterns in navigating Web pages [3]. The goal of Web usage mining, in particular, is to capture and model Web user behavioral patterns. The discovery of such patterns from the enormous amount of data generated by Web and application servers has found a number of important applications. Among these applications are systems to evaluate the effectiveness of a site in meeting user expectations [4], techniques for dynamic load balancing and optimization of Web servers for better and more efficient user access [5], and applications for dynamically restructuring or customizing a site based on users' predicted needs and interests .

From the data-source perspective, both Web structure and Web content mining target the Web content, while Web usage mining targets the Web access logs. Web usage mining (WUM) comprises three major processes: data pretreatment, data mining, and pattern analysis [3]. Pretreatment performs a series of processing on Web log files, which are data conversion, data cleaning, user identification, session identification, path completion, and transaction identification. Next, mining algorithms are applied to extract user navigation patterns. A navigation pattern represents the relationships among Web pages in a particular Web site. Some pattern analyzing algorithm is applied to extract data from data mining part for the recommendation system. Recently, a number of Web usage mining (WUM) systems have been proposed to predict user's preferences and their navigation behaviors.

More recently, Web usage mining techniques have been proposed as another user-based approach to personalization which alleviates some of the problems associated with collaborative filtering. In particular, Web usage mining has been used to improve the scalability of personalization systems based on traditional CF-based techniques. In [6] we advance an architecture for online predicting in Web usage mining recommendation system and propose a novel approach to classifying user navigation patterns for predicting users' future requests. The approach is based on using the longest common subsequence (LCS) algorithm in classification part of the system. All of these works attempt to find architecture and algorithm to improve accuracy of personalized recommendation, but the accuracy still does not meet satisfaction especially in large-scale Websites. Fig. 1 illustrates the state of the proposed system.

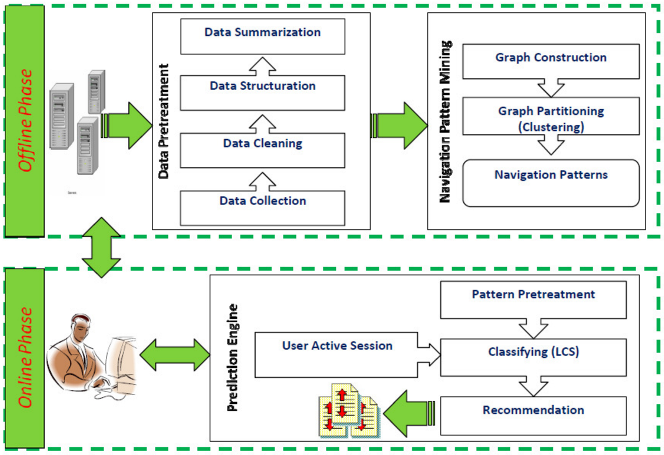


Fig.1 A WUM system framework [6]

To improve the quality of the results of the WUM systems, domain knowledge about a Web site can be integrated to the WUM process. Domain knowledge can be integrated into the Web usage mining process in many ways. This includes leveraging explicit domain ontologies or implicit domain semantics extracted from the content or the structure of documents or Web site. In general, however, this process may involve one or more of three critical activities [7]:

2.1.1 Domain ontology acquisition

The process of acquiring, maintaining and enriching the domain ontologies is referred to as "ontology engineering". For small Web sites with only static Web pages, it is feasible to construct a domain knowledge base manually or semi-manually. The outcome of this phase is a set of formally defined domain ontologies that precisely represent the Web site. Good representation should provide machine understandability, the power of reasoning, and computation efficiency.

2.1.2 Knowledge base construction

While the first phase generates the formal representation of concepts and relations among them, the second phase, knowledge base construction, can be viewed as building mappings between concepts or relations on the one hand, and objects on the Web. The goal of this phase is to find the instances of the concepts and relations from the Web site's domain, so that they can be exploited to perform further data mining tasks. Information extraction methods play an important role in this phase.

2.1.3 Knowledge-enhanced pattern discovery

Domain knowledge enables analysts to perform more powerful Web data mining tasks. For example, semantic knowledge may help in interpreting, analyzing, and reasoning about usage patterns discovered in the mining phase.

In the following we introduce semantic Web which can be integrated to the Web usage mining process.

2.2 Semantic Web

The Semantic Web aims to obtain machine-understandable information from WWW which is based on a vision of Tim Berners-Lee, the inventor of the WWW. The great success of the current WWW leads to a new challenge: a huge amount of data is interpretable by humans only; machine support is limited. He suggested to enrich the Web by machine-processable information which supports the user in his tasks. For instance, today's search engines are already quite powerful, but still frequently return overly large or inadequate lists of hits. Machine-processable information can point the search engine to the relevant pages and improve the quality of the results. Fig. 2 shows the layers of the Semantic Web as suggested by Berners-Lee. This architecture is discussed in detail for instance in [8] and [9].

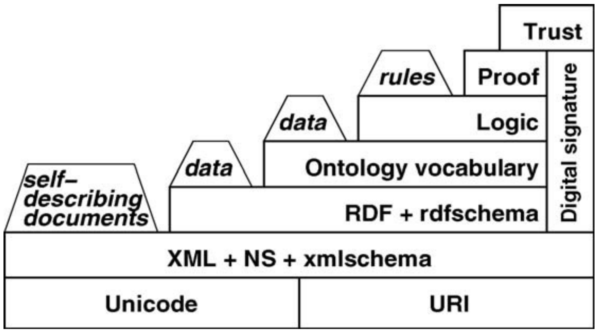


Fig. 2 the layers of the Semantic Web

Today, it is almost impossible to retrieve information with a keyword search when the information is spread over several pages. Consider the query for Web mining experts in a company intranet, where the only explicit information stored are the relationships between people and the courses they attended on one hand, and between courses and the topics they cover on the other hand. In that case, the use of a rule stating that people who attended a course which was about a certain topic have knowledge about that topic might improve the results.

The process of building the Semantic Web is today still under way. Its structure has to be defined, and this structure should be brought to life. Given the primarily syntactical nature of data Web usage mining operates on, the discovery of meaning is impossible based on these data only. Therefore, semantic knowledge of Web documents and navigation behaviors can be utilized in recommendation systems for predicting different types of complex Web document and object based on underlying properties and attributes especially in large-scale and dynamic Web sites. Mapping between user navigation transactions in Web usage mining to semantic transaction based on concepts and objects can improve the accuracy of the Web usage mining personalization system.

3 Semantic Clickstream Mining

Our goal in this section is to provide a road map for the integration of semantic and ontological knowledge into the process of Web usage mining. Semantics can be utilized for Web Usage Mining for different purposes which are introduced in this section.

To better integrate semantic Web and WUM it would be desirable to have a rich semantic model of content and structure of a site. This model should capture the complexity of the manifold relationships between the concepts covered in a site, and should be ”built into” the site in the sense that the pages requested by visitors are directly associated with the concepts and relations treated by it. This leads to semantic Web usage mining. Semantic Web usage mining involves the integration of domain knowledge into Web usage mining [7] . Utilizing semantic knowledge can lead to deeper interaction of the Website's user with the site. Integration of domain knowledge allows such systems to infer additional useful recommendations for users based on more fine grained characteristics of the objects being recommended, and provides the capability to explain and reason about user actions.

The interpreting, analysing, and reasoning about usage patterns discovered in the mining phase can be done by using semantic knowledge. Moreover, it can improve the quality of the recommendations in the usage-based system.

Several studies have considered various approaches to integrate content-based semantic knowledge into traditional usage-based recommender systems. An overview of the existing approaches as well as a some framework for integrating domain ontologies with the personalization process based on Web usage mining is given in the following.

Our main research question is; Can usage patterns reveal further relations to help build the Semantic Web? This field is still rather new, so we will only describe an illustrative selection of research approaches.

Ypma and Heskes propose a method for learning content categories from usage [10]. They model navigation in terms of hidden Markov models, with the hidden states being page categories, and the observed request events being instances of them. Their main aim is to show that a meaningful page categorization may be learned simultaneously with the user labeling and intercategory transitions; semantic labels (such as “sports pages”) must be assigned to a state manually. The resulting taxonomy and page classification can be used as a conceptual model for the site, or used to improve an existing conceptual model.

Chi et al.[7] identify frequent paths through a site. Based on the keywords extracted from the pages along the path, they compute the likely “information scent” followed, i.e. the intended goal of the path. The information scent is a set of weighted keywords, which can be inspected and labeled more concisely by using an interactive tool. Thus, usage creates a set of information goals users expect the site to satisfy. These goals may be used to modify or extend the content categories shown to the users, employed to structure the site's information architecture, or employed in the site's conceptual model.

Stojanovic, Maedche, Motik, and Stojanovic [11] propose to measure user interest in a site's concepts by the frequency of accesses to pages that deal with these concepts. They use these data for ontology evolution: Extending the site's coverage of high-interest concepts, and deleting low-interest concepts, or merging them with others.

The combination of implicit user input (usage) and explicit user input (search engine queries) can contribute further to conceptual structure. User navigation has been employed to infer topical relatedness, i.e. the relatedness of a set of pages to a topic as given by the terms of a query to a search engine. A classification of pages into “satisfying the user defined predicate” and “not satisfying the predicate” is thus learned from usage, structure, and content information. An obvious application is to mine user navigation to improve search engine ranking [12].

Many approaches use a combination of content and usage mining to generate recommendations. For example, in contentbased collaborative filtering, textual categorization of documents is used for generating pseudo-rankings for every userdocument pair [8]. In [9], ontologies, IE techniques for analyzing single pages, and a user's search history together serve to generate recommendations for query improvement in a search engine.

In [13], Authors have presented a general framework for using domain ontologies to automatically characterize usage profiles containing a set of structured Web objects. Their motivation has been to use this framework in the context of Web personalization, going beyond page-level or item-level constructs, and using the full semantic power of the underlying ontology. They considered a Web site as a collection of objects belonging to certain classes (resulting in a concept

Hierarchy of Genre's a portion of which is shown in Fig. 3. Given a collection of similar user sessions (e.g., obtained through clustering) each containing a set of objects, they have shown how to create an aggregate representation of for the whole collection based on the attributes of each object as defined in the domain ontology (Fig. 4). This aggregate representation is a set of pseudo objects each characterizing objects of different types commonly occurring across the user sessions. They have also presented a framework for Web personalization based on domain-level aggregate profiles.

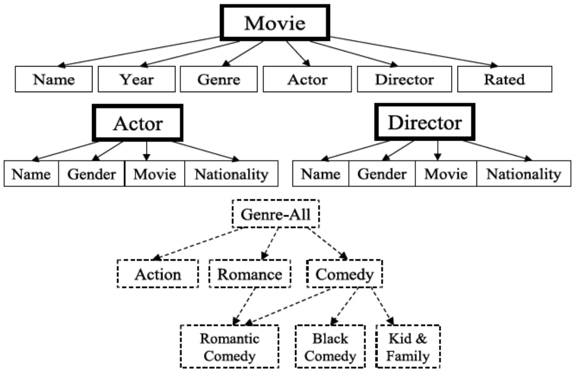


Fig. 3 The Ontology for a movie Web site

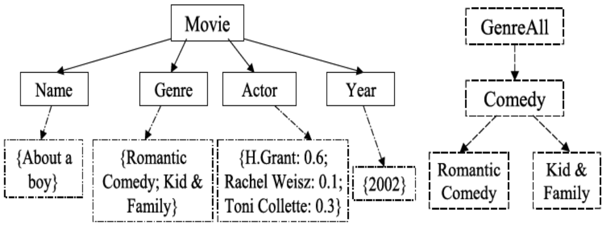

Movie
Name
Genre
Actor
Year
{About a boy}
{Romantic Comedy; Kid & Family}
{H.Grant: 0.6; Rachel Weisz: 0.1; Toni Collette: 0.3}
{2002}
GenreAll
Comedy
Romantic Comedy
Kid & Family

Fig. 4 An Example of an Object in Class Movie

In [14], they proposed an approach to track user interaction data and preserving semantic knowledge on complex and interactive Web sites. They showed that the approach comes with some major enhancements compared to some existing solutions. The usage of Microformats enables an easy integration into existing Web sites and allows then to interrelate data on these sites (Microformats are small patterns of HTML to represent commonly published things like people, events, blog posts, reviews and tags in web pages. Microformats enable the publishing of higher fidelity information on the Web; the fastest and simplest way to provide feeds and APIs for the information in your website.). This also allows them to obtain fine-grained information connected with semantic knowledge that opens new chances to personalize Web sites.

Recommender systems rely on relevance scores for individual content items; in particular, pattern-based recommendation exploits co-occurrences of items in user sessions to ground any guesses about relevancy. To enhance the discovered patterns' quality, the authors in [15] propose using metadata about the content that they assume is stored in a domain ontology.

Their approach comprises a dedicated pattern space built on top of the ontology, navigation primitives, mining methods, and recommendation techniques.

Web usage mining (WUM) approaches often use terms and frequencies to represent a Web site for the mining process. In [16], the authors show that these representations lead to poor results. Therefore, it is proposed to perform a semantic Web usage mining process to enhance quality of the mining results. In this paper it was used a concept-based Web usage mining process to generate more semantically related results.

The approach was used to enhance a real Web site and it was evaluated by comparing it with four different WUM methods. It was defined two quality measures (interest and utility) in order to evaluate the results. These measures are obtained using surveys to 100 visitors of the site. Based on interest and correlation measures, it was proved that concept-based approach allows obtaining results closer to visitors' real browsing preferences. Moreover, information produced by the proposed approach lead to the discovery of enhancements. The proposed method also finished the generalization task in a few minutes which is not too much compare with other methods.

In [10], they proposed the integration of semantic information drawn from a Web application's domain knowledge into all phases of the Web usage mining process (preprocessing, pattern discovery, and recommendation/prediction). The goal is to have an intelligent semantics-aware Web usage mining framework. This is accomplished by using semantic information in the sequential pattern mining algorithm to prune the search space and partially relieve the algorithm from support counting. In addition, semantic information is used in the prediction phase with low order Markov models, for less space complexity and accurate prediction that will help solve ambiguous predictions problem. Experimental results show that semantics-aware sequential pattern mining algorithms can perform 4 times faster than regular non-semantics-aware algorithms with only 26% of the memory requirement.

Fig. 5 illustrates the proposed architecture for a semantic Web usage mining system which can be used in a recommender system. In the offline phase of the system to perform semantic data pretreatment, Web site ontology and a knowledgebase which are created based on the content and structure of the Web site can be utilized in the process of this module. On the other hand to create semantic usage data which further will be used in semantic navigation pattern, and to understand semantic knowledge about user semantically' sessions in a particular website, this module needs to integrate with those ontology and knowledgebase.

In the next module, semantic navigation patterns will be extracted from the sessions by utilizing semantic clustering algorithm. In the online phase, the system recommends some pages and concepts which the current users intend to navigate them through the particular Web site. This phase in similar to our work which is described in [6].

In summary, all of these works attempt to find reference architecture and framework to improve quality of the Web usage mining systems by integrating semantic web to WUM, In the proposed framework, we advance a framework to integrate semantic web and WUM, which by using appropriate semantic clustering algorithm and well-done ontology and knowledgebase design, the quality of the semantic Web usage mining can be enhanced as future work.

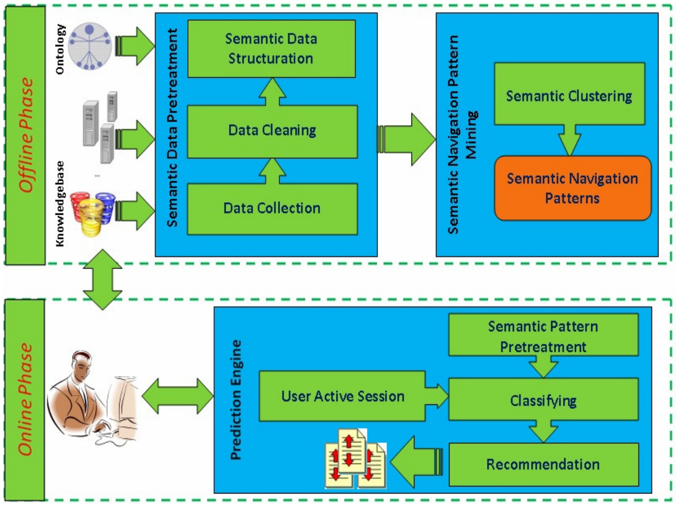


Fig. 5 The proposed framework

4 Conclusion and Future Work

In this paper, we have studied the integration of the two fast developing research areas Semantic Web and Web usage mining. We discussed how Semantic Web usage mining can improve the results of Web usage Mining systems by by exploiting the semantic structures in the process of the Web usage mining. Moreover, a proposed framework to integrate semantic Web and Web usage mining discussed in this paper. As a future direction, we plan to develop a semantic Web usage mining system which uses the proposed framework.

5 References

[1] R. Cooley, et al., "Web mining: information and pattern discovery on the World Wide Web," in Ninth IEEE International Conference on Tools with Artificial Intelligence, Newport Beach, CA, USA, 1997, pp. 558-567.

[2] J. Srivastava, et al., "Web usage mining: discovery and applications of usage patterns from Web data," ACM SIGKDD Explorations Newsletter, vol. 1, pp. 12-23, 2000.

[3] B. Mobasher, et al., "Creating adaptive Web sites through usage-based clustering of URLs," in Knowledge and Data Engineering Exchange, Chicago, IL, USA, 1999, pp. 19-25.

[4] M. Spiliopoulou, "Web usage mining for web site evaluation," Communications of the ACM, vol. 43, pp. 127-134, 2000.

[5] J. E. Pitkow, et al., WebVis: A Tool for World Wide Web Access Log Analysis: Graphics, Visualization & Usability Center, Georgia Institute of Technology, 1994.

[6] M. Jalali, et al., "WebPUM: A Web-based recommendation system to predict user future movements," Expert Systems with Applications, vol. 37, pp. 6201-6212, 2010.

[7] E. H. Chi, et al., "Using information scent to model user information needs and actions and the Web," in Proceedings of the SIGCHI, 2001, pp. 490-497.

[8] P. Melville, et al., "Content-boosted collaborative filtering for improved recommendations," in Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2002, pp. 187-192.

[9] S. Parent, et al., "An adaptive agent for web exploration based on concept hierarchies," in In Proceedings of the 9th International Conference on Human Computer Interaction, 2001.

[10] N. R. Mabroukeh and C. I. Ezeife, "Using domain ontology for semantic web usage mining and next page prediction," in CIKM '09 Proceeding of the 18th ACM conference on Information and knowledge management, 2009, pp. 1677-1680.

[11] L. Stojanovic, et al., "User-driven ontology evolution management," Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, pp. 133-140, 2002.

[12] C. Kemp and K. Ramamohanarao, "Long-term learning for web search engines," Principles of Data Mining and Knowledge Discovery, pp. 243-311, 2002.

[13] H. Dai and B. Mobasher, "Using ontologies to discover domain-level web usage profiles," Semantic Web Mining, p. 35, 2002.

[14] T. Plumbaum, et al., "Semantic web usage mining: Using semantics to understand user intentions," User Modeling, Adaptation, and Personalization, pp. 391-396, 2009.

[15] M. Adda, et al., "Toward recommendation based on ontology-powered web-usage mining," IEEE Internet Computing, pp. 45-52, 2007.

[16] S. R os and J. Velásquez, "Semantic web usage mining by a concept-based approach for off-line web site enhancements," 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 234-241, 2008.

Annotation of Hyperlinks with Semantic Meaning

Bilal Gonen[1], **Samir Tartir**[2], **Ravi Pavagada**[3]
[1]Computer Science and Engineering Department, University of Nevada, Reno, Reno, Nevada, U.S.A.
[2]Department of Software Engineering, Philadelphia University, Amman, Jordan
[3]IT, Excelacom Inc, Reston, VA, U.S.A.

Abstract— *Web pages in the web represent certain concepts in the domain they fall in, and the connections between them represent the relations between the concepts they represent. In the current web, people are using links blindly without knowing what these links point to, or what kind of relationship this link represents. With the advent of the semantic web, concepts and relationships among them are represented in an ontology. This can be utilized to make links more meaningful. Web pages can be searched, browsed or even reorganized based on their concept and relationship labels. Links in a webpage can render useful information about the page it is pointing to. We can annotate a webpage and its links with appropriate concepts from ontology. This paper presents a new idea of propagating concept from a webpage to the links pointing to that page or from the links to the webpage. Propagation of concepts is based on certain criteria which will be discussed later in this paper. We also propose a new idea of automated voting which is used to choose the right concept or relation from a number of concepts and relation matches.*

Keywords: semantic web, link annotation, data mining

1. Introduction

The network of hyperlinked documents, as it exists now, lacks semantic information in machine understandable form. It can only be browsed or searched by keywords -not concepts. There exist projects that automatically or semi-automatically annotate web pages with concepts taken from ontology. This effort makes web pages more understandable for machine processing and searching. In our project we would like to focus more on navigational implications of adding semantic annotation to web pages. Currently user or machine navigates between web pages by traversing them via hyperlinks. Decision if accessed page is relevant to the undertaken search can be made only after retrieving and analyzing the destination web page. In our project, we would like to add more semantic meaning to links themselves on the source page, so concepts included on target page can be evaluated without retrieving page itself.

In this paper, we use well formed computer science department ontology to annotate links and web pages with concepts. Web pages and links of the page can then be associated with concepts and relations from ontology. For example, web pages from computer science department of University of Georgia web site can be associated with concepts such as faculty, department, course, lecturer, research assistant etc... These web pages can therefore be treated as concept instances. Relationship can be defined between a webpage and its link. For instance, a student's webpage might have a link to his course page. In ontology there could be a relation say "takes" between the student and the course. This information will be annotated in the links along with the link concept "course". We have used ontology dictionary which associates labels to each concept and relations in the ontology. These labels are very useful in concept matching. Labels play the key role, since they are matched with the page contents and link window to extract appropriate concepts. We havenŠt used NLP techniques to get the concept matches. Our goal is to start with set of plain, connected web pages and by extracting information and matching them with the ontological concepts and also annotate the links with concepts and relations joining them. In this project, we would like to utilize already known algorithms and solution for page annotations. We think that combining different approaches of page annotation and information/concept propagation between web pages can improve the overall quality of annotated data.

Paper is organized as follows. In Section 2, we describe the related work and our ideas. Section 3 briefs our work and discusses the architecture of the proposed system. Section 4 explains the approach we took in building the proposed system. Section 5 describes propagation and voting schemes used. Section 6 describes the testing and experimental results, and Section 7, Conclusion and Future work.

2. Related Work and Our Ideas

There are papers on HTML Tag tree extraction or deriving link context [1] [2] [3]. One of them is ŞDeriving link-context from HTML tag treeŤ by Gautam Pant et al [4]. There are also other papers on automated semantic annotations [5] [6] [7] [8] [9] [10]. "SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotations" [11] talks about automation of web page annotations. "Mining the link structure of semantic web", by Souman Chakrabarti et al. [12] talks about HITS algorithm which takes advantage of the hubs in some fields and uses techniques that take advantage of social organizations of the web and allocates weights for the hub pages and

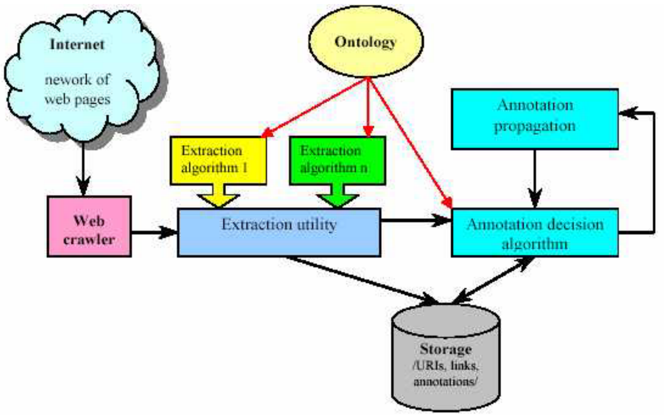


Fig. 1: General System Architecture

authorities in iterative process. The paper "On extracting link information by relationships instances from a website" by Myo-Myo Naing et al [13] talks about a web page which is being associated with a concept in ontology and links two different web pages based on the relationship between concepts in the ontology.

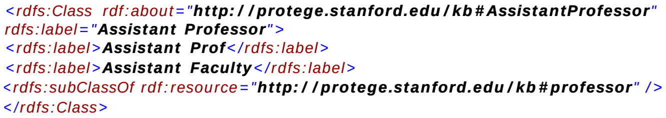
Our work is slightly different from their work. We incorporate voting of relations whenever there are more than one relation matches between two concepts. Concept matching is an area in itself and there are lots of papers on it. There are lots of AI and natural language processing techniques used to achieve this. As mentioned previously, we have concentrated more on concept labels defined in the ontology to find a concept match. Our work concentrate more on the information which is around the link, i.e. link context and match the link to a concept. New idea of concept propagation is proposed which would propagate concepts from a Webpage to the links pointing to that page if there is a tie in the number of concept matches for the given set of links. Propagation from links to page is done if most of the links agree on a single concept. Voting of concepts and relations is done whenever there is ambiguity.

3. Architecture Overview at a high level

We would like to make our system modular and expandable for future needs. As we cannot modify the content of web pages, we can only keep discovered annotations of pages and links in snapshot of selected web pages.

3.1 WebCrawler

Web Crawler, crawls the web structure and supplies the raw data for further analysis. HTML from web pages is analyzed by extraction utility. The extraction mechanism tries to match the whole page to some concepts in ontology.



```
<rdfs:Class rdf:about="http://protege.stanford.edu/kb#AssistantProfessor"
rdfs:label="Assistant Professor">
<rdfs:label>Assistant Prof</rdfs:label>
<rdfs:label>Assistant Faculty</rdfs:label>
<rdfs:subClassOf rdf:resource="http://protege.stanford.edu/kb#professor" />
</rdfs:Class>
```

Fig. 2: Labels for a concept

3.2 Ontology Dictionary

Ontology dictionary is the key part of our project. Dictionary labels are assigned for concepts and relations in the ontology. Dictionary labels for relations and concepts are comprised of hypernyms, synonyms and homonyms. We have added RA as a label for Research Assistant, TA as a label for Teaching Assistant etc.

We add labels to each of the concept in the ontology. Figure 2 is an extract from the ontology which describes an rdf:class Assistant Professor and its associated labels namely Assistant Professor, Assistant Faculty etc.

3.3 Extraction utility

Extraction utility is comprised of page and link analyzer, which analyses the page for the tags and assembles a vector of number of concept matches for each tag. The vector size is determined by the number of concepts in the ontology. We have prioritized various html tags in the webpage based on its importance. For example. <Title>, <Head>, and <Body> tag are given the most importance. It also tries to categorize links in this web page based only on information contained in the link window. Link extractor extracts the text of information based on the window sizes or number of bytes of text before and after the link. It then assembles the concept weights based for each of the link window sizes namely 0, 50 (25 words before and after the link) and 150 (75 words before and after the link).

3.4 Annotation decision

Voting is done whenever there is more than one concept matching a given page or a link. Based on the values set in the configurator, i.e. relative importance of tags or it could be based on relative size of the windows 0, 50 or 150, the voter calculates the new vote by calculating the product of the weight vector to the weights assigned in the configurator. Our configurator is flexible and easier to change. We have assigned weights of 0.5, 0.3, and 0.2 weights for anchor text window sizes of 0, 50 and 150 respectively. We have assigned weights of 0.4, 0.3, and 0.3 for "title", "head", and "body" tags respectively.

3.5 Database Storage for persistence

Once the voting is done for the web pages and links, we update the webpage and links table in the database. All extracted information is stored in persistent storage along with the matched concept for web page and its links. One advantage of our approach is that we have designed the project in such a manner that the all the web page and web link information are stored in the tables of our database. We crawl the web pages and load the tables in the database with the concepts. Then we follow the links to store its content and the relevant concept matches. Then propagation tables of the web pages and the links are updated along with the concept and their relation matches.

3.6 Propagation of concepts

Finally, the decision and propagation loop occurs. At this final step, the extracted information is analyzed again. Some of the extracted information may be deleted from page; some can be inferred or pushed from links to page. In this step, web pages are analyzed in network and we allow annotation flow between nodes. Both from page to describing link and from link to described page. This is an iterative process and in a few iteration the network reaches some stable (or near-to-stable) state. In such state, we say that the selected network is annotated and can be used in semantic navigation. Propagation of concepts and voting is discussed in section 5.

4. Approach

The approach will work in two general phases: Preparation and Annotation.

4.1 Preparation

Here a deep analysis of the Computer Science department in the University of Georgia will be conducted, resulting in building an ontology that represents the current structure of the department. This resulting ontology will be used in the next phase for annotation.

4.2 Annotation:

This is where the actual process of page and link annotation will take place. This phase is divided into three stages:

1) Page annotation
2) Link annotation
3) Relationship annotation

4.2.1 Page annotation

In this stage, all the pages in the Computer Science department site will be analyzed in one of the current methods, or a new method that we might need to develop. The result of this analysis will be a mapping between a certain page, and a node in the ontology designed in phase I.

4.2.2 Link annotation

Here, each page will be scanned for links that point to pages in the same domain, and each link will carry the annotation of the page it points to.

4.2.3 Relationship annotation

This is the final stage that defines which type of relationship the link defines. This relationship is obtained from the ontology based on the types (concepts) of the page with the link, and page the link points to. The resulted annotated pages will be stored in a database the application has access to write to and issue queries against.

5. Voting and Propagation

Voting for webpages, links and relationships are illustrated in algorithm 1, algorithm 2, and algorithm 3 respectively.

Algorithm 1 $Voting\ for\ Webpages$

```
for all tag entry in the configurator do
    for all concept in the ontology do
        Calculate the weights based on the number of matches
    end for
end for
Select the maximum concept weight among all the vectors
```

Algorithm 2 $Voting\ for\ Web\ links$

```
for all window size of the hypertext do
    for all concept in the ontology do
        Calculate the weights based on the number of matches for each concept and weight assigned to each of the individual anchor text sizes.
    end for
end for
Select the maximum concept weight among all the vectors
```

Algorithm 3 $Voting\ for\ Relations$

```
Given two concepts
Get all relations between the two concepts
Traverse all the concept nodes that are above a given two concept in the ontology and extract the relations between them
Match these relations with the text surrounding the hypertext window.
Choose the concept that has the maximum number of keyword matching among all the hypertext window vectors
```

The main drawback at this point with respect to relation voting is that we havenŠt concentrated on weighting relations between the concepts with different weights.

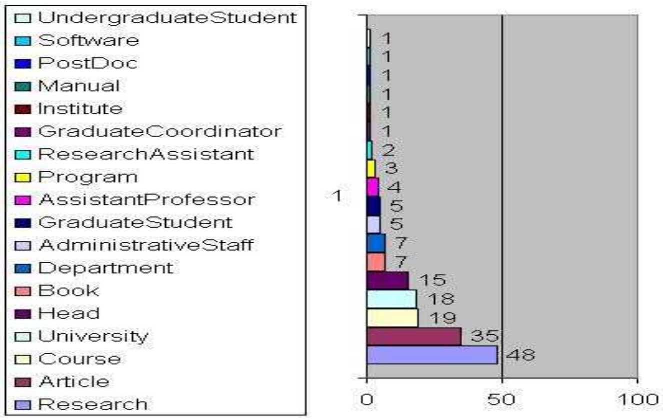


Fig. 3: Page concepts and number of matching pages

5.1 Propagation Algorithm

Propagation of concepts is done to increase the accuracy in the concept matches for a given web page and its links. Propagation is done after the voting stage. At the end of the voting we would have the concepts for web pages and its links stored in the database. Propagation of concepts are explained in algortihm 4:

Algorithm 4 *Propagation Algorithm*

```
Get the concept match to a page querying the database.
Query database and get the concept matching for the links pointing to a given page.
Propagate the concept from links to page if majority of links matches to a given concept.
If there is a tie, then propagate the concept from the webpage to the links. (Reason behind this is based on the intuition that webpage concept has higher priority than the concepts of the tied links.)
```

6. Testing and Experimental results

Testing was done on www.cs.uga.edu domain. Web crawler started crawling from http://lsdis.cs.uga.edu and was allowed to crawl only the CS domain. We ran our test crawling 200 web pages and the total number weblinks crawled were about 4599. We used the following test cases:

1) Finding concept for page
2) Finding concept for link
3) Propagating down concept, i.e. from a webpage to the the links pointing to it.
4) Propagating concept up, i.e. propogation occurs from the weblinks to the webpage.
5) Finding relation for a given link

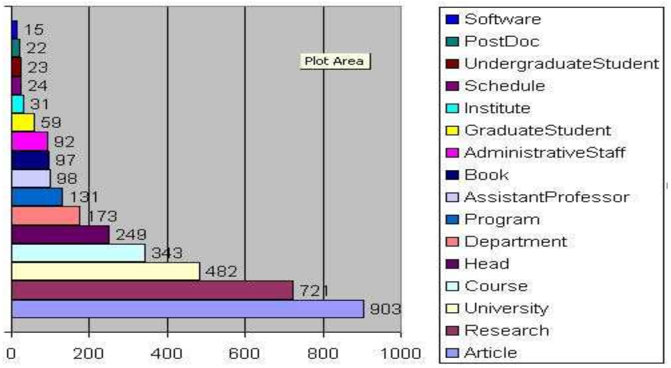
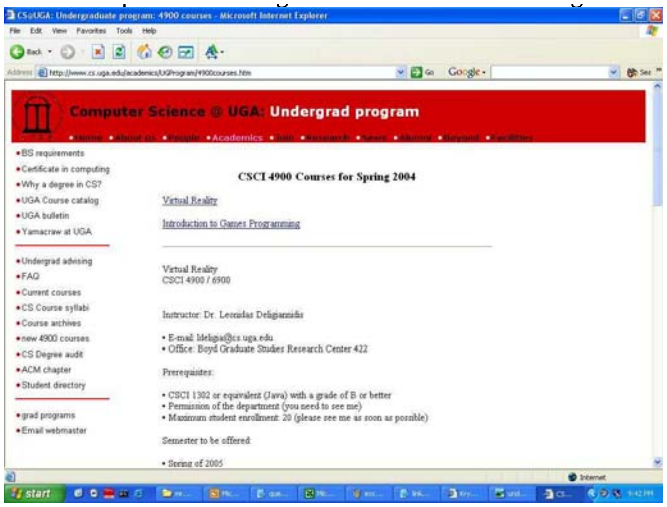


Fig. 4: Concepts and number of matching links


CS@UGA: Undergraduate program: 4900 courses - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Address http://www.cs.uga.edu/academics/UGProgram/4900courses.htm
Go
Google
Internet
start

Computer Science @ UGA: Undergrad program

•Home •About us •People •Academics •Join •Research •News •Alumni •Beyond •Facilities

•BS requirements
•Certificate in computing
•Why a degree in CS?
•UGA Course catalog
•UGA bulletin
•Yamacraw at UGA

•Undergrad advising
•FAQ
•Current courses
•CS Course syllabi
•Course archives
•new 4900 courses
•CS Degree audit
•ACM chapter
•Student directory

•grad programs
•Email webmaster

CSCI 4900 Courses for Spring 2004

Virtual Reality

Introduction to Games Programming

Virtual Reality
CSCI 4900 / 6900

Instructor: Dr. Leonidas Deligiannidis

- E-mail: ldeligia@cs.uga.edu
- Office: Boyd Graduate Studies Research Center 422

Prerequisites:

- CSCI 1302 or equivalent (Java) with a grade of B or better
- Permission of the department (you need to see me)
- Maximum student enrollment: 20 (please see me as soon as possible)

Semester to be offered:

- Spring of 2005

Example for the best concept match.
URL: http://www.cs.uga.edu/academics/UGProgram/4900courses.htm

Fig. 5: ..

Total concept matches for the pages crawled were about 174. Concepts were not matched to a page since there was not any concept label matching to the pages.

There were about 3463 links matching to a concept out of a total of 4599.

Some of the good examples for concept matching before propagation;

- concept: “AssistantProfessor” for “http://webster.cs.uga.edu/ budak/”
- concept: Research” for “http://webster.cs.uga.edu/ Ekochut/Research”
- concept: Article” for “http://lsdis.cs.uga.edu/Projects/METEOR-S/Downloads”
- concept: Department” for “http://www.cs.uga.edu”

Propogation of Concepts
Some of these webpages initially had “Research” as the page

concept. After propogating from links to the page we had the following results. Below are some of the examples of propogating concepts from links to the pages.

- "Department" for "http://www.cs.uga.edu/ jam"
- "Department" for "http://lsdis.cs.uga.edu/ devp"
- "GraduateStudent" for "http://lsdis.cs.uga.edu/ mperry"
- "GraduateStudent" for "http://lsdis.cs.uga.edu aleman"
- "Department" for "http://lsdis.cs.uga.edu/ cthomas"
- "Department" for "http://lsdis.cs.uga.edu/ kunal"
- "Department" for "http://lsdis.cs.uga.edu/ kaarthik"
- "TeachingAssistant" for "http://lsdis.cs.uga.edu/ mperry"

For Relations

- The page "http://lsdis.cs.uga.edu/about/index.php?page=1" matched to a concept "AdministrativeStaff" and "http://www.uga.edu" matched to concept "University"
- There was only one relation between them in the ontology. The relation found between these two concepts were "works"

Summary of our experimental test cases: In our experiment, first of all, we crawled 200 web pages. We set starting pages as http://lsdis.cs.uga.edu and http://www.cs.uga.edu. We limited the crawling area as in the "cs.uga.edu" domain. After crawling 200 pages, we crawled the links within each pages. Thus, the total number of link crawled were 4599. By running our algorithm on these 200 webpages, our algorithm assigned some concepts to 174 pages of them. Also running our algorithm on the 4599 weblinks, our algorithm was able to assign concepts to 3463 links.

Propagation algorithm used showed good results. Below is an example of its effectiveness.
The Computer Science web page had concept "Research" before propagation. By applying the propagation method considering 93 pointing links, original page concept "Research" has been changed as concept "Department". Concept was propagated from Links to Page.

7. Conclusion and Future Work

Our approach of using ontology labels for relations and concepts in ontology was very beneficial in concept matching. We were able to match most of the web pages to the concept in the ontology. Labels were represented by hypernyms, synonyms and homonyms. Our propagation algorithm showed excellent results. We were able to compare the effectiveness of the algorithm by comparing it with concept before propagation. Voting was done based on the importance of individual tags and also based on the importance of the various anchor text window sizes. Relation voting seemed to work pretty well. Relation voting was done whenever there were more than one relation matches between two concepts. Our algorithm or methodology could be changed by adding different weights to relations between concepts, i.e. we traverse the ontology tree to find all the possible relations between the concepts by traversing the tree in a bottom up fashion. Our algorithm uses these relations and matches the keywords around the anchor text window. Our future work is to include different weights to the relations as we traverse the tree in a bottom up fashion. We still need to tune the ontology as there are no concept matches for some of the web pages crawled. Using various label names to a given concept may not be the best idea compared to NLP techniques. Our ontology is not populated with instances so that we could use it for semantic web search. Initial experimental results were very promising, and we wish to work on this further.

References

[1] G. Pant, S. Bradshaw, F. Menczer, Search engine-crawler symbiosis: Adapting to community interests, in: ECDL, 2003, pp. 221–232.

[2] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, J. Kleinberg, Automatic resource compilation by analyzing hyperlink structure and associated text, in: IN PROCEEDINGS OF THE SEVENTH INTERNATIONAL WORLD WIDE WEB CONFERENCE, 1998, pp. 65–74.

[3] I. Varlamis, M. Vazirgiannis, Web document searching using enhanced hyperlink semantics based on xml, in: Proceedings of the International Database Engineering & Applications Symposium, IDEAS '01, IEEE Computer Society, Washington, DC, USA, 2001, pp. 34–43.

[4] G. Pant, Deriving link-context from html tag tree, in: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, DMKD '03, ACM, New York, NY, USA, 2003, pp. 49–55. doi:http://doi.acm.org/10.1145/882082.882094. URL `http://doi.acm.org/10.1145/882082.882094`

[5] L. Fraser, O. Fraser, C. Locatis, Effects of link annotations on search performance in layered and unlayered hierarchically organized information spaces, Journal of the American Society for Information Science and Technology 52.

[6] Semantic blogging and bibliography management, `http://www.w3.org/`.

[7] M.-M. Naing, E.-P. Lim, D. G. Hoe-Lian, Ontology-based web annotation framework for hyperlink structures, in: Proceedings of the Third International Conference on Web Information Systems Engineering (Workshops) - (WISEw'02), WISEW '02, IEEE Computer Society, Washington, DC, USA, 2002, pp. 184–. URL `http://portal.acm.org/citation.cfm?id=832313`

[8] P. Brusilovsky, J. Eklund, A study of user model based link annotation in educational hypermedia, Journal of Universal Computer Science 4 (1998) 429–448.

[9] M. Engelhardt, T. C. Schmidt, Semantic linking - a context-based approach to interactivity in hypermedia, Tech. Rep. cs.IR/0408001 (2004).

[10] S. Mukherjee, G. Yang, I. V. Ramakrishnan, Automatic annotation of content-rich html documents: Structural and semantic analysis, in: In Intl. Semantic Web Conf. (ISWC, 2003, pp. 533–549.

[11] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, S. Rajagopalan, A. Tomkins, J. A. Tomlin, , J. Y. Zien, J. Y. Zien, Semtag and seeker: Bootstrapping the semantic web via automated semantic annotation, ACM Press, 2003, pp. 178–186.

[12] S. Chakrabarti, B. E. Dom, S. R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, D. Gibson, J. Kleinberg, Mining the web's link structure, Computer 32 (1999) 60–67. doi:http://dx.doi.org/10.1109/2.781636. URL `http://dx.doi.org/10.1109/2.781636`

[13] M.-M. Naing, E.-P. Lim, R. H. L. Chiang, Extracting link chains of relationship instances from a web site, J. Am. Soc. Inf. Sci. Technol. 57 (2006) 1590–1605. doi:10.1002/asi.v57:12. URL `http://portal.acm.org/citation.cfm?id=1165013`

Relation-Centric Semantic Annotation using Semantic Role Labeling and Coreference Resolution

Chia-Hung Lin[1], Sheng-Hao Hung[2], Chien-Hsiang Liao[3]
[1] Department of Management Science, R.O.C. Military Academy, Taiwan
[2] Computer and Network Center, National Chi Nan University, Taiwan
[3] Department of Information Management, National Central University, Taiwan

Abstract - *Automatic semantic annotation based on domain-specific ontologies is a one of the critical issues for the success of the semantic web. Most existing approaches focused on the detection of concepts such as named entities, dates, monetary amounts. This study explores automatic semantic annotation techniques for applications using relation-centric ontologies which represent domain knowledge using a set of concepts with many inter-class relations. We propose a framework to detect event-based concepts and inter-concept relations using semantic role labeling and coreference resolution techniques. We gave an illustration of the processes by a semantic annotation application using CIDOC-CRM as the underlying ontology. Experiments using archives with a large number of image descriptions were conducted. The primitive results show that the accuracy is about 80% or so.*

Keywords: Semantic Web, Semantic Role Labeling, Semantic Annotation, Ontology, Coreference Resolution.

1 Introduction

The development of Semantic Web is an important step to facilitate the knowledge-based information integration across different resources. To create document annotations with well-defined semantics, the Semantic Web proposes annotating document contents based on domain ontologies [3] that often formally identify "concepts" and inter-concept "relations" between concepts in a specific domain. One major challenge in the semantic annotation is that the annotations by human are often laborious and error-prone. In the Semantic Web community, there is a thirst for technologies that perform the semantic annotation in more automatic manners.

One approach for automating the semantic annotation from free-text resources is the "string-matching" technique. Based on the string-matching technique, in the literature, a number of tools have been developed for Semantic Web applications. For example, Popov et al. [28] and Kiryakov et al. [18] proposed frameworks for semantic annotation, indexing and retrieval. They applied name entity recognition technique to detect a variety of knowledge such as named entity, money amount in sentences. Other similar works that rely on string matching techniques for retrieving knowledge includes Open Ontology Forge, COHSE annotator, Mnm, Melita, Parmenides, Armadillo, SmartWeb, PANKow, KIM, and Magpie [2; 4; 5; 8; 9; 10; 11; 13; 27; 28; 33]. An extensive review of relevant studies on semantic annotation can be found in Uren et al. [32]. Overall speaking, as concluded in [32], most of these semantic annotation systems are designed mainly to recognize "concept" instances and values from texts, but they often are not able to establish explicit "relations" between concepts. Hence, the directly applicability of the existing string-matching approaches in applications using "relation-centric" ontologies, which represent domain knowledge using a set of concepts with many inter-class relations, is questionable. One such real-life relation-centric ontology is CIDOC-CRM (Conceptual Reference Model) which is designed for cultural heritage applications and the only one having acquired the status of an International Standard [7].

The primary role of the CIDOC-CRM is to enable information exchange and integration between heterogeneous sources of cultural heritage information. It aims at providing the semantic definitions and clarifications needed to transform disparate, localised information sources into a coherent global resource. The CIDOC CRM contains classes and logical groups of relations (properties). Those relation groups are used to express facts regarding identification, classification, participation, structure and parthood, location, influence and motivation, assessment and reference. The CRM can describe the semantics of hundreds or more schema in use for museum object documentation with a small set of 90 concepts and 148 properties.

In the CRM, the essential knowledge in cultural heritage domain is structured as a semantic net with classes and the associated properties between classes. Most of the relationships people intuitively describe between classes are actually deductions from specific kinds of "events." For example, an *E67 Birth Event* comprises several relations to relevant concepts, such as event participants, time, and location (Figure 1).

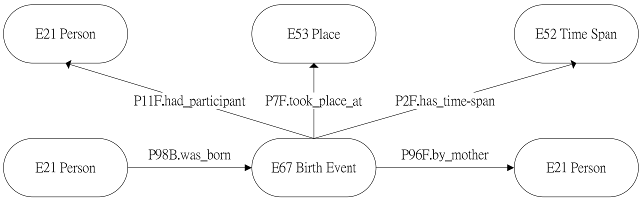


Figure 1: The CRM representation of a birth event

Considering the complexity of the CRM classes and relations, as in most Semantic Web applications, annotations by human based on the CRM are certainly laborious and error-prone. Current, there is a need for automatic semantic annotation techniques in cultural heritage communities. Unfortunately, existing string-matching based approaches fail to effectively identify inter-class relations that are required in CRM-based applications. The purpose of this study is to explore technologies that perform such a "relation-centric" semantic annotation in a more automatic manner. We propose mechanisms to retrieve event related knowledge from free-text resources using state-of-the-art natural language processing techniques. The objective is to automatically detect particular CRM-based event-related information, such as, subject, object, time and location in texts and map the knowledge into CRM instances for Semantic Web applications. In this following, the underlying principles and process for our methodology will be elaborated.

2 Methodology

In this paper, we propose a framework that adapt and integrate a number of state-of-the-art natural language techniques, including, lexical pattern-based knowledge retrieval, semantic role labeling, and coreference resolution techniques. In the first stage, a possible CRM event is obtained by matching the texts to a number of lexico-syntactic patterns that unambiguously represent the desired semantic relations. In the second stage, the core semantic roles, such as subject, object, location and temporal in the extracted sentences are obtained using semantic role labeling techniques. In the third stage, coreference techniques are applied to identify the exact entity the extracted subject or object of the sentence refers to. In the final stage, the coreference-resolved semantic roles are mapped into corresponding CRM instances based on a variety of mapping rules. In the following, we elaborate the underlying principles and the detailed operations of each stage.

2.1 Stage 1: sentence retrieval based on lexical-syntactic pattern matching

In principle, automatic discovery of a particular knowledge must start with a thorough investigation of the lexical terms and syntactic forms used to reliably express the desired semantic relation between entities. In practice, there are varieties of lexico-syntactic patterns that express a particular desired CRM event (e.g., creation, modification, destruction) describing a cultural artifact. To accumulate such syntactic patterns, we first come out with certain popular "seed events" which are expected to be widely described in abundance of web pages. Based on the seed events, appropriate query strings are formulated to query web search engines. The retrieved snippets from the web search results are investigated manually to identify the syntactic patterns that unambiguously indicated the desired event.

We take the "donation" event as an example to illustrate the syntactic pattern collection process. We first come out with a well-known event, for example, "France donates Statue of Liberty to U.S", as the seed events to query the web search engines. Examples of sentences retrieved that are used to explicitly refer to the knowledge come from the web search results at least include:

- France *donated* the Statue of Liberty to the USA
- It was in recognition of this that France *bequeathed* the Statue of Liberty to New York City in 1886.
- During the centennial France *offered* the Statue of Liberty as a gift.
- The people of France *presented* the Statue of Liberty to the minister of the United States in Paris.
- France *sent* the Statue of Liberty as a gift to the U.S. in order to celebrate.
- The people of France *gave* the Statue of Liberty to the people of the United States in 1886 in recognition of the friendship established during the American Revolution …

Based on the collected sentences, we realize a donation event possibly exist if a sentence contains phrases such as present, donate, bequeathed, "offered … as a gift", "send … as a gift", gave, etc. Raw sentences containing these listed terms will be candidates that possibly contain the desired event. These candidate sentences are collected and feed into semantic role labeling and coreference processor for distilling the knowledge desired. To increase the coverage of the query results, the query formulations are expanded by incorporating morphological variations such as verb tenses.

2.2 Stage 2: semantic role identification using semantic role labeling

Once the candidate sentences that contains a desired event that are included in CRM, the next stage is to identify the subject, object, location, and temporal information associated with the event. We use the semantic role labeling technique to achieve this goal. A general overview on the state-of-the-art semantic role labeling techniques can be found in [6; 14; 25]. Roughly speaking, in a sentence, a verb (predicate) indicates an event. The verb's syntactic arguments generally are associated with the participants of the event. A semantic role is the relationship that a syntactic argument has with the verb. One of the most commonly-used schemes for specifying the semantic roles are proposed to construct a large-scale corpus - the PropBank [17; 21]. In PropBank, the arguments of a verb are labeled sequentially from ARG0 to ARG5, where ARG0 is usually the subject of a transitive verb; ARG1, its direct object, etc. A variety of adjunctive arguments, such as ARGM-LOC, for locative, and ARGM-TMP, for temporal, are also tagged. Semantic role labeling techniques automatically identify the semantic roles of a sentence. Automatically tagging the semantic roles with high precisions is difficult since an event can often be referred using varieties of lexical items with different syntactic realizations. In the literature, there are a number of studies proposed different methodologies for such purpose, i.e., [14; 19; 24]. These methodologies have obtained well accurate results about 88% on ARG0, 82% on ARG1, and 70% on ARGM-LOC, ARGM-TMP, for sample data from Wall Street Journal [19].

In this study, the SRL technique is applied to obtain this fine-grained information associate with the event. As an example, consider a sentence in the description texts for the artifact "Tomb of Pope Paul III", give as:

In 1628 "Tomb of Pope Paul III" was modified by Bernini.

The SRL results for the given sentence is given as

[$_{ARGM\text{-}TMP}$ In 1628] [$_{ARG1}$ Tomb of Pope Paul III] [$_{Target}$ was modified] [$_{ARG0}$ by Bernini]

In such a case, the obtained semantic roles are actually ready for mapping to CRM instances. Nevertheless, as described below, there are many situations that coreference resolution techniques are required to give a serviceable annotation.

2.3 Stage 3: Coreference Resolution for semantic roles

In the free-text descriptions for artifacts in cultural archives, quite often a complete semantic relation is expressed in different contextual sentences. The coreference needs to be resolved automatically to identity which entity a noun phrase or pronoun actually refers to. For example, in a sentence give as:

In 1628 "Tomb of Pope Paul III" was modified by him.

The ARG0 in this case is given by a SRL tool as "him", which surely does not give a valuable knowledge when mapped to a CRM instance. In such case, the coreference techniques need to be applied to resolve the coreference so as to identify the exact roles implied in neighboring sentences. In linguistics, coreference occurs when different expressions in a sentence or contextual sentences refer to a same entity in real world. Two expressions (noun phrases or pronouns) are said to be co-referring to each other if both of them resolve to a unique entity (i.e., the referent) unambiguously. For example, in the sentences, "Leonardo da Vinci was one of the greatest painters of the Italian Renaissance. He left only a handful of completed paintings, among his works, the Mona Lisa is the most famous painting", the "Leonardo da Vinci" and "he" are most likely coreferent. Coreference resolution is the task of resolving noun phrases or pronouns to the entities that they refer to. It has been an active research topic in natural language processing for decades. The coreference resolution techniques are widely used in areas such as named entity extraction, question answering, machine translation and so on. In the literature, quite a number of methodologies have been proposed for solving the coreference resolution. Most early attempts heavily rely on linguistic and domain knowledge [15]. On the other hand, most recent approaches apply machine learning techniques with sophisticated syntactic parser and tagger, e.g., [16; 23; 26].

2.4 Stage 4: Mapping semantic roles to CRM instances

Once the semantic roles of a candidate sentence are extracted with coreference resolved, certain event-specific heuristic rules will be applied so as to correctly map the semantic roles into a sensible CRM instances. The rules are manually designed based on extensive investigations on the possible syntactic constituents of the sentences containing the event-trigger patterns. For example, in many sentences, certain heuristic rules are required to filter undesired relevant information about the person. For example, in the sentence "Fra Angelico, a famous painter, was born in Guido di Pietro", the parsed ARG0 is given as "Fra Angelico, a famous painter". In such a case, only the proper noun "Fra Angelico" is mapped to the CRM instance. The detections of proper noun can often be done using coreference tools.

3 Primitive Evaluations

We carried out an experiment to investigate the performances of the proposed methodology in real life cultural digital archives. First, a large set of images with textual descriptions are collected from a number of online archives, including Louvre Museum[20], Web Gallery of ART[34], Rijksmuseum[29], Manchester Art Gallery [22] and The Metropolitan Museum of ART[30].

A collection of 30,300 artifacts and 173,000 sentences were taken from the five archives. The average sentence numbers in a painting description is about 10. The average word number in a sentence is 22. The textual data are parsed, sentence-by-sentence, using a public available semantic role labeling engine- ASSERT [24]. The parsed semantic roles for each image are managed in a database. For the coreference resolution, we applid the "Gate tool" [12]. The approaches of Gate can be found in http://gate.ac.uk/. A list of the artist names given in ULAN [31] are feed to the Gate tool such that the proper nouns of persons can be successfully detected. A variety of heuristic rules have been manually designed so as to map the semantic roles in a sentence to a corresponding CRM event instances.

Table 1: A list of lexico-syntactic patterns used in the experiments

CRM Event	Lexico-syntactic patterns
E6 Destruction	was destroyed
	ruin
	caused damage to
E8 Acquisition	bequeath
	give
	offer
	send
	made a contribution of
	endow
	contribute
	donate
	made a donation of
E11 Modification	adapt
	modify
	make alteration in
	made an amendment to

For the evaluation experiment, we applied the proposed approaches to detect 3 core CRM events, including E6 Destruction, E8 Acquisition and E11 Modification. Table 1 lists the lexico-syntactic patterns that are used to access the target raw sentences. Table 2 lists the evaluation results. We measured the retrieval effectiveness by precision rate [1]. Precision is the number of relevant items retrieved as percentage of the total number of items retrieved. Precision will degrade by incorrectly relevant items. Hence, Precision is mainly to measure the ability of a system to present only relevant items.

$$precision = \frac{Number\ of\ relevant\ items\ retrieved}{Total\ number\ of\ items\ retrieved}$$

Table 2 shows the corresponding precision rate for three CRM events. The precision rates, ranging from 79 to 83 %, appear to be fair satisfactory. Based on the observation on those instances that are un-correctly mapped, the major source of errors was originated from the erroneous parsing result from the semantic role labeling tool applied. In the future, with the possible improvement on the state-of-the-art semantic role technologies, the proposed approach in the paper appears to be promising to get higher precision results.

Table 2: Evaluation results

CIDOC CRM Event	Precision
E6 Destruction Event	83%
E8 Acquisition Event	80%
E11 Modification Event	79%

4 Conclusions

In this paper, we proposed a methodology for automatically retrieving event-based knowledge for semantic annotation from texts. We applied state-of-the-art natural language techniques, including the semantic role labeling and coreference techniques to achieve the goal. We use a well-developed relation-centric ontology in cultural domain – CIDOC CRM to illustrate the semantic annotation process. The evaluation results show that the accuracy is rather satisfactory. The ease of implementation also indicates that the proposed methodologies can be easily realized using public-available resources.

5 Acknowledgement

This work was supported by the National Science Council of Taiwan (R.O.C.), under grant NSC 99-2410-H-145-002.

6 References

[1] Baeza-Yates, R. and Ribeiro-Neto, B. "Modern Information Retrieval." Addison-Wesley Longman Publication Co., Inc., Boston, MA, pp. 24-60, 1999.

[2] Bechhofer, S., Goble, C., Carr, L., Hall, W., Kampa, S. and Roure, D.D. "COHSE: Conceptual Open Hypermedia Service." In S. Handschuh, S. Staab (Eds.), Annotation for the Semantic Web, IOS Press, Amsterdam, 2003.

[3] Berners-Lee, T., Hendler, J. and Lassila, O. "The Semantic Web." Scientific American, 2001.

[4] Black, W.J., McNaught, J., Vasilakopoulos, A., Zervanou, K., Theodoulidis, B. and Rinaldi, F. "CAFETIERE Conceptual Annotations for Facts, Events, Terms, Individual Entities, and Relations." Parmenides Technical Report, TR-U4.3.1, 2005.

[5] Buitelaar, P. and Ramaka, S. "Unsupervised Ontology Based Semantic Tagging for Knowledge

Markup." In Proceedings of the Workshop on Learning in Web Search at the International Conference on Machine Learning, Bonn, Germany, 2005.

[6] Carreras, X. and Màrquez, L. "Introduction to the CoNLL-2005 shared task: Semantic Role Labeling." In Proceedings of the CoNLL-2005, pp. 152-164, 2005.

[7] CIDOC CRM, available at: http://cidoc.ics.forth.gr/, last visited Apr. 30, 2011.

[8] Cimiano, P., Handschuh, S. and Staab, S. "Towards the Self-Annotating Web." In Proceedings of the 13th International World Wide Web Conference (WWW 2004), New York, NY, 2004.

[9] Ciravegna, F., Chapman, S., Dingli, A. and Wilks, Y. "Learning to Harvest Information for the Semantic Web." In Proceedings of the First European Semantic Web Symposium, Heraklion, Greece, 2004.

[10] Ciravegna, F., Dingli, A., Petrelli, D. and Wilks, Y. "User-System Cooperation in Document Annotation Based on Information Extraction." In Proceedings of the 13th International Conference on Knowledge Engineering and KM (EKAW02), Spain, 2002.

[11] Collier, N., Kawazoe, A., Kitamoto, A., Wattarujeekrit, T., Mizuta, Y. and Mullen, A. "Integrating Deep and Shallow Semantic Structures in Open Ontology Forge." In Proceedings of the Special Interest Group on Semantic Web and Ontology, JSAI (Japanese Society for Artificial Intelligence), Vol. SIG-SWO-A402-05, 2004.

[12] Dimitrov, M., Bontcheva, K., Cunningham, H. and Maynard, D. "A Light-weight Approach to Coreference Resolution for Named Entities in Text." In Proceedings of the Fourth Discourse Anaphora and Anaphor Resolution Colloquium (DAARC), Lisbon, 2002.

[13] Dzbor, M., Motta, E. and Domingue, J. "Opening up Magpie via Semantic Services." In Proceedings of the 3rd International Semantic Web Conference, Hiroshima, Japan, 2004.

[14] Gildea, D. and Jurafsky, D. "Automatic Labeling of Semantic Roles." Computational Linguistics, Vol. 28, No. 3, pp. 245-288, 2002.

[15] Hobbs, J.R. "Resolving pronoun references." In Barbara J. Grosz, Karen Sparck-Jones, and Bonnie L. Webber, editors, Readings in Natural Language Processing, pp. 339-352, Morgan Kaufmann, Los Altos, California, 1986.

[16] Kehler, A., Appelt, D., Taylor, L. and Simma, A. "The (Non)utility of Predicate-argument Frequencies for Pronoun Interpretation." In Proceeding of 2004 North American chapter of the Association for Computational Linguistics annual meeting, pp. 289-296, 2004.

[17] Kingsbury, P. and Palmer, M. "From Treebank to PropBank." In Proceedings of the LREC, 2002.

[18] Kiryakov, A., Popov, B., Terziev, I., Manov, D. and Ognyanoff, D. "Semantic Annotation, Indexing, and Retrieval." Journal of Web Semantics, Vol. 2, No. 1, 2005.

[19] Koomen, P., Punyakanok, V., Roth, D. and Yih, W-T. "Generalized Inference with Multiple Semantic Role Labeling Systems." In Proceedings of the Ninth Conference on Computational Natural Language Learning, CoNLL-2005, pp. 181-184, 2005.

[20] Louvre Museum. available at: http://www.louvre.fr/, last visited Apr. 30, 2011.

[21] Marcus, M., Kim, G., Marcinkiewicz, M.A., MacIntyre, R., Bies, A. Ferguson, M., Katz, K. and Schasberger, B. "The Penn-treebank: Annotating Predicate Argument Structure." In ARPA Human Language Technology Workshop, 1994.

[22] Manchester Art Gallery. available at: http://www.manchestergalleries.org/, last visited Apr. 30, 2011.

[23] Ng, V. and Cardie, C. "Improving Machine Learning Approaches to Coreference Resolution." In Fortieth Anniversary Meeting of the Association for Computational Linguistics, ACL-02, pp. 104-111, 2002.

[24] Pradhan, S., Ward, W., Hacioglu, K., Martin, J. and Jurafsky, D. "Shallow Semantic Parsing Using Support Vector Machines." In Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics annual meeting, pp. 233-240, 2004.

[25] Pradhan, S., Hacioglu, K., Ward, W., Martin, J. and Jurafsky, D. "Semantic Role Parsing: Adding Semantic Structure to Unstructured Text." In Proceedings of the International Conference on Data Mining (ICDM 2003), 2003.

[26] Ponzetto, S.P. and Strube, M. "Semantic Role Labeling for Coreference Resolution." In Companion Volume of the Proceedings of the 11th Meeting of the European Chapter of the Association for Computational Linguistics, pp. 143-146, 2006.

[27] Popov, B., Kirayakov, A., Ognyanoff, D., Manov, D. and Kirilov, A. "KIM—A Semantic Platform for Information Extraction and Retrieval." Natural Language Engineering, Vol. 10, No. 3-4, pp. 375-392, 2004.

[28] Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A. and Goranov, M. "Towards Semantic Web Information Extraction." In Proceedings of the Human Language Technologies Workshop at 2nd International

Semantic Web Conference (ISWC2003), Florida, USA, 2003.

[29] Rijksmuseum. available at:
http://www.rijksmuseum.nl/ , last visited Apr. 30, 2011.

[30] The Metropolitan Museum of Art. available at:
http://www.metmuseum/.org, last visited Apr. 30, 2011.

[31] ULAN, Union List of Artist Names, available at:
http://www.getty.edu/research/conducting_research/vocabularies/ulan, last visited Apr. 30, 2011.

[32] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E. and Ciravegna, F. "Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art." Journal of Web Semantics: Science Services and Agents on the World Wide Web, pp. 14-28, 2005.

[33] Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A. and Ciravegna, F. "MnM: A Tool for Automatic Support on Semantic Markup." KMi Technical Report, TR No. 133, 2003.

[34] Web Gallery of Art. available at:
http://www.wga.hu/, last visited Apr. 30, 2011.

World-altering Semantic Web Services Discovery and Composition Techniques - A Survey

Hadi Saboohi[1], **and Sameem Abdul Kareem**[1]
[1]Department of Artificial Intelligence,
Faculty of Computer Science and Information Technology,
University of Malaya (UM), Kuala Lumpur, Malaysia

Abstract— *Semantic Web services evolved from traditional computational services by semantic descriptions. Recently, there have been many research efforts in the field of semantic Web services, which reveals enormous potential for Service-Oriented Architecture to be promoted to an improved architecture. However, world-altering services have been largely disregarded because of the limited facilities in current description languages to express required conditions. Enterprise Application Integration systems need world-altering services because most of the business services need preconditions to be held prior to their service execution. Moreover, they generate effects, both of which must be contemplated in the service environment. To exploit the semantic Web services in reality, efficient discovery and composition approaches need to be developed to complement the service environment requirements. This paper intends to overview selective methods for discovery and composition of world-altering semantic Web services.*

Keywords: Semantic Web Service, Discovery, Composition, World-altering Action, Precondition, Effect

1. Introduction

Service-Oriented Architecture (SOA) describes the abstract concept of interaction between the *service provider* and the *service consumer* through provision, discovery, and usage of services over the Internet. The provider introduces the core functionality (service interface) that will be utilized by requesters. The term "service" will be used henceforth to refer to the software engineering community, i.e., the computational parts of concrete services. The business community, on the other hand, designates the whole process, including actual interactions, as a service.

Service descriptions are published in a repository arranged by a third participant of SOA, called the *service broker*. The service broker also mediates negotiations between providers and consumers [1, chap. 2]. This mediation commences early on, particularly during **discovery** of the consumer's desired service. The mediator may also **compose** several services when prevailing atomic services are incapable of complying with the requester's demand. Currently, service brokers propose various **failure recovery** mechanisms (such as [2]) to provide resiliency for composition and execution tasks.

This paper explains selective approaches to service discovery and composition. The remaining sections give details on current discovery and composition techniques that explicitly consider world-altering services. Section 2 introduces semantic Web services, a categorization of services based on their actions, and a classification of their effects. Section 3 briefly presents some approaches to discovery and matchmaking of services. Section 4 gives an overview of exclusive service composition methods. The paper concludes with future plans in the final section.

2. Semantic Web Services

"Semantics" describes the formal meanings of functional and non-functional behaviors of services. Semantic Web services supplement traditional services with semantic specifications. The syntactic specification (WSDL[3]) employed by current practical services hinders automatic mediation at runtime. A study carried out by Lu et al. [4] shows that there are few actual services annotated by semantics, implying that semantic Web services were disregarded in the empirical study. Nevertheless, collections of semantic Web services are not difficult to find. SWS-TC, generated manually by Ganjisaffar and Saboohi[1], contains 241 semantic Web services, mostly real Web services. Additionally, OPOSSum[2][5] assembles data from SWS-TC and others to create an assemblage of *semantic Web services* (SWS) with different description languages – it presently contains over 1500 services [6].

Despite the creation of these collections, there are not enough semantically annotated services (in contrast with a very large, indeterminate number of existing services described syntactically) to accommodate requesters' needs, especially when requests are complex.

2.1 Service Actions and Terminology

There are two categories of services: *information-providing* and *world-altering* [7].

[1] Semantic Web services' test collection available at `http://www.semwebcentral.org/projects/sws-tc/`

[2] Online portal for semantic services available at `http://fusion.cs.uni-jena.de/OPOSSum/`

Information-providing services (also known as "information-generation" or "information-gathering" in the literature) are services that produce or gather information and generate *output*, usually based on *input* provided by the requester (i.e., they return information regarding the user's request).

From the agent's perspective, *information-providing services* have actions that only change the knowledge of the agent. These services sometimes require specific world states or conditions (called *preconditions*) to be held preceding their execution initiation time. These conditions are evaluated with respect to the client's environment before execution of the action [8] and guarantee the successful accomplishment of the services.

However, **world-altering services** (also known as "world transition") change the state of the world by their execution. In other words, thorough execution of the operation delivered by a service produces some valid facts about the world. This type of service can also have input, output, and preconditions. Moreover, world-altering services produce *effects* (the new state of the world) after their execution.

Finally, service descriptions may have a *post-condition* that identifies the input-output relationship along with conditions, both of which are evaluated in the server context [8] and guaranteed to be held over the output.

World-altering actions are used in ubiquitous (pervasive) computing, business-related services, interoperability among systems, and Enterprise Application Integration (EAI).

Sirin categorizes service effects as **world-altering effects** and **knowledge effects** [9]. In general, world-altering service actions and their accomplishment effects can arguably be classified as belonging to three families:

1) Service actions altering concrete objects in the world, such as shipment of products to customers.
2) Series of activities modifying "compensable information changing," such as data manipulation in a database.
3) Operations affecting "non-compensable information," such as accepting payment by credit card.

There are various semantic Web services' description languages declared by different groups with distinctive objectives. These languages include but are not restricted to OWL-S [10], [11] (formerly known as DAML-S[3]), WSML [12], SAWSDL [13], and DIANE [14], [15]. In this paper, we will presume some familiarity with these languages, hence their lack of presentation. This work is not restricted to any formalism, and all major languages have been investigated.

OWL-S is one of the major efforts to annotate services. In their latest release (1.2) [16], OWL-S Coalition added some other possible languages for indicating different conditions of services as compared with the previous releases. OWL-S now allows that preconditions and effects to be expressed in logic languages like KIF [17], DRS [18], SWRL [19], RDQL [20], and SPARQL [21].

WSMO (Web Service Modeling Ontology) [22] defines a model to describe semantic Web services, based on the conceptual design set up in the WSMF (Web Service Modeling Framework) [23]. Successive to the key aspects noticed in the Web Service Modeling Framework, WSMO distinguishes four top-level elements as the main concepts: Ontologies, Web services, Goals and Mediators. Moreover, WSML (Web Service Modeling Language) as a formal language is used to describe ontologies and Semantic Web services. WSML contains all aspects of Web service descriptions pinpointed by WSMO.

SAWSDL is evolved from WSDL-S [24] and takes a bottom-up approach, building on top of WSDL [25]. SAWSDL is the only semantic Web Service language which is a W3C Recommendation and even other major ones are still Member Submissions.

DIANE will be presented separately in the "Discovery and Matchmaking" approaches (Section 3).

2.2 A Survey

a) Problem Statement: In the literature, although there are thousands of approaches for discovery and composition of semantic services, most of them ignore the world-altering services and just use information services, due to factors such as simplicity or inefficiency of service description languages for expressing service pre/post-conditions and effects.

b) Significance of study: Considering preconditions and effects of services is crucial in various aspects of mediation. Different services may have the same input and output types and categorically diverse operations semantics [8].

To find appropriate atomic services or to construct them to generate a valid composition, services' preconditions and effects specifications help to a better ranking of candidates or a choice of the most accurate service.

Furthermore, to recover a service-based software application in case that a failure occurs, the mediator needs to undo service execution effects using "compensation needs" specified in service description even by calling another service to perform the restoration to the previous conditions.

Obviously, service discovery, composition, and failure recovery need to be done in an automatic manner. This need is due to increasing number of Web services, especially semantic Web services, emerging in today's computing world. Approaches such as [26] claim the finding of services by their preconditions and effects are not necessary by proposing the concept of manual tagging of services cannot help.

[3] DARPA agent markup language for services, `http://www.daml.org/services/daml-s/2001/05/`

3. Discovery and Matchmaking

Semantic service mediation begins with the finding of demanded service to carry out actual functionality. The user specifies the requirements, and a mediator tries to match them with a service description by a service matchmaking algorithm. Two closely related approaches use the algorithm. The matchmaking algorithm usually starts with situating a fully conformed service to the user's requirements based on capabilities. If such a service does not exist, the algorithm can identify all relevant services to construct a valid solution (cf. to Section 4) to fulfill the user's goal [27].

Matchmakers usually consider just inputs and outputs of services. Functional semantics, preconditions and effects of the services have to be regarded to find fully matched services. There are different services that have the same inputs and outputs, but with unrelated functionalities. Besides, other important, non-functional properties of services, such as quality of service, should be considered as well.

Authors in [28] argue that, in various aspects of service mediation, especially in Enterprise Application Integration (EAI), preconditions and effects must be scrutinized. They have fostered an extended matchmaking algorithm. This algorithm is used in the composition process of semantic Web services to pick out concrete services and substitute them instead of abstract sub-tasks. The extension adds preconditions and effects of service descriptions to the matching process as well as matching rules.

One point that is neither discussed nor even explicitly cleared in [28] is the way of finding the final degree of match of services. As [27] proposes, resulting matches are scored and sorted. Then the headmost service will be given to the requester. Adding preconditions and effects properties of services to inputs and outputs for matchmaking algorithm should clearly be stated as to how it affects this degree of match.

Pessoa et al. state in their recent survey [29] that, in the composition approaches studied, among all, METEOR-S [30] annotates service descriptions with preconditions and effects. Then these descriptions are used in service matching and selection, particularly in ranking of services.

Furthermore, MoSCoE [31] which uses OWL-S as service description language, considers preconditions and effects in addition to inputs and outputs in service discovery.

WSMF [23] also considers pre/post-conditions and effects in service description and dynamic binding of services at the runtime.

Authors of [32] propose a solution to discovery problems of *SWS-Challenge*[4]. They present DIANE Service Description (DSD) [15] as a language for describing semantic Web services along with a related matchmaking algorithm. The language is equipped with world-altering operations with one or more effects by various suggested elements, such as *operational elements* and *aggregational elements*.

One problem of DIANE is that there are not publicly available service descriptions in this language.

Authors of [33] present a precondition- and effect-enabled matchmaking algorithm for Web services using satisfiability checking of $\mathcal{SHOIN}^{+}(\mathbf{D})$ description logic reasoner. The algorithm's complexity claimed to be NExpTime-complete.

The approach presented in [34] proposes to use various degrees of matching for preconditions and effects along with input and output. They claim that the language which is possible to use for precondition and effect descriptions can be any of KIF [17] and PDDL [35]. They add the degree of matching of preconditions and effects, one level below the degree of matching of input and output. To match conditions between advertised conditions and queried conditions they define three phases. These phases include Parameter compatibility, condition equivalences and condition evaluation. For comparison purpose, they use both the concepts in conditions and operators.

One thing that is not clear in [34] is the translation method of preconditions and effects. In the OWLS-TC version they seem to use, there is no formal description of preconditions and effects, so they may translate the informal description of preconditions and effects and then use them in their algorithm. This is not specified in their experimental results. The way in which they interpret the operators for conditions is unclear as well.

In [8], authors propose the use of RDQL for services' preconditions and results descriptions. RDQL was a W3C submission[5] for RDF [36] data query language[6]. The goal (agent's goal) is also represented using RDQL query. Authors propose use of case reasoning for checking of applicability of the result. Results are claimed to be checked in the context of server and not the agent. They assume that the service is executed and that the results are available. Then they infer that the result conditions are true and add this new knowledge to the knowledge base and check the satisfiability of the goal. Therefore, the usefulness of the service is checked. Their focus is on *information providing* services, but because of the use of results (effects), the approach is also applicable to *world-altering* services.

The authors of [8] later evolved their work to [37] using SPARQL instead of RDQL. The approach is based on the use of SPARQL as the expression language of semantic Web services described in OWL-S. Preconditions, result conditions, and effects of OWL-S are modeled by SPARQL query forms[7]. The query form returns a RDF graph which describes the new world's state following process execution.

[4] Semantic Web Service Challenge: Evaluating Semantic Web Services Mediation, Choreography and Discovery (http://www.sws-challenge.org/)

[5] `http://www.w3.org/Submission/RDQL/`

[6] RDQL is now obsolete and replaced by SPARQL[21]. SPARQL is now a W3C Recommendation for RDF data query language

[7] They suggest the usage of SPARQL CONSTRUCT query form for a process result.

They claimed that the advantage of using SPARQL for this matter is the compactness of definitions of the process results and agent's goal. This effort primarily regards information services; however, they claim its applicability to world-altering services.

Authors of [38] used the same approach as [37] for SAWSDL description languages regarding service conditions and agent's goal. They have classified four semantic modeling aspects, namely Functional Semantics, Data Semantics, Non-functional Semantics, and Behavioral Semantics.

Bener et al. [39] proposed a matchmaking architecture to match Web services based on input and output descriptions and preconditions and effects rules. They have used SWRL as annotations for preconditions and effects. A test collection of 100 services described in OWL-S, including precondition and effect annotations in SWRL, has been created, and the architecture has been evaluated by 20 test queries. The result shows better precision at different recall levels for input, output, precondition, and effect matching in comparison with only input and output matching.

Authors in [40] use precondition and effect specifications equally with input and output signatures in their discovery approach. They use logical formulas for preconditions and effects. They claim that the approach is not restricted to any formalism. The language they have used for their implementation is WSML. The idea is to use different formalism for describing service offers and requests.

Therefore, matchmaking algorithms trying to find any functional match that satisfies user's specified goal need to take into account pre/post-conditions and effects of services, along with inputs, outputs, and non-functional properties such as quality, cost, or security.

4. Composition

Service composition generates a structure containing existing services and correlates them based on outputs, post-conditions, and effects of one service to inputs and preconditions of another service respectively. These services comprise the needed functionalities of the so-called composite service.

Composition approaches are differentiated as manual, semi-automatic, and automatic. Moreover, another characteristic of composition methods is binding time of actual constituent services, which can be static binding or dynamic binding.

As previously stated, in all subtasks of service composition, accounting for the specific features of world-altering services, like their preconditions and effects, is crucial in achieving a proper composed service.

Shin et al. [41] claim that, without specifying preconditions and effects of services, composers are unable to generate the correct service compositions, so functional semantics of services have to be respected.

In [42] authors use SWRL to represent functional properties, i.e., inputs, outputs, preconditions, and results, of services in OWL-S. Their study implies an encoding method of OWL-S atomic processes to semantic Web rules and SWRL, as well as use of them in a composition algorithm.

Hristoskova et al. in a recent study [43] introduce a Dynamic Composer, which constructs a service composition by matching preconditions of a service to effects of the previous service in a composition structure. This matching is claimed to be done similar to input-output matching. The Dynamic Composer also uses the approach in [37] to translate preconditions and effects to SPARQL.

Many publicly published research papers have been studied to investigate the capabilities of their proposed composition methods that use world-altering services. Among all, techniques shown here declare their approach to support both world-altering and information-providing services.

5. Conclusion

To the best of our knowledge, this is the first survey on discovery and composition of semantic Web services that clearly indicates world-altering category of services. In recent years, excessive researches have been conducted on the field of semantic Web services, but most of them only use information-providing services and ignore the existence of world-altering actions.

We are still investigating other published systematic ways to find existing world-altering semantic Web service discovery, composition, invocation, and monitoring and failure recovery methods. At the same time, evaluation of all recognized strategies is being conducted.

References

[1] D. Kuropka, P. Tröger, S. Staab, and M. Weske, Eds., *Semantic Service Provisioning*. Berlin: Springer, 2008.

[2] H. Saboohi, A. Amini, and H. Abolhassani, "Failure recovery of composite semantic web services using subgraph replacement," in *International Conference on Computer and Communication Engineering (ICCCE)*, Kuala Lumpur, Malaysia, May 2008, pp. 489–493.

[3] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," World Wide Web Consortium, W3C Note, March 2001. [Online]. Available: http://www.w3.org/TR/wsdl

[4] J. Lu, Y. Yu, D. Roy, and D. Saha, "Web service composition: A reality check," in *WISE*, ser. Lecture Notes in Computer Science, B. Benatallah, F. Casati, D. Georgakopoulos, C. Bartolini, W. Sadiq, and C. Godart, Eds., vol. 4831. Springer, 2007, pp. 523–532.

[5] U. Küster and B. König-Ries, "Towards standard test collections for the empirical evaluation of semantic web service approaches," *International Journal of Semantic Computing*, vol. 2, no. 3, pp. 381–402, 2008.

[6] H. Saboohi and S. Abdul Kareem, "A resemblance study of test collections for world-altering semantic web services," in *International Conference on Internet Computing and Web Services (ICICWS) in The International MultiConference of Engineers and Computer Scientists (IMECS)*, vol. I, International Association of Engineers. Hong Kong: Newswood Limited, March 2011, pp. 716–720.

[7] S. A. McIlraith, T. C. Son, and H. Zeng, "Semantic web services," *IEEE Intelligent Systems*, vol. 16, no. 2, pp. 46–53, 2001.

[8] M. L. Sbodio and C. Moulin, "Denotation of semantic web services operations through OWL-S," in *Workshop on Semantics for Web Services*, ser. CEUR Workshop Proceedings, M. O. Shafiq, Ed., vol. 316. Zurich, Switzerland: CEUR-WS.org, 2006.

[9] E. Sirin, "Combining description logic reasoning with AI planning for composition of web services," Ph.D. dissertation, Computer Science Department of University of Maryland, College Park, MD, USA, 2006, advisor-James Hendler.

[10] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. A. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "OWL-S: Semantic Markup for Web Services," Website, 2004. [Online]. Available: http://www.w3.org/Submission/OWL-S

[11] D. Martin, M. Paolucci, S. A. McIlraith, M. H. Burstein, D. V. McDermott, D. L. McGuinness, B. Parsia, T. R. Payne, M. Sabou, M. Solanki, N. Srinivasan, and K. P. Sycara, "Bringing semantics to web services: The OWL-S approach," in *First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC)*, San Diego, CA, USA, July 2004, pp. 26–42.

[12] J. d. Bruijn, C. Bussler, J. Domingue, D. Fensel, M. Hepp, U. Keller, M. Kifer, B. König-Ries, J. Kopecky, R. Lara, H. Lausen, E. Oren, A. Polleres, D. Roman, J. Scicluna, and M. Stollberg, "Web Service Modeling Ontology (WSMO)," Website, 2005. [Online]. Available: http://www.w3.org/Submission/WSMO/

[13] SAWSDL Working Group, "Semantic Annotations for WSDL and XML Schema," Website, 2007, http://www.w3.org/TR/sawsdl/.

[14] M. Klein, B. König-Ries, and M. Mussig, "What is needed for semantic service descriptions: A proposal for suitable language constructs," *International Journal of Web and Grid Services*, vol. 1, no. 3/4, pp. 328–364, 2005.

[15] U. Küster, B. König-Ries, M. Stern, and M. Klein, "DIANE: an integrated approach to automated service discovery, matchmaking and composition," in *16th international conference on World Wide Web (WWW)*. New York, NY, USA: ACM, 2007, pp. 1033–1042.

[16] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. A. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara, "OWL-S: Semantic markup for web services - Pre-Release 1.2," March 2006. [Online]. Available: http://www.ai.sri.com/daml/services/owl-s/1.2/

[17] M. R. Genesereth and R. E. Fikes, "Knowledge Interchange Format," 1998, draft proposed American National Standard (dpANS), NCITS.T2/98-004. [Online]. Available: http://logic.stanford.edu/kif/dpans.html

[18] D. McDermott, "DRS: A set of conventions for representing logical languages in RDF," January 2004. [Online]. Available: http://www.cs.yale.edu/homes/dvm/daml/DRSguide.pdf

[19] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, and M. Dean, "SWRL: A semantic web rule language combining OWL and RuleML," World Wide Web Consortium, W3C Member Submission, 2004. [Online]. Available: http://www.w3.org/Submission/SWRL

[20] A. Seaborne, "RDQL - a query language for RDF (member submission)," Hewlett-Packard, January 2004. [Online]. Available: http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/

[21] E. Prud'hommeaux and A. Seaborne, "SPARQL query language for RDF," Website, W3C, January 2008, http://www.w3.org/TR/rdf-sparql-query.

[22] D. Roman, U. Keller, H. Lausen, J. de Bruijn, R. Lara, M. Stollberg, A. Polleres, C. Feier, C. Bussler, and D. Fensel, "Web Service Modeling Ontology," *Appl. Ontol.*, vol. 1, no. 1, pp. 77–106, 2005.

[23] D. Fensel and C. Bussler, "The web service modeling framework WSMF," *Electronic Commerce Research and Applications*, vol. 1, no. 2, pp. 113–137, 2002.

[24] R. Akkiraju, J. Farrell, J. Miller, M. Nagarajan, M.-T. Schmidt, A. Sheth, and K. Verma, "Web Service Semantics - WSDL-S," World Wide Web Consortium, W3C Member Submission, November 2005. [Online]. Available: http://www.w3.org/Submission/WSDL-S/

[25] K. Verma and A. Sheth, "Semantically annotating a web service," *IEEE Internet Computing*, vol. 11, pp. 83–85, 2007.

[26] H. Meyer and M. Weske, "Light-weight semantic service annotations through tagging," in *Service-Oriented Computing - ICSOC 2006*, ser. Lecture Notes In Computer Science, A. Dan and W. Lamersdorf, Eds., vol. 4294. Springer, 2006, pp. 465–470.

[27] M. Paolucci, T. Kawamura, T. R. Payne, and K. P. Sycara, "Semantic matching of web services capabilities," in *First International Semantic Web Conference on The Semantic Web (ISWC)*. London, UK: Springer-Verlag, 2002, pp. 333–347.

[28] K. Zhang, R. Xu, Y. Zhang, Y. Sai, and X. Wang, "An ontology supported semantic web service composition method in Enterprise Application Integration," in *International Multi-Symposiums on Computer and Computational Sciences (IMSCCS)*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 222–227.

[29] R. M. Pessoa, E. Silva, M. v. Sinderen, D. A. C. Quartel, and L. F. Pires, "Enterprise interoperability with SOA: a survey of service composition approaches," in *12th Enterprise Distributed Object Computing Conference Workshops (EDOCW)*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 238–251.

[30] K. Verma, K. Gomadam, A. P. Sheth, J. A. Miller, and Z. Wu, "The METEOR-S approach for configuring and executing dynamic web processes," LSDIS Lab, University of Georgia, Athens, Georgia, USA, Tech. Rep., 2005.

[31] J. Pathak, S. Basu, and V. Honavar, "Modeling web services by iterative reformulation of functional and non-functional requirements," in *Service-Oriented Computing (ICSOC)*, ser. Lecture Notes In Computer Science, vol. 4294. Springer, 2006, pp. 314–326.

[32] U. Küster and B. König-Ries, "Semantic service discovery with DIANE service descriptions," in *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops (WI-IATW)*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 152–156.

[33] H. Wang, Z. Li, and L. Fan, "Capability matchmaking of semantic web services with preconditions and effects," in *Third Chinese Semantic Web Symposium (CSWS)*, August 2009.

[34] U. Bellur and H. Vadodaria, "On extending semantic matchmaking to include preconditions and effects," in *International Conference on Web Services (ICWS)*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 120–128.

[35] M. Ghallab, C. K. Isi, S. Penberthy, D. E. Smith, Y. Sun, and D. Weld, "PDDL - The Planning Domain Definition Language," CVC TR-98-003/DCS TR-1165, Yale Center for Computational Vision and Control, Tech. Rep., 1998.

[36] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax," World Wide Web Consortium, W3C Recommendation, February 2004. [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[37] M. L. Sbodio and C. Moulin, "SPARQL as an expression language for OWL-S," in *OWL-S: Experiences and Directions Workshop at 4th European Semantic Web Conference (ESWC)*, 2007, pp. 60–68.

[38] K. Iqbal, M. L. Sbodio, V. Peristeras, and G. Giuliani, "Semantic service discovery using SAWSDL and SPARQL," in *Fourth International Conference on Semantics, Knowledge and Grid (SKG)*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 205–212.

[39] A. B. Bener, V. Ozadali, and E. S. İlhan, "Semantic matchmaker with precondition and effect matching using SWRL," *Expert Systems with Applications*, vol. 36, no. 5, pp. 9371–9377, 2009.

[40] M. Junghans, S. Agarwal, and R. Studer, "Towards practical semantic web service discovery," in *The Semantic Web: Research and Applications*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2010, vol. 6089, pp. 15–29.

[41] D.-H. Shin, K.-H. Lee, and T. Suda, "Automated generation of composite web services based on functional semantics," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 7, no. 4, pp. 332–343, 2009.

[42] D. Redavid, L. Iannone, T. R. Payne, and G. Semeraro, "OWL-S atomic services composition with SWRL rules," in *ISMIS*, ser. Lecture Notes in Computer Science, A. An, S. Matwin, Z. W. Ras, and D. Slezak, Eds., vol. 4994. Springer, 2008, pp. 605–611.

[43] A. Hristoskova, B. Volckaert, and F. D. Turck, "Dynamic composition of semantically annotated web services through QoS-aware HTN planning algorithms," in *Fourth International Conference on Internet and Web Applications and Services (ICIW)*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 377–382.

SESSION

KNOWLEDGE DISCOVERY + ONTOLOGIES

Chair(s)

TBA

A Context Centric Model for Building a Knowledge Advantage Machine Based on Personal Ontology Patterns

Luyi Wang[1], Ramana Reddy[1], Sumitra Reddy[1] & Asesh Das[2]

[1]SIPLab, Lane Department of Computer Science & Electrical Engineering West Virginia University,
Morgantown, WV 26506, USA
[2]School of Business & Computer Technologies
Pennsylvania College of Technology (Penn State) Williamsport, PA 17701, USA

Abstract - *A knowledgeable person, in old times, has been regarded as a saint who knows everything without stepping out of house. Current technologies provide us with the ability of acquiring knowledge as one saint did before. Today, overloading of information always results in the quest for efficiently taking possession of useful information. With the advent of pervasive computing, information gathering is happening in real time. It has become possible that a collaborative knowledge network is constructed by utilizing little resource, but accurate under a carefully established environment. In this paper, we propose a new architecture, knowledge advantage machine (KaM), that helps people to construct a knowledge network. And we would like to apply a term, "call it once", into a novel type of knowledge discovery method. A typical User of KaM machine covers only one particular domain. The Domain information consists of a user defined ontology, a set of user collection of knowledge unit, we called them JANs, and some necessary tools to facilitate the knowledge discovery process. The "call it once" model transforms the traditional knowledge discovery process to a few basic simple steps (three steps).*

Keywords: knowledge discovery, knowledge network, knowledge advantage machine, personal ontology, artificial intelligence.

1 Introduction

The internet has indeed changed the habit of people searching of knowledge. We often talk on how Google or Facebook has changed our life. In reality however, knowledge itself is not provided by search engines. Only a consolidated body of related information is provided following a few key words. As a sequel, too much uncategorized information is delivered, causing wastage of time and frustration in finding a typical set of knowledge. For instance, people, who intent to learn programming on smart phones, may expect a body of knowledge different from those who are trying to buy a new phone. This obviously amounts to saying that knowledge discovery is not some straight forward disciplines, there are some other crucial factors affecting it. When portable devices have become prevalent, in a short time domain-specific, on-demand call for knowledge is a mandate. This brings the need for making newer knowledge discovery methods developed, which will be quick, to-the-point and will require little effort from the user. Such an expectation simply means that knowledge discovery process should be context-centric. It has been well-known that by monitoring and analyzing user behavior a detailed personal profile can be built, which however, often becomes very general and devoid of contextual bearing. This problem is overcome merging ontology and domain knowledge. A user ontology is regarded as speculations on concepts and relationships among user's particular domain taxonomy. To provide an on-demand information, this domain taxonomy with information can be investigated leading to providing one particular knowledge of a particular user's particular interest. In this paper, we are proposing a new architecture which is suitable for building up a user context centric knowledge network. The architecture is named as Knowledge Advantage Machine (KaM). The KaM is aimed in effectively helping people to find the right information with lesser effort. The paper is organized as follows. The section 2 narrates previous research in this line. The section 3 focuses on the concept of the KaM Machine and Vijjana model. The section four is about the current approach in constructing a user context centric mode. The section 5 concludes the work.

2 Related research

In the last two decades, people have spent an enormous of time and effort in the science and technologies of knowledge discovery. Gruber's 1993 work [1] on the techniques of developing ontology has become a classic research. It was demonstrated that ontology can be treated as agreements on knowledge sharing leading to reusable knowledge component and knowledge based services. In 1995 Chen and Ng [2] proposed two new algorithms on knowledge discovery using weight factors on knowledge relationships measuring similarities in between. In 2005, the concept of semantic desktop emerged which illustrated the technology for personal workspace and semantics-enabled desktop [3]. In 2006, semantic desktop 2.0 discussed about how to organize personal user's resources relying on metadata [4]. In 2008, people began to explore semantic web from mobile device perspectives [5]. In 1995, proposal [6] was made to build up profiles by monitoring user webpage browsing history. Starting from 1999, techniques on building personal profile started to categorize information with abstract structure. Proposals to generate user interest listing [7, 8], and methods of organizing information in to hierarchically evolving information evolved. Some leads on context awareness were introduced. Based on user feedbacks Researchers [9,10] tried to define what circumstances the user is in, and to categorize the user interests in to concepts and forms hierarchy. The use of Open Directory Project [11] as source ontology, and using the traditional vector space modeling methodologies were delivered [12].

3 KaM concept and the Vijjana model

In this paper, we proposed our knowledge discovery approach called the “Knowledge advantage Machine" which focuses on binding ontology pattern from user activities and discovering knowledge based on user context. “Knowledge advantage Machine" term is derived from “Mechanical advantage". The first two words, “Knowledge Advantage", is stressing the knowledge importance. In the industrial era, the mechanical advantage helped people to promote productivity in utilizing different mechanical tools. When time changed to the current era--the information era, the fast accelerating information processing power not only forced people to learn more but also required people to alter unuseful information. The last word “machine" relates to architecture. KaM consists of many components which work collaboratively. To address the context aware capacity, we would emphasize that our "KaM" should be applied only on one domain at one time. Here is how we introduced the KaM.

1. One KaM should be defined within a particular domain, enriching itself with domain information.
2. Within one domain, it can contain more than one KaM.
3. One KaM should reside in only one domain, but it can interactively communicate with other KaM residing on any other domain.
4. Ontology information of one KaM constitutes parts of ontology of the domain it resides in.
5. One KaM consists of at least one agent working on knowledge residing in it.

Here is a scenario to illustrate the aforesaid KaM architecture: The schedules for Jim on Tuesday is mostly depending upon the research Jim carrying on. He is writing one proposal for a certain science/technology funding. He communicates with professors via email in other universities to collaborate on the same proposal. All of them spend a great amount of time in discussing several innovative ideas that utilize some theories abstracted from biological phenomenology in the technology of computing. All these phenomena appear in our daily life. Researches from biological sciences summarize their findings into a formative theory system, which benefit distributed computing in many areas, such as resource utilization, load balancing and also clustering. The innovative ideas proposed by Jim are not new in biological sciences, but it is an adventure in computing. So Jim spends some time with other professors in discussing the adventurous nature of this new idea. In this scenario, there is a group of people involved. One is Jim and the other is his co-workers. The knowledge units reside in Jim’s research ontology is not enough to solve the problem. So he communicates with people within and outside his domain. When domain is expanded into much more detailed areas, there might be some crossing area shared within the domains. When we define the ontology for these domains, the ontology shared between them can be reasoned in both directions. Also the discussion conducted between Jim and other professors should cover the resources referenced in the common ontology. From the KaM perspective, we can see that the domain information is crucial for KaM. It covers the first and the second requirements of knowledge advantage. Meanwhile it calls for information which needs to be corrected and also targeted. The ontology information and agents provide the basis for satisfying the third requirement. It should be effectively reaching the required information in a real time manner. The efficiency requirement calls for all resources easy to distinguish and categorize. Meanwhile this architecture should be applicable for all resources. Based on this feature, we bring in the idea of JAN which is an abstract object for all the general resources. It provides an abstract layer above the resources to achieve the uniqueness for different users share on the same resources. The JAN object is constructed according to the IEEE LOM (learning object metadata) standard. To better organizing JAN, we use the Resource-Oriented Architecture (ROA) architecture as the resource infrastructure for the whole model. The ROA architecture not only allows us to neglect the issue caused by resource duplication, it also eliminates the issues caused by resource control. All operations supported in our ROA implementation are stateless, which support our distributed architecture when the knowledge network data storage expands in an exponential speed. Also the ROA resource naming strategy provides us a more effective way in consistency checking and resource organization. From user perspective, they would leverage knowledge more effectively, which demands that the KaM be aware of user context. In our approach, we defined the context by analyzing user ontology pattern. The user ontology construction process is relying on calculating similarity between user's JAN and taxonomies refereed from universal ontology. The foundation for this calculation is based on the phrases extracted from both the sources. we developed our own key phrase extraction algorithm, the VKE algorithm [13] which combines the statistical information and heuristic rules together. Once user ontology is constructed, we applied two methods to detect the user context. First is the timeline model, according to the distribution of user activities along the timeline, we selected the taxonomy with the highest probability as user context. Second is to monitor user behavior and generate the transition model upon user's interest score. We predicted user's context based on the conditional probability of taxonomy, with which the user may be moving on.

3.1 Vijjana model

Vijjana is the detailed implementation of KaM concept. It works for one domain with ontology information and facilitates with agents running on it on the purpose of acquiring knowledge. The vijjana model is defined as:

Vijjana-X = {J, T, R, dA, oA, cA, vA, sA, rA, CA},

where X = the Domain name;

J= the collection of knowledge units called Jans in the Vijjana-X;

T = the Taxonomy used for classification of Jans [knowledge units];

R= the domain Specific Relations;

d_A = the Discovery Agent which find relevant Jans; o_A = the Organizing Agent which interlinks the Jans based on R;

c_A = the Consistency / Completeness Agent;

v_A = the Visualization Agent,

s_A = the Search Agent,

r_A = the Rating Agent,

C_A = the Collaboration Agent.

In the Vijjana model, the discovery agent (d_A) helps user to gain new knowledge units through performing key feature extraction and comparison between universal ontology. Organizing Agent (o_A) and Consistency Agent (c_A) are responsible for finding the right user taxonomy to place the knowledge unit. Visualization Agent (v_A) construct the knowledge network from a users personal perspective or any global views and display them in several structured format such as radial view. Based on all of the above, we will present the KaM architecture with the Figure 1.

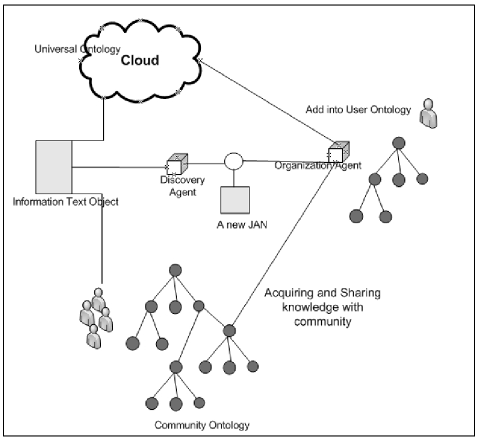


Figure 1: KaM Architecture

4. Our Implementation

4.1 Implementation: The hierarchy structure

Being different from other resources, knowledge usually is regarded as an abstract object. To better interpret that, people classify knowledge into different categories, which enrich knowledge unit with a hierarchy of information. Some resource websites, like open directory [11], already provides users a large amount of knowledge units with well classified ontology information hierarchy. This Meta information is based on the assumption of Resource-Oriented Architecture (ROA) [12] and also addressing a simple resource retrieval problem. Also, according to the Web 3.0 standard, knowledge unit as a type of information, should be retrieved with semantic relation, which to some extent, is extracted from information hierarchy.

4.2 Construct: User knowledge abstraction layer

The basic knowledge unit in the Vijjana, JAN, is abstracted from one of the text objects, like document, webpage, email and other information into a resource that can be reusable in the ROA architecture. By the definition of LOM, it contains important fields such as, annotations, references and also categories. These segmented information are helpful when dealing with knowledge alteration or recommendation. However this information should not be identical from different user's perspective. The annotations should not be simply a key word list generated by our Vijjana Key phrase extraction (VKE) algorithm; the references are also not a static list referencing other JANs; and moreover, one JAN may be mapped to more than more categories. Generally, a JAN is a representation of knowledge unit from a user. It contains meta information describing its resource, a conceptual meaning which is defined and adjusted by the user. In another words, a JAN is only meaningful when it is combined with user ontology information. So we need to have a user knowledge abstraction layer, which consists of JANs, and which are unique for different users

4.3 User profile based on ontology

Practically, a user always crosses over multiple domains. For instance, Professor Jim smith (as in section 3) has his domain in academic study. Besides this, he also has domains in personal life and others categories. Based on the KaM definition, one KaM resides only in one domain. For a particular domain, it owns ontology which defines its resource classification and relationship among them. Here the ontology information is not only the generalized classification cited from universal ontology, also is enriched with personal preference. Considering this, a user profile augmented with user ontology is necessary. A typical user profile contains two parts. One is the user's preferences and the other is the user's behavior rules. The prior part is mostly the user ontology, generalized from user's behavior history that records which web page the user browsed or what document the user read. From the content of the information object, an agent can generalize the classification and converge them into user preferences.

When a new information object comes in, the O_A in Vijjana framework will find a suitable category for it to reside in. This process is transformed into queries upon user's profile and the return result reflects the category information. In Vijjana model, the T set defines these concepts as taxonomies. For each JAN, its original resource content is extracted using VKE and a similarity test is performed along all user taxonomies to determine which is the most matched concept. If there is no concept in, which requires the probability should be over the threshold probability, then a consecutive similarity test is performed on the universal ontology. Based on [15], approximately 3,000 terms will cover all general concepts for a specific domain. A user, using finite taxonomies can cover all domains he/she crosses over.

4.3.1 Construct: User ontology

To construct user profile with ontology information, we used the Open Directory Project (ODP) as our referenced ontology. The ODP is regarded as the largest taxonomy store for web directory. The taxonomies are organized in hierarchy structure. In [12], authors concluded that only the first three levels of the taxonomy as references will promote the ontology hit accuracy. In Vijjana framework, we would also use the taxonomies in the first three levels as our concept set. For a taxonomy used in the universal ontology, we will train it with a collection of documents. For each trainable document, we preprocess it by stemming algorithm and extract phrases form it using our VKE algorithm. And then merge all key phrases into one vector in which each key phrase is distinct. In our VKE algorithm, phrases are evaluated by its entropy value. Here we used the phrase entropy value as its weight. When a new JAN is brought into vision, its reference document also needs to go through the stemming process and the key phrase extraction process. Once we have these two vectors, new vectors are generated from JAN's document and taxonomy vectors, and we need to apply the cosine similarity method to determine which taxonomy the JAN should reside in. Another issue of the method is: it loses the user preference since its weight is same as one in the taxonomy. Considering from user perspective, the keyword of JAN should be verified by user to strengthen the view of the user. Before we apply the cosine similarity, there is a normalization process to ensure the weight value is between 0 and 1. This step ensures that the final cosine value is between -1 to 1, where its absolute value closer to 1 stands for a higher similarity. For each word in both vectors, its weight w_i is calculated using following formula:

$$w_i = \frac{w_i}{\sqrt{\sum_{i=0}^{n} w_i^2}} \qquad (1)$$

The cosine similarity of vectors can be expressed as:

$$cosine(t_i, jan_j) = \sum w_{ik} * w_{jk} \qquad (2)$$

where, w_{ik} indicates the weight of term k appears in the taxonomy i and w_{jk} indicates the weight of term k appears in the new JAN j. For term that doesn't appear in either vector, the weight should be 0. The highest similarity taxonomy should be selected as part of user ontology.

4.3.2 User behavior model

Another part of user profile is the user's behavior model. For a finite number of taxonomy, we can construct finite rules to describe the transition between user activities. From the top level ontology, the rules are more generalized compared to detailed rules derived from the lower level. Also the next state in transition can also be predicted using a statistical probability model. For instance, for a user who has a strong interest in sports may more likely to spend more time in reading sports news rather than financial news after finishing his daily chores. So the transition on switching from work to news is more likely to choose the sports news as the end state. This behavior model is used in context awareness analysis. In order to evaluate the interest, we keep an interest score taxonomy hit number as its interest score. For each taxonomy, the interest score is calculated with

$$I_{t_i} = \frac{total\_hit\_number}{taxonomy\_size}, Where \qquad (3)$$

the I_{ti} stands for the taxonomy i, and taxonomy size is the number of JANs in this taxonomy. Using the Interest score we can construct the relationship between taxonomies. This is explained in the following section.

4.4 Context awareness

As illustrated above, a user behavior model is the base for the context awareness. To detect which context the user resides in, the basic method can fall into two categories: timeline and knowledge hits statistics. Before explaining our context awareness method, we need first to define what is a context. From its semantic meaning, a context should stand for: *where is the user*. In the KaM concept, we can define the context as the ontology the user resides in so that the context is defined with a main taxonomy as the current state and several sub-taxonomy as the possible next state. Context = {Current, {Next}}.

4.4.1 Timeline context awareness

In the Vijjana model, a JAN is contributed by the user. For each one, Vijjana records it with a timestamp as its submission time and revision time. On daily basis, for a particular user, if in certain time period there is a distinguishable increase of JAN submission or revision for certain taxonomy, then we can mark this time period with this taxonomy and correspondingly pick it as user's context for this time slot.

Using this method, one day is divided into two periods, user active period and inactive period. The basic time unit can be set as one hour. The inactive period is time units without any user activity. In opposite, user has activities during the active period. For a certain long enough time phase like one week or month, the user activity can be categorized by these two periods. For the user active period, we can cluster the taxonomy if we already know taxonomies the user owns. Based on the user activities for taxonomies, we can calculate the probability for each taxonomy on time phases, P (t_i|$time_j$), and then choose the highest one as the user's context. We can see this from the Figure 2

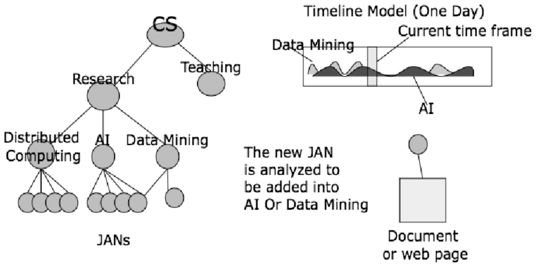


Figure 2: The Time Line model

4.4.2 Interest Driven Context Awareness

Another context awareness model has concerns on the taxonomy interest I_{ti} . For each concept, it has an interest score calculated. For all concepts, statistically each concept has factorial a value which is between zero to one to stand to support for its hit probability.

$$PI_{t_i} = \frac{I_{t_i}}{\sum I_{t_i}} \qquad (4)$$

By this interest score, we also can calculate the probability of certain transition to happen, which is a conditional probability of user moving from one taxonomy to another one, P (t_{next}|t_{prior}). The taxonomy with the largest probability stands for (more likely) user may be moving on, as the above example illustrated in user behavior model. According to the conditional probability, we formed a priority queue which stores the possible taxonomies ranked by their probabilities. The next state of context is selected from this queue. Meanwhile a *happened* transition will update the interest score and consequentially update PI_{ti}. Generally, for a small knowledge network with low average hit number, the update operation would not be costly. However when treating with knowledge network, the update should be done only when reaching hit percentage threshold, like 5%, which controls the update frequency.

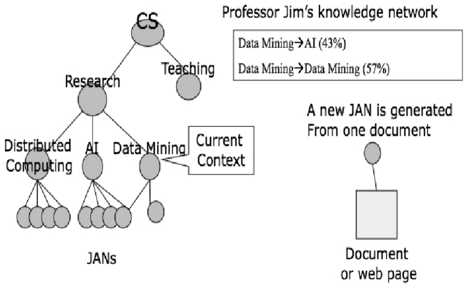


Figure 3: Interest driven model

4.5 Knowledge discovery process

One user, at any time context point, should reside in only one context. As defined before, the context provides the current taxonomy and next possible set of taxonomies. For each user, his/her ontology shares part of the universal ontology. By this feature, we can regard users who share the same ontology as a community. For instance, Professors doing research in computer science should share the ontology on computer science. For the communities, all share the universal ontology. This also leads to our knowledge discovery process into a three-step procedure. We call this procedure set as "*Call it once*". The discovery first happens locally in the user context, then expands to communities where the user resides in the same ontology, and then finally explores the universal cloud. The knowledge discovery can be initiated by a user in certain context, or by an agent during context switching. No matter in which way it goes, it is performed by queries upon taxonomy. A typical query is constituted by a set of phrases.

4.5.1 Knowledge discovery: Local

The local search is confined in user taxonomies. Within one taxonomy range, we can use phrase matching on taxonomy vocabulary. If it appears in the taxonomy vocabulary (keyword set), all JANs related to this keyword are returned, ordered by its weight. Otherwise, we need to look up its synonyms and process the searching procedure again.

4.5.2 Knowledge discovery: Community

Since people in the same community share the same ontology, so we can use the collaborative filtering (CF) technique to recommend JANs to users. Recall in the KaM model, in order to eliminate the problems for organizing and consistence checking, we applied the ROA architecture which required all items should be uniquely identified. For an original resource, it is abstracted in to JAN to add into user's knowledge network, which is unique in the whole knowledge network. So here we can't directly apply the CF technology upon the JANs. There are two methods for solving this problem. First, also a rudimentary one- is to use the JAN's referencing the original resource as item. The other is using keyword to replace the JAN as the comparison item. For the first method, we can construct the user-item matrix, in which item's value is the hit number of the JAN. Here it is:

$$UI_{ij} = \begin{cases} hitnumber_i & \text{if item i appears} \\ & \text{in } user_j\text{'s KN} \\ 0 & \text{Otherwise} \end{cases} \qquad (5)$$

Once we have this matrix, we can use the adjusted cosine similarity to compare two JANs. The JANs with highest similarity should catch the user's eye. For the second method, we use the keyword to replace the JAN, so the user-item matrix is formed as in the following expression. Here the keyword stands for a set of JANs which use this keyword as index.

$$UI_{ij} = \begin{cases} 1 & \text{if keyword i appears} \\ & \text{in } user_j\text{'s KN} \\ 0 & \text{Otherwise} \end{cases} \qquad (6)$$

We calculated the similarity between the two key-words and recommend were JANs indexed by the highest similarity keyword for the user.

4.5.3 Knowledge discovery: Universal

Since the second step of "call it once" has happened on community ontology which is also a part of the universal one, the final step is to search along with ontology which the community ontology relates to. These relationships are defined in the universal ontology and the search is already out of user's context. So here the query is without any user preference. For each related ontology, we performed a local search on all its taxonomies and returned JANs as a compensation of result from local search and community search. The overall discovery process can be viewed as the Figure. 4

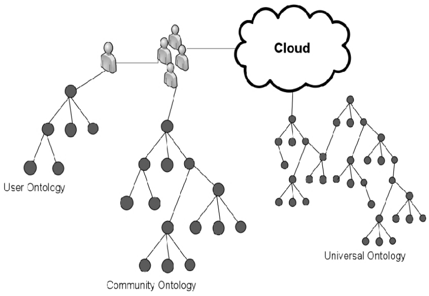


Figure 4: The whole discovery process

5 Conclusion

In this paper, we discussed the knowledge discovery problem coming with the current enormity of information overhead we encounter in web searching. To address this problem, we proposed the idea of the KaM, which is a new architecture designed to help people to effectively retrieve knowledge with ease and contextual support from peers. Inside the KaM, we created user ontology by using two well defined user context awareness models, one is a "timeline" model and the other is the "user interest driven" model. And then, we applied a novel knowledge discovery approach: "Call it once" to retrieve any desired knowledge. This approach transformed the discovery sequences into three basic steps: local, community and universal. The whole architecture is founded upon the ROA architecture, which ensured a high utilization of available resources.

6. References

[1] T. R. Gruber, Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In: International Journal Human-Computer Studies. Vol 43, p. 907-928.

[2] H. Chen and T. Ng, An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): symbolic branch-and-bound search vs. connectionist Hopfield net activation," Journal of the American Society for Information Science, vol. 46, pp. 348-369, 1995.

[3] http://www.deepamehta.de/.

[4] L. Sauermann, G. Aastr, M. Kiesel, H. Maus, D. Heim, D. Nadeem, B. Horak, and A. Dengel, A.: Semantic desktop 2.0: The Gnowsis Experience, in International Semantic Web Conference. Volume 4273 of Lecture Notes in Computer Science. Springer, 2006, pp. 887- 900.

[5] W. Woerndl and G. Groh, A social item filtering approach for a mobile semantic desktop application," 2008.

[6] H. Lieberman, Letizia: An agent that assists web browsing," in IJCAI, 1995, pp. 924-929.

[7] D. Mladenic, Text-learning and related intelligent agents: A survey," IEEE Intelligent Systems, vol. 14, no. 4, pp. 44-54, 1999.

[8] S. E. Middleton, N. R. Shadbolt, and D. C. De Roure, Capturing interest through inference and visualization: ontological user profiling in recommender systems," in K-CAP '03: New York, NY, 2003, pp 62-29

[9] C. C. Chen, M. C. Chen, and Y. Sun, Pva: A self-adaptive personal view agent, 2002.

[10] H. R. Kim and P. K. Chan, Learning implicit user interest hierarchy for context in personalization," in IUI '03: Proceedings of the 8th international conference on Intelligent user interfaces. New York, NY, USA: ACM, 2003, pp. 101-108.

[11] O. D. Project, http://dmoz.org," April 2002.

[12] L. Chen and K. Sycara, Webmate: a personal agent for browsing and searching," in AGENTS '98:Proceedings of the second international conference on Autonomous agents. New York, NY, USA: ACM, 1998, pp. 132-139.

[13] L. Wang, Vijjana key phrase generating algorithm, *unpublished: WVU internal document*.

[14] L. Richardson and S. Ruby, Restful Web Services. O'Reilly Media, 2007.

[15] H.M. Haav and T.L. Lubi, A survey of concept-based information retrieval tools on the web," in 5th E.European Conference, ADBIS 2001, Vilnius, Lithuania, 2001.

Building Ontology for the Political Domain

Sameera Al Shayji[1], Dr. Nahla El Zant El Kadhi[2], and Prof. Zidong Wong[3]
[2]Department of Management Information Systems, Ahlia University, Manama, Kingdom of Bahrain
[1,3]Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex, U.K

Abstract: *Ontology aims to define the semantics necessary for structuring and interrelating the stages and various activities of the deliberation processes with legal information and participating stockholders. The information can be taken from different domains. A common language is thus needed to describe such information that requires human knowledge for interpretation. Many applications have been developed [5, 16, 18] to provide and enhance delivery of services to citizens and businesses. However, little work has been done in building knowledge-based ontologies that facilitate communication, identify the processes and describe the data of these applications. This research focuses on developing prototype architecture for intelligent DSS that can help top political decision makers. In this paper, we propose this prototype architecture for generating ontology by extracting knowledge from various data sources. We propose to build an ontology using the Protégé-OWL editor to help political decision makers to strengthen bilateral economic relationships.*

Keywords*:* Fuzzy Logic, Protégé, Ontology, Fuzzy-Logic-Based Ontology, Governmental System.

1 Introduction

A lot of knowledge has been generated, organized and digitized in various governmental sectors, but it is still not readily accessible at any time or in one, convenient place for decision makers. Existing relationships between countries can be described from a variety of perspectives, such as historical, respectful, friendly, neighboring, traditional, religious, political and economical. Apart from such a variety of relationships, all nations seek to build bridges of cooperation with other countries in various ways. One way to build these relationships is to strengthen economic relationships, whereby the decision maker must take into consideration many factors and variables that influence the promotion of an economic relationship. This information and these factors are diversified and may be taken from different sectors. From the research viewpoint, the challenges lie in recognizing, finding and extracting these different variables. A conscientious decision-maker who takes responsibility for promoting and strengthening bilateral economic relationships needs access to well-structured information relevant to his/her decisions. Unfortunately, in reality, the actual information is unstructured, non-centric and scattered across different domains, including the political and investment domains. This makes it extremely difficult for the decision maker to understand the concepts, restraints and facts that exist in these domains. Due to the existence of various factors that influence decisions that aim to strengthen economic relationships with other countries, there is an urgent need to develop a proper system that analyzes the data gathered from different sectors and that produces precise and certain outputs that could be useful to the decision makers. In Kuwait, the scattered data mostly lies in various governmental sectors, including the Kuwait Fund for Development, the Kuwait Investment Authority, the Ministry of Foreign Affairs, the Prime Minister's Office, the Embassies of Kuwait and the Decision Maker's Office. Due to the various forms of political data existing in so many contrasting domains, certain imperfections such as imprecision, uncertainty and ambiguity appear. A popular way to handle scattered data is to construct fuzzy ontology, as presented in [20]. Ontology is useful for sharing knowledge, building consensus and building knowledge-based systems. Many projects of ontology are then implemented, such as the Semantic Web. The fundamental problem is to respect the diversity of languages and concept presentations of the world while encouraging the exchange of information. In this paper, we focus on proposing prototype architecture for generating ontology by extracting knowledge from various data sources. These sources may take on various forms, such as textual data, knowledge based data and regular documents.

2 Ontology in the E-government domain

In recent years, many countries have used ontology in e-government projects [4, 16]. Apostolou et al. [2] presented the OntoGov project, which aims to develop an ontology platform in order to facilitate the consistent configuration and reconfiguration of e-government services. More recent work in the field of ontology in governments was presented by Ortiz-Rodriguez [15]. They used a set of government methods of ontology to represent Mexican local government processes. Further work in ontology was conducted by Alexopoulos et al. [1] in order to detect fraud in e-government systems. Other methods of ontology have been built to facilitate transactions between companies across EU countries [8]. In addition, Salhofer et al. [17] described an approach to a model of ontologies for the e-government domain as a basis for an integrated e-government environment.

3 Methodology

In the literature, different methodology approaches for building ontology have been proposed [11, 3, 6]. Until now, there has been no standard method for building ontology. The approach described in this paper was adopted from Noy and McGuinness [13] and Fernandez-Lopez's [11] ontology modeling approach. Our ontology will cover the two main important government sectors in Kuwait: the Kuwait Investment Authority (KIA) and the Ministry of Foreign Affairs (MFA). It is important in the first step to know how these two sectors model and present their major trends, actions, norms and principles. It is crucial to describe the domains and the relationship between them and to understand the complexity involved in making decisions and how building ontology can be helpful and beneficial for decision makers. The second step consists of identifying the ontology concepts, including the definition of classes and subclasses, the properties between classes, the classes' shared elements and the description of entities within these classes. This will enable us to describe the domains and the relation between them. Ontology editors create and manipulate ontologies. Examples of such editing tools include Protégé, which is an ontology editor and knowledgebase framework, and Fuzzy Logic Toolbox, which extends the technical computing environment with tools that design systems based on fuzzy logic. Before defining these classes of ontology, we should determine and specify the domain that the ontology will cover and define its goal of use. The third step consists of listing the main terms that will be used in the ontology without considering any overlaps between them. In the fourth step, we choose an approach to define the classes and their hierarchy. There are two different approaches, the top down and the bottom up. This paper will follow the first approach. We start by defining the most general concepts and then add different specifications to those concepts. In the fifth step, we find the properties of classes and the slots, such as intrinsic, extrinsic and relationships between different members of the class. We should mention here that every subclass inherits all slots from superclasses. Step six consists of defining the facets of the slots, such as the cardinality, type, allowed values and instance with the relationship to another instance. Afterwards, we need to define the domain of the slot and the classes to which it is attached. The last step consists of creating the instances by choosing the class and filling in the slot values.

4 Purposes of the Ontology

One of the methods for determining the scope of an ontology is to write a list of questions to which an ontology-based knowledge should respond; such questions will be later subjected to the litmus test: Does ontology contain sufficient information to respond to this kind of question? Do the responses require a particular level of detail or the representation of a particular domain? Our approach consisted of building a set of questions that need to be answered by the ontology in order to fulfill their purposes. The concepts of the ontology are terms that define the domain or activities carried out in the domain [4]. Starting from this list of questions, the ontology includes information about the different elements and different types of conditions to be taken into account in order to make a recommendation about whether to invest or not invest in a specific country.

5 Existing tools to edit Ontologies

Ontology describes the concepts in the domain as well as the relationships that hold those concepts. Many existing tools are used to edit ontologies. 'Altova Semantic Works' is a visual RDF and OWL editor that auto-generates RDF/XML or nTriples based on visual ontology design, but no open source version is available. Different ontology languages provide different facilities. The most recent development in standard ontology language is OWL, from the World Wide Web Consortium (W3C). Like Protégé, OWL makes it possible for users to describe concepts but it also provides new facilities. It has a richer set of operators (e.g., intersection, union and negation). It is based on a different logical model, which makes it possible for concepts to be defined as well as described. Complex concepts can therefore build definitions out of simpler concepts. Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. The Protégé platform supports two main ways of modeling ontologies: via the Protégé-Frames editor or via the Protégé-OWL editor. Protégé ontologies can be exported into a variety of formats, including RDF(S), OWL and XML Schema. "http://www.mkbergman.com/904/listing-of-185-ontology-building-tools/" counted a total of 185 extant tools for editing ontologies. Noman et al. [14] have done a survey of existing ontology editing tools, and the comparison between them is presented in Table 1. The Protégé-OWL editor is used to build our ontology in the bilateral relationship domain.

6 Ontology and semantic relation

In this section, we will specify the practical aspects of drawing a class diagram. This diagram will include the necessary information of classes, such as identifying classes. Identifying classes is fundamental to object-oriented analysis. Through successive iteration, the dynamic interaction will be presented among classes. At this stage, it is important to identify and specify classes correctly. Class specification includes attributes, and each attribute has a different meaning. Common primitive data types include Boolean (true or false), character (any alphanumeric or special character), integer (whole numbers) and floating-point (decimal numbers). Figure 1 presents a diagram for Bilateral Trade Ontology with Semantic or Linguistic Relation.

Table 1: A comparison of ontology editing tools

Tools	Free	Open	Java	Extensibility	collaboration	Archtecture	Import	Export Languages	Tools
		Source	Based				Languages		
Protégé	√	√	√	√	No	Standalone	RDF(S),OWL	RDF(S),OWL,CLI	Pellet
OntoEdit(Free)	√	No	√	√	No	Standalone	RDF(S),DAML	RDF(S),DAML+OI	None
DOE	√	No	√	No	No	Standalone	RDF(S),OWL	RDF(S),DAML+OI	None
IsaViz	√	√	√	√	No	Standalone	RDF(S),N-	RDF(S),N-Triple	Jena
Ontolingua	√	No	No	No	√	Client Server	CLIPS, DAML+OIL	CLIPS, DAML+OIL	ATP
Altova SemanticWorks TM	No	No	No	No	No	Standalone	RDF(S),OWL	RDF(S),OWL	Built-in Reasoner

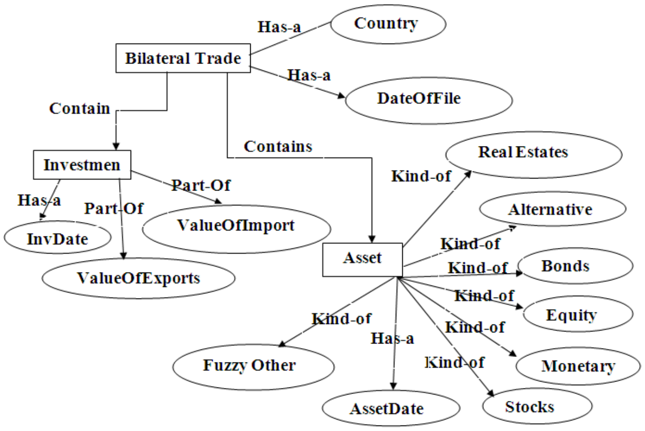


Figure 1: Bilateral trade ontology with semantic Relations.

7 Fuzzy Set and Membership

The aim of this section is to present a proposal that integrates fuzzy logic in ontology. Undoubtedly, the success of fuzzy logic applications describes vague information in addition to drawing our attention to addressing certain applications in government sectors, since such information needs a common language to describe its concept.

Fuzzy set and fuzzy logic allows users to model imprecise and vague data. Fuzzy logic can combine different priority functions. Fuzzy logic allows any value between 1 and 0 as a logic value. Fuzzy logic is based on natural languages in order to provide convenient methodologies to represent human knowledge [12]. The fuzzy membership value μ is used for the relationship between the objects in question, where 0‹μ‹1, and μ correspond to a fuzzy membership relation, such as “low,” “medium” or “high,” for each object. The purpose of fuzzy control is to influence the behavior of a system by changing the inputs or outputs to that system according to a rule or set of rules under which the system operates.

In the first step of our research, we extend the domain’s ontology to generate fuzzy ontology. This fuzzy concept includes a set of membership degrees in each concept: the political and investment domains with the relationships between them. The first step will provide a complete framework based on ontology in a particular domain. The proposed bilateral relation domain ontology contains vague descriptions. Table 2 presents different classes with different properties in the bilateral relation domain. For example, “StrongFriend” is a property of the concept and “RelationName” describes the name of a relation in the bilateral relation domain. Thus, “StrongFriendRespect,” “WeakRespect,” “Respect” and “StrongFriend” are properties that describe the type of relation between two countries, which require human knowledge for interpretation. The second step consists of adding a degree of membership to all terms in the ontology without overloading the problem. The third step consists of generating an extension of the domain ontology with a fuzzy concept [7]. Table 2 presents the first step in the political approach, which extends the domain’s ontology into a bilateral relationship domain in order to generate fuzzy ontology. Some concepts included in this domain are “CountryClassification,” “CountryName” and “RelationName,” where the “CountryName” class describes different classifications of countries, the “RelationName” class describes the type of relation between two countries and the “CountryName” class includes groups of different countries. A concept is considered to be a class in ontology with a set of properties in the bilateral relationship domain. In addition, the political decision-maker should consider various forms of information, such as textual data, knowledge base and regular documents. Certainly, extracting knowledge from various data sources can be described by a common language that requires human knowledge. For example, “coalition countries” includes all the states that had a positive attitude toward the State of Kuwait during the Iraqi invasion. This means that “coalition countries” represent the countries that condemned Iraq’s invasion, participated in the coalition forces and participated in reconstructing the country after the Liberation (see Table 3). Such definitions are used for the decision making process when it comes to strengthening the bilateral economic relationships between Kuwait and other nations. Such decisions are influenced by certain definitions and well defined concepts. In addition, different criteria for certain factors and variables are not described by degrees of interval [0.1] but described by linguistic terms. For example, to describe the concept of the “existing bilateral relation” as classes between countries, it can be described from a variety of perspectives with a set of properties, including “historical,” “respectable,” “coalition countries,” “antibody states” and “friendly.”

Table 2: Examples of semantic relations in “CountryClassification” and “RelationName” classes.

Country Classification	Relation name	Country name
Coalition countries	Strong-Friend-Respect	a b c d e f
sectarian States	Respect	J k l
investment states	Strong-friend	b c d ea g y t
Arab states	Respect-culture	a b x b p k d
EU states	Strong-respect-friend	a b c d r
GCC	History-neighbour- Religion	A b d f c
States voted in favour of the issue of Kuwait	Encourage very strong	A b c e
Crisis States	weak	G w

These properties cannot be evaluated and are sometimes even immeasurable. A commons langue is thus needed to describe such properties that require human knowledge for interpretation. For example, “coalition countries” is a property of the concept “existing relationship.” The value of the “country classification” class, such as “coalition countries,” has a fuzzy concept. Its link with the linguistic “RelationName” property is also a fuzzy concept. This does not help the decision maker to measure the “RelationName” fuzzy concept’s link with another fuzzy concept. This makes it extremely difficult for the decision maker to understand the concepts, restraints and facts. A decision support process must be empirical in order for the decision maker to assess the different fuzzy factors, fuzzy variables and the relationship between them in order to reach proper decisions. Therefore, coalition countries have more investments than other countries but are the measurement for other classifications. Examples of different factors and variables that may be assessed when defining the “coalition countries” are presented in Table 3, which illustrates the positions of states towards the issue of Kuwait in front of the United Nations. It includes the vote on the resolutions of the Security Council in the United Nations, such as the vote on the resolution of human rights in Kuwait during the Iraq invasion, etc. The “Coalition countries” class includes linguistic terms such as “Agree,” “Abstention,” “Disagree,” etc. Most factors and variables that are described are extracted from the political domain. Correspondingly, there are many existing variables in the investment field. Certain variables have a direct impact on strengthening the economic bilateral relationship-fuzzy concept, such as “prevent” or “reduce.” These variables also have an impact on the political bilateral relationship. They cannot be evaluated, because such inputs are very inaccurate and need human interpretation. The existing information includes linguistic variables for the evaluations. This linguistic variable can be proposed by expert rules and fuzzy inference for the decision making process.

Table 3: The “coalition countries” class by generating different subclasses

Year	Voted the resolution of human rights in Iraq	Voted the resolution of human rights in Kuwait	Voted the resolution of implication on the	Demanded the withdrawal of foreign forces from the region	Called for the lifting of economic embargo	Addressed the issue of Kuwait
1990		Absent		Agree		No
1991	Abstention	Agree	Absent			No
1992	Abstention		Absent			No
1993	Disagree			Agree	Agree	
1994	Disagree			Agree	Agree	
1995	Disagree			Agree	Agree	
1996	Disagree			Agree	Agree	
1997	Disagree			Agree	Agree	
1998	Disagree			Agree	Agree	
1999	Disagree			Agree	Agree	
2001	Disagree					

It is difficult for the decision maker to understand the dimensions of these linguistic variables while deciding to strengthen bilateral economic relations with this country. Identifying those variables related to this definition would enhance many decisions. Achieving the integration of information with rich concepts undoubtedly helps the political decision maker in making the appropriate and correct decisions. We propose to use ontology to integrate these scattered data from political and investment domains by extracting key concepts and relations between sets of information and by integrating fuzzy logic with ontology to obtain a solution that is more suitable for solving the uncertainty of problems in these intelligent decision support systems. In the first step, we need to break down the concept of the investment indicator. The “InvestmentIndicatorName” class has different properties that can be described from a variety of perspectives, such as “encourage,” “limit,” “prevent,” etc. (see Table 4).

Table 4: Fuzzy values assigned to “InvestmentRelation” class in the bilateral relationship domain.

Investment Relation	Weight
Encourage	0.2
Limit	0.4
Prevent	0.3
Encourage with strong	0.8
Caution	0.6
Warned	0.7
Opportunity	0.5

Therefore, a need emerges for giving different interpretations according to the context. Table 5 presents the proposed “InvestmentIndicatorName” class with linguistic and semantic properties.

Table 5: Fuzzy logic assigned to “CountryName” and “InvestmentIndicator.”

Country name	Relation name in bilateral relation domain "Investment Indicator"
A	Encourage very strong
B	Weak
C	encourage
D	prevent
F	Caution
E	carful
F	Encourage with caution

8 Fuzzy ontology structures

An ontology can be converted into fuzzy ontology by adding the relation weight to any fuzzy relation, as presented in [12, 19]. This ontology includes the weight for every relation (see Figure 2).

9 Case studies methodology

In this research, we introduce a fuzzy ontology approach and apply this approach to two main important government sector representatives in Kuwait: the Kuwait Investment Authority and the Ministry of Foreign Affairs. It is very important to understand how these sectors presented their major trends and broke their concepts down into objectives, actions, norms and principles. This will help us to identify the appropriate ontology concepts, including classes and subclasses, to characterize the properties between classes, to share all elements, to describe the entities in those classes and to explain the domain and the relation between them. The aim of conducting the fuzzy ontology approach is to provide insight into how knowledge can be represented and handled so that the decision maker has support from an intelligent decision process.

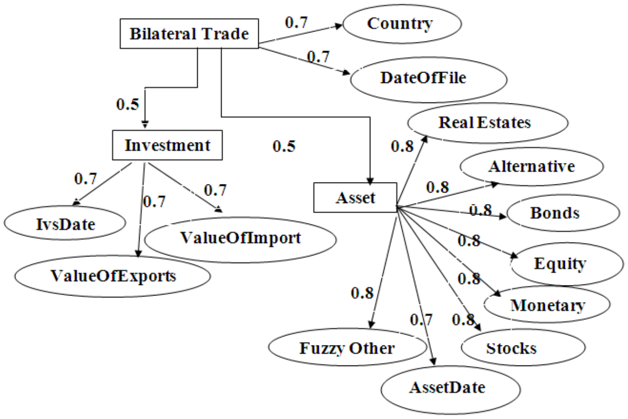


Figure 2: Ontology representing bilateral trade domain.

Figure 3 presents the Ontology diagram to explain the relation between the Ministry for Foreign Affairs and the Kuwait Investment Authority. We can see in this figure different links between the different classes of the ontology. For example, the “MinistryOfForeignAffairs” class is directly linked to the “BilateralRelationship” class. The “BilateralRelationship” classes have different attributes, such as “StartDate,” “BilateralRelationType,” etc. The “MinistryOfForeignAffairs” class is subdivided into the “FuzzyIranNuclearFile” class, “FuzzyIraqiAffair” class and the “FuzzyPalestinianIsrael” class. In addition, “MinistryOfForeign Affairs” has strong links, relations and influences on the activity of the “KuwaitInvestmentAuthority.” The relationship of trade in the “KuwaitInvestmentAuthority” has the different attributes of “ValueOfImport,” “ValueOfExport,” “ValueOfAsistance,” “ValueOfGrants,” and “LoanValue.” On the other hand, the type of relationship between the two countries has an impact on the continuity of the loan. The Ministry of Foreign Affairs handles the workflow for multiple files, such as Iraqi affairs, Iranian affairs and Palestinian affairs. The answers to these functioning files usually take the form of “yes,” “no,” “strong,” “very strong,” etc. In this context, we propose prototype architecture for an intelligent decision support systems that can help top political decision makers to strengthen bilateral economic relationships. We present the integration of data across different sectors and produce a seamless system that enables valid design support for top political decision makers by employing natural language. Figures 3 and 4 show the semantic relation and fuzzy ontology for political and investment sectors.

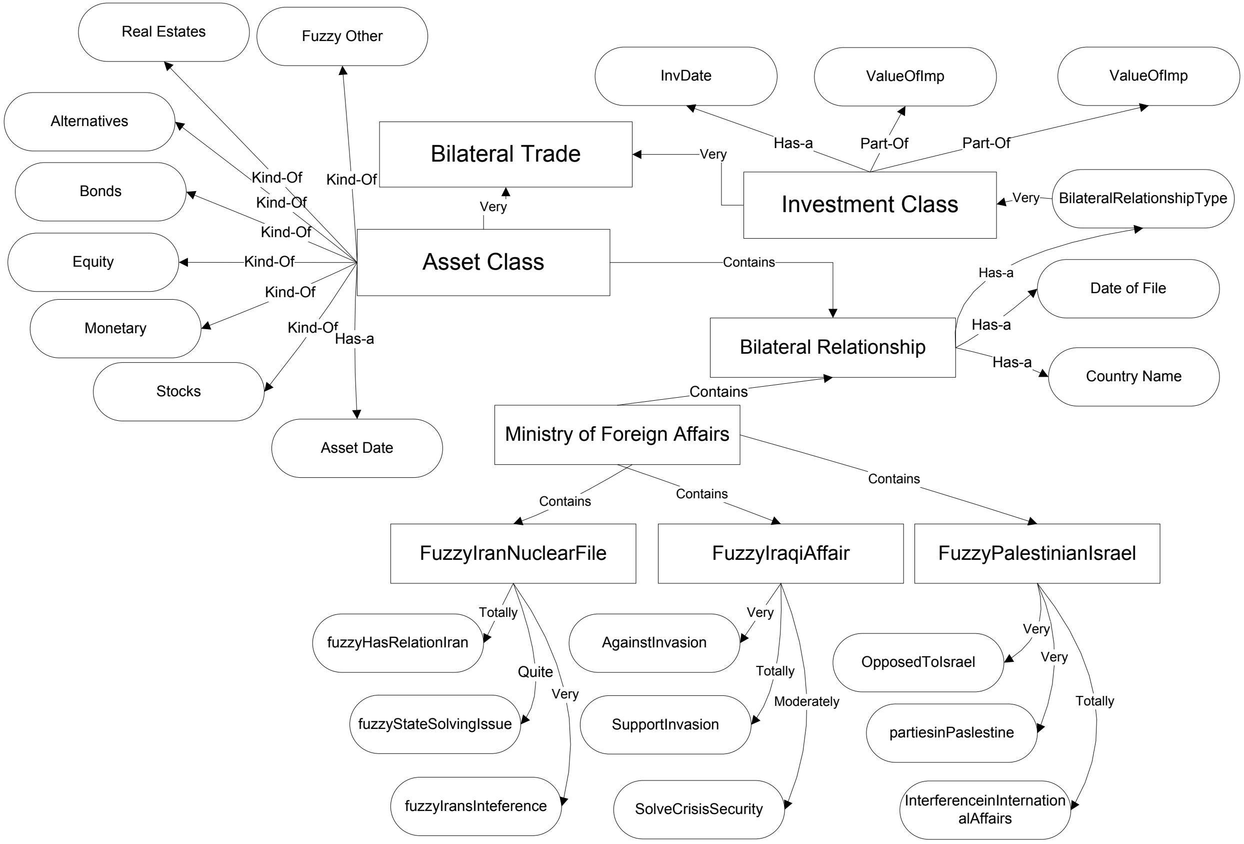


Figure 3: Semantic ontology for the relation between the Ministry for Foreign Affairs and the Kuwait Investment Authority.

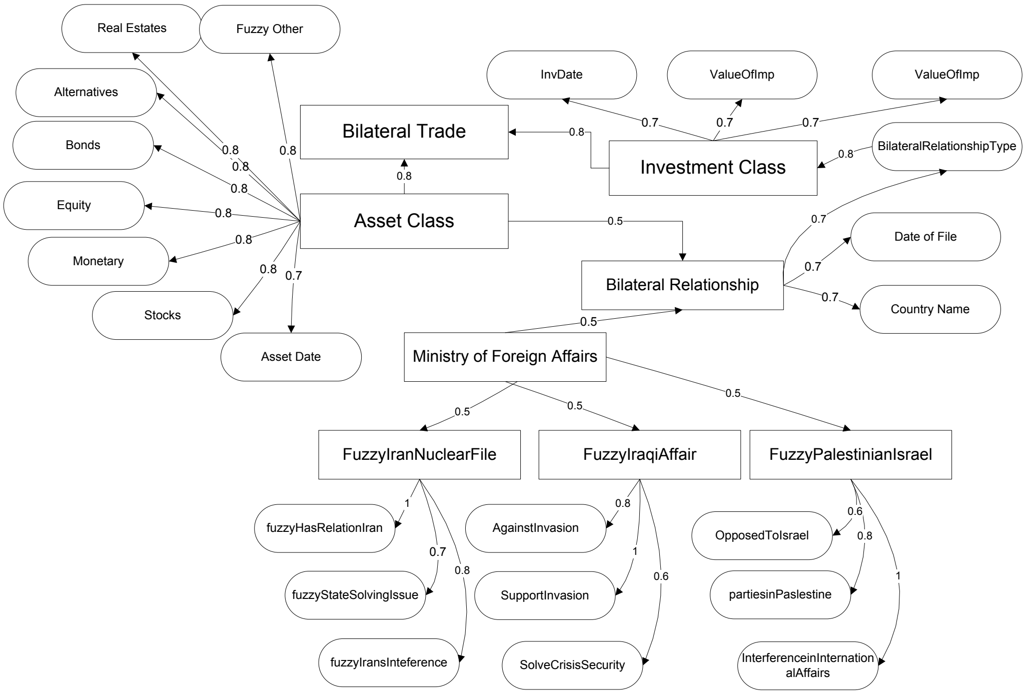


Figure 4: Fuzzy ontology for the relation between the Ministry for Foreign Affairs and the Kuwait Investment Authority.

10 Conclusion

In this paper, we propose a methodology to develop a fuzzy ontology approach and discuss how to conduct this approach in two main important government sectors in Kuwait: the Kuwait Investment Authority and the Ministry of Foreign Affairs. To build this ontology, it is very important to understand how these sectors represent their major trends by breaking these sectors down into objectives, actions, norms and principles. This helps to identify the proper ontology concepts for each sector, to characterize the properties between them, their sharing elements, the entities in those classes and the domain and the relationships between them. The aim of conducting the fuzzy ontology approach is to provide insight into how knowledge can be represented and handled in order to provide the decision maker with aid from an intelligent decision process.

11 References

[1] Alexopoulos, P., Kafentzis, K., Benetou, X., Tagaris, T., and Georgolios, P. 2008. Towards a generic fraud ontology in e-government. In Proceedings of the International Conference on Security and Cryptography, Portugal, June 2008.

[2] Apostolou, D., Stojanovic, L., Lobo, T.P., Miro, J.C., and Papadakis, A. 2005. Configuring e-government services using ontologies. IFIP International Federation for Information Processing, Springer Boston, Vol.189, 141-155.

[3] Beck, H., and Pinto, H.S. 2003. Overview of approach, methodologies, standards, and tools for ontologies. Agricultural Ontology Service (UN FAO).

[4] Bettahar, F., Moulin, C., and Barthes, J.P. Ontologies supporting e-government services. In Proceedings of the IEEE Artificial Intelligence Conference, Portugal, Dec 2005, 1000-1005.

[5] Bwalya, K.J. 2009. Factors affecting adoption of e-government in Zambia. The Electronic Journal on Information Systems in Developing Countries, Vol.38, 1-13.

[6] Calero, C., Ruiz, F., and Piattini, M. 2006. Ontologies for software engineering and software technology. Calero, Ruiz, Piattini (Eds.), Springer-Verlag, Berlin, Heidelberg, New York, USA.

[7] Claire H., Jarvis 1, Neil Stuart2 and William Cooper2. 2002. Infometric and statistical diagnostics to provide artificially-intelligent support for spatial analysis: The example of interpolation, Vol.17, 495-516, UK.

[8] Herborn, T., and Wimmer, M. Process ontologies facilitating interoperability in e-government: A methodological framework. Workshop on Semantics for Business Process Management, the 3rd Semantic Web Conference, Montenegro, June 2006.

[9] Lee, J. and Kim, S. 2009. An intelligent priority decision making algorithm for competitive operators in list-based scheduling. IJCSNS International Journal of Computer Science and Network Security, Vol.9 No. 1, January 2009.

[10] Kaaya, J. 2004. Implementing e-government services in East Africa: Assessing status through content analysis of government websites. Electronic Journal of E-Government, Vol.2, 39-54.

[11] Fernandez-Lopez, M. 2003 Overview of methodologies for building ontologies. Journal Data & Knowledge Engineering, Vol.46.

[12] Abulaish, M., and Dey, L. Interoperability among distributed overlapping ontologies: A fuzzy ontology framework. Proceedings of the 2006 IEEE/IWC/ACM International Conference on Web Intelligence.

[13] Noy, N., and McGuinness, D. Ontology development 101: A guide to creating your first ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05, Stanford Medical Informatics Technical Report SMI-2001-0880, March 2001.

[14] Islam, N., Abbasi, A.Z. and Zubair, A. 2010. Semantic Web: Choosing the right methodologies. Tools and Standards. IEEE.

[15] Ortiz-Rodriguez, F. Mexican e-government ontologies: An adaptation. The 4th International Latin American and Caribbean Conference for Engineering and Technology, Mayagez, Puerto Rico, June 2006.

[16] Ralf, K. Towards ontology for e-document management in public administration – the case of Schleswig-Holstein. In Proceedings of the 36th Hawaii International Conference on System Sciences 36, Hawaii, USA, IEEE Computer Society, January 2006.

[17] Salhofer, P., Stadhofer, B., and Tretter, G. 2009. Ontology driven e-government. Electronic Journal of E-government, Vol.7, 415-424.

[18] Salles, M. 2010. Supporting public decision making: A progressive approach aided by an ontology. International Journal of Decision Support System Technology, Vol.2, Issue 1, 21-35.

[19] Calegari, S., and Ciucci, D. 2010. Integer fuzzy logic in ontologies, Int. J. Approx. Reasoning 51, 391-409.

[20] Inyaem U., Meesad P., and Haruechaiyasak C. Dat Tran: Construction of fuzzy ontology-based terrorism event extraction. Third International Conference on knowledge Discovery and Data Mining, IEEE. DOI 10.1109/WKDD.113, 2010.

An Approach to build Ontology Library (OntoLib) For Academic e-Library

Hadeel Showket AL-Obaidy[1], Nahla El Zant El Kadhi[2], Manal El Zant[3]

[1] Engineering College: Computer Engineering Department, Ahlia University, Bahrain

[2] Business College: Management Information Systems, Ahlia University, Bahrain

[3] Laboratoire d'informatique Fondamentale de Marseille (LIF), UMR 6166, France

Abstract- *The aim of this paper is to suggest an approach to build ontology library for academic portal using Semantic Web technologies: RDF, XML, ontology, Simple Protocol and RDF Query Language (SPARQL) besides ASP.NET language. Ontologies capture the semantics of information from various sources. OntoLib enables the academic portal faculty to have their research papers published in one centralized place and make it easy for them to find the research papers that belongs to their colleges. This paper defines an OntoLib as Semantic Web components that lead to an intelligent Web. It discusses ontology and its uses in semantic web for digital libraries.*

Keywords: Ontology, Knowledge Base System, RDF, OntoLib

1. Introduction

Currently, the Web is essentially syntactic. The structure of documents or resources on the Web is well defined, however, only humans can read their contents which are inaccessible to machine processing. Machines can only browse for routine treatment or linking data between different Web pages. It means that almost all Web content is intended to be read by a human user. Therefore, it cannot be handled intelligently by computer programs. Searching for information on the Web is quite imprecise, and indeed, computer has no reliable methods to deal with semantic information. Today, the objective of the semantic web is to improve the search engine to be able to retrieve the correct information needed. It is desired to have more effective methods, best accuracy, promote sharing and reusing knowledge, and also an association of semantic metadata to documents and knowledge. To solve the problem of the actual web, in this paper we suggest an approach to build ontology library for academic portal that used Semantic Web technologies such as RDF, XML, ontology, Simple Protocol and RDF Query Language (SPARQL). This paper describes the Ontology which is a major component of Semantic Web.

2. Related Works

Yuangui Lei [12] define the semantic data quality and the semantic web portal infrastructure. They specified three main components for the infrastructure, an automated metadata extraction tool that supports the extraction of high quality metadata from heterogeneous sources, an ontology driven question answering tool which makes use of the domain specific ontology and the semantic metadata to answer questions in natural language format, and semantic search engine which enhances traditional text based searching. Their infrastructure contains a source data layer, extraction layer, a semantic data layer, a semantic service layer, and a presentation layer. Viljanen et al. [10] discussed the requirements of an ontology library system to support the different phases of an ontology life cycle and related user needs for creation, publishing, maintaining and using ontologies. Viljanen et al.[11] argue that the various ontology servers on the web should be made accessible using a common API that would provide a simple but universal methods for accessing the ontology content. As a solution, they propose the LOOS API and a metadata schema for describing the services. Siricharoen [9] introduces in their paper the realistic use of the available ontologies which are provided online. The purpose of the research is to transform ontologies to Unified Modeling Language (UML) object diagram using ontology editor “Protégé”. This research shows how it works efficiently with the real case study by using ontology classes in travel/tourism domain area. They specify the need to combine classes, properties, and relationships from more than two ontologies in order to generate the object model. The paper also presents a simple methodology framework which explains the process of discovering objects.

3. Semantic Web

Tim Berners-Lee, the creator of the Web, has declared that the Semantic Web is the next evolution of the Web [1, 2] which means an intelligent Web where information is stored understandable by computers in order to provide the user really seeking. Search engines help us by answering two questions: what are the pages containing a Word? And what are the most popular pages on a specific subject? The current Web is built primarily around the identifier URI, HTTP protocol and language HTML. Semantic Web is also based on URI, HTTP and the RDF language. The current Web automation capabilities are limited because the Internet has been designed to publish unstructured documents. It is difficult nowadays to access the right information needed online. For example: If you want to find a manufacturer of doors and Windows by typing the words "gates" and "windows" using the popular search engine Google, the results were not met your request and your result will include the Mr.Bill Gates pages and Microsoft Windows pages. One of the goals of the Semantic Web is to refine search on the Internet. To do this, we should add to the existing information a metadata layer that computers can develop it. The Semantic Web relies on three additional steps. First, it adds metadata to each Web resource. Then, it certifies their authenticity, and finally fixes the HTML errors. It is found that the use of XML and Resource Description Framework (RDF) should fix this problem. The XML can be used to solve this problem of research because this language has the ability to represent the semantics of data in a structured form. When searching information, it is possible to limit also the search to the documents matching particular elements. Also ambiguous words can be illustrious by the context they appear in. For example, a user searching the term “brown” in academicals documents, could be looking for papers written by Donald Brown, or papers published at Brown University, or also papers about the brown bear. So in this case, the person can specify even he wants to find <author>Brown</author>, <university>Brown</university>, or <subject>brown</subject>. W3C therefore propose to enrich existing information metadata RDF. The RDF format allows defining metadata to specify the characteristics of the information. RDF is triplets who will associate the defined metadata by group of three. One can describe a triplet as three URIs. The current Web uses links which are pairs, e.g. association "Mr. Tim BernersLee" and"W3C". Then, the Semantic Web seeks to present information by adding a third term. For this example, "founder" " <tim bernerslee=""> <fondateur> <w3c> can see the relationship between M.TBL and the W3C organization. In this case, the computer can perfectly determine what fact logically must be attached to another.

To resume, the goal of the semantic web is to express the meaning of web data in an appropriate way for automatic reasoning. This means that the descriptive data or metadata in machine readable form is to be stored on the web and used for reasoning. The challenge of the Semantic Web is to provide a language which expresses both data and rules. Then, any system of knowledge representation rules can be exported on the Semantic Web. Rules are added to the Web to give the possibility to make inferences, and answer user questions. There are two important developments of Semantic Web technologies: XML and RDF. XML allows adding an arbitrary structure to the documents without specifying the significance of structures. RDF is used by Semantic Web and it allows machines to understand documents and semantic data [8]. Figure 1 shows the Semantic Web stack diagram from the W3C.

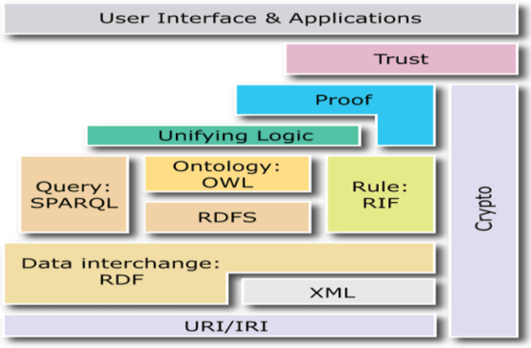


Figure 1: Semantic Web Diagram from the W3C [13].

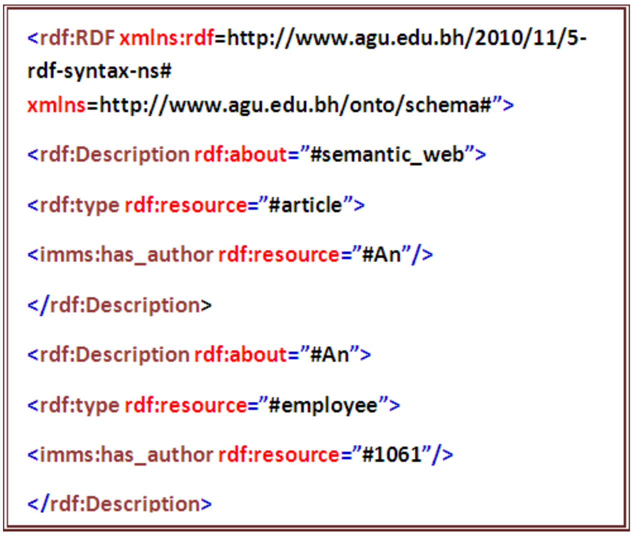
4. Definition of Ontology

In the literature, we can find different definitions for the word ontology. Sharing knowledge and reusing them are important features of any semantic web application. The ontology is responsible for sharing the common understanding of domain between people and machine and that's important for providing the interoperability feature to the semantic web application. The ontology is defined as a set of concepts (classes) and the relationship between these concepts. Ontology, is also defined as an explicit specification of a conceptualization [6] often considered as a reusable and shareable model. Geographical ontology can be used for exploration and extraction information and also for interworking of GIS [5]. In this paper, the authors specify how the ontology is considered also as a keystone of the SMA (System Multi Agents) using a high level communication. Ontology is defined as common vocabulary for persons who need to share information in a specific domain. Ontologies are used in different domains (Geography, Biology, etc.) to share common understanding of the structure of information among people or software agents, to analyze domain knowledge, and enable reuse of domain knowledge. In our paper, we define an ontology as a description of concepts in a domain (classed, concepts) when the properties of each concept describe various features and attributes of the concept (properties, roles), and slots that describe the properties of classes and instances. According to Gruber [6], ontology is an explicit specification of conceptualization that refers to an abstract model of a domain. It represents the way we choose to express our views through words, expression of concept and elements, and relations between entities.

The Ontology or the data schema is the tool that connects people with machines and makes them communicate in a smooth way. The idea that gives the ontology its ability to learn is to have the real world conceptualization in a knowledge base and digitalize it using a readable language such as Ontology Web Language (OWL). In the ontology, every element in the real world is expressed as <owl: Thing> composed of properties and instances. Every element is drill down to classes that holds the concepts which is drill down to object properties that describes the properties of the concepts. A concept can be defined as an entity composed of three distinct elements: the terms expressing the concept into language, the meaning of the concept also called the notion or concept intension, and the objects designated by the concept. The object is composed by instances that depict the properties. The combination of them is composing the knowledge base.

5. Building Ontology

Building ontology is an iterative process that consists in different steps. The first step consists in defining the classes of the ontology, and arranging them in a taxonomic hierarchy. During this step we should define the relation between the classes and specifying the super and subclasses. The second step defines slots, describe the allowed values for them and filling in the values for slots for instances. The third step consists in creating a knowledge-base by defining individual instances, filling the slots with specific values and adding restrictions to slots. The component of ontology are, the classes which are the concepts or things that describe any object in the world (e.g. Person, table, chair, etc), the relations used to provide relations between classes (e.g. hasName, hasArticle, etc), the functions that represent the relations with one result, and the instances of the classes (e.g. An, apple, etc). The following is an example of ontology in xml showing the components of this ontology described in the previous section. Figure 2 shows the diagram of the example described above:



```
<rdf:RDF xmlns:rdf=http://www.agu.edu.bh/2010/11/5-rdf-syntax-ns#
xmlns=http://www.agu.edu.bh/onto/schema#”>

<rdf:Description rdf:about=”#semantic_web”>

<rdf:type rdf:resource=”#article”>

<imms:has_author rdf:resource=”#An”/>

</rdf:Description>

<rdf:Description rdf:about=”#An”>

<rdf:type rdf:resource=”#employee”>

<imms:has_author rdf:resource=”#1061”/>

</rdf:Description>
```

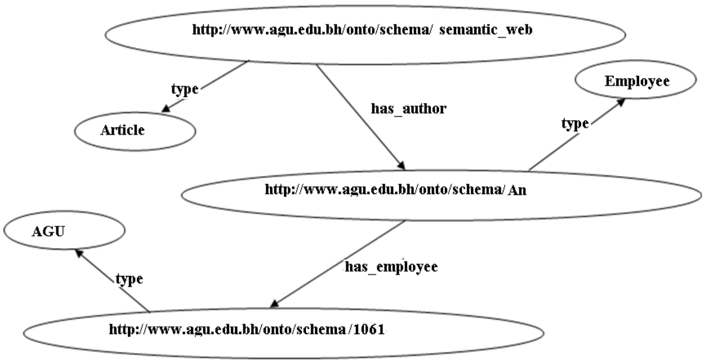
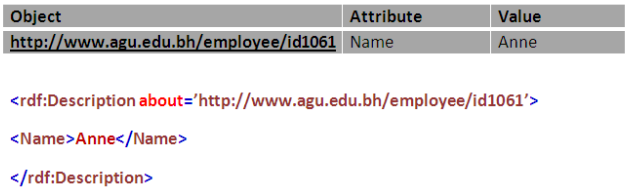

http://www.agu.edu.bh/onto/schema/ semantic_web
type
Employee
has_author
Article
type
http://www.agu.edu.bh/onto/schema/ An
AGU
has_employee
type
http://www.agu.edu.bh/onto/schema /1061

Figure (2): An ontology of Structure description

6. Resource Description framework

Resource Description Framework (RDF) is a knowledge representation language from W3C for semantic web. The RDF is an assertional logical language. It's providing the semantic web application with interoperability feature because RDF is readily for any program and facilitates data merging, no matter what schema used. Storing knowledge using this standard done by decomposing it into triples (assertions). One triple is composed of object, attribute and value. In another way it is composed of a resource (object), named property (attribute) and value for the property (value). So, any entity in the semantic web is related to another entity through a property. RDF allows structured and semi structured data to be exchanged between applications by using URI to identify each relationship between data in a triple. The triples can be expressed in three ways: tables, xml files and graphs. The easiest view is the graph view. For example: Name ('http://www.acadimic portal .edu.bh/employee/id1061", " Anne ")[7]. This example has three views table, xml and graph (see table 1).



Object	Attribute	Value
http://www.agu.edu.bh/employee/id1061	Name	Anne

```
<rdf:Description about='http://www.agu.edu.bh/employee/id1061'>
<Name>Anne</Name>
</rdf:Description>
```

Table 1: Table View of RDF example and the xml expression

7. Simple protocol and RFD Query Language

Simple Protocol and RDF Query Language (SPARQL) is a query language just like Standard Query Language (SQL) and used to perform manipulations add, update and delete the native graph stored in RDF stores. The results of the executed query using SPARQL are a set of RDF graphs, XML, JSON and HTML. The query of SPARQL should, declare Prefix using URIs, define RDF dataset and specifying the graph to be queried, identify which information should be returned as a result of the query, decide what the information to query for, and also contain the arranging query like ordering the resulted data. The following figure presents an example of SELECT query in SPARQL:

```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?student?university
WHERE {?name foaf:student ?student.
?name foaf:university?university.}
```

To execute SPARQL query via HTTP, the SPARQL end point must be used for querying from RDF stores that can be accessed through Web [4].

8. The proposed System Design

This paper proposed website for academic portal library that has a semantic web technology to search in the instructors' papers. Build an OntoLib model for academic portal's library is important for a university. OntoLib enables the Academic portal faculty to have their research papers published in one centralized place and make it easy for them to find the research papers that belongs to their colleges. This research papers add to the OntoLib using a Semantic Web model and stored in a knowledge base using ontology. As mentioned above, the Semantic Web is a component of Web 3.0 that its main purpose is the intelligent addition to Web. The proposed system increases the reputation of academic portal as well as it moves the ranking of its website to upper levels. As a result, visitors of the website are expected to increase by having more knowledge available for the academic portal faculty and the visitors.

The proposed system used an OntoStudio, dotNetRDFstore, OntoMat, Ontobroker, SQL server 2005 and Visual Studio 2008 programs will be used also in this paper to achieve the proposed OntoLib and create ontologies library component. It permits academics to search for information based on semantic not syntax elements. The following are the different components for our Ontolib. The first step in designing the semantic service is to design the ontology. The resulted ontology is stored as RDF documents. Figures 3 and 4 present the use case diagram and sequence diagram of the proposed system. In this paper we present the design of the Ontology for scientific articles as it is very important part of the academic portal. The most important elements that we can search about and we should specify for any article are the paper publication (paper, thesis, etc), paper title, paper library as a Knowledge base, any related thesis, etc. Figure 5 presents a view of the ontology publication concept.

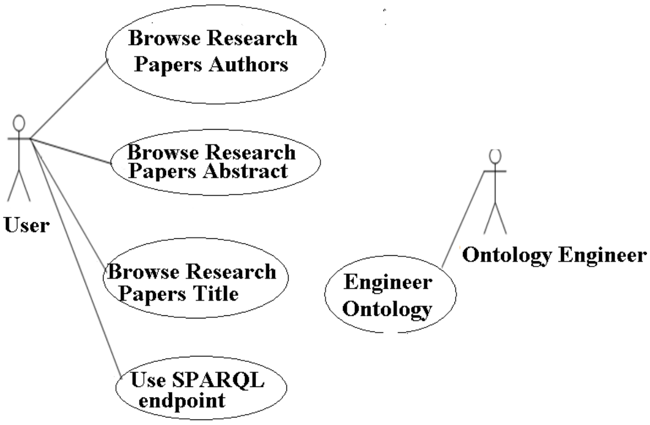

Browse Research Papers Authors
Browse Research Papers Abstract
User
Browse Research Papers Title
Use SPARQL endpoint
Engineer Ontology
Ontology Engineer

Figure (3): Use case diagram

Figure (6) visualizes the rules of the ontology. As we can see we have relations between the topic, the author and the publication. A paper is written by one or many authors. The topics and the type of the paper should be specified. Figures (7and 8) show the RDF diagram of the proposed ontology.

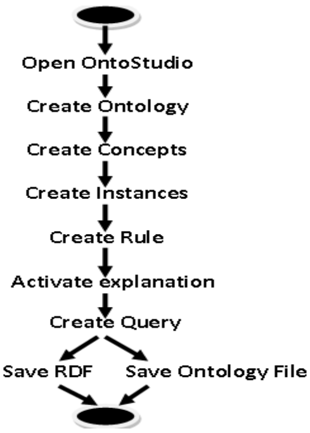


Figure (4): Activity diagram for engineering ontology.

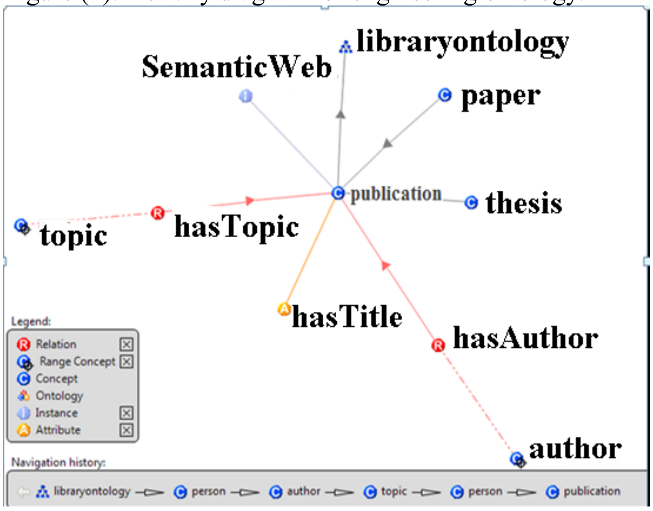


Figure (5): Diagram visualizing the ontology-publication concept.

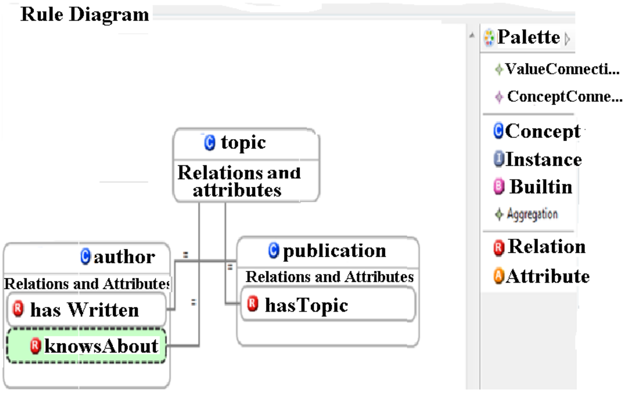


Figure (6): Rule diagram for the ontology

As mentioned before, the RDF graph is a set of triples (subject node, predicate arc, and object node). The RDF diagram of our example is shown in figure (7 and 8) and state that, for instance, the resource identified by <http:www.agu.edu.bh> is related to the resource denoted by <http:www.agu.edu.bh.LibOntology#hasTitle> via predicate <http:www.agu.edu.bh.LibOntology#instanse12475308772.

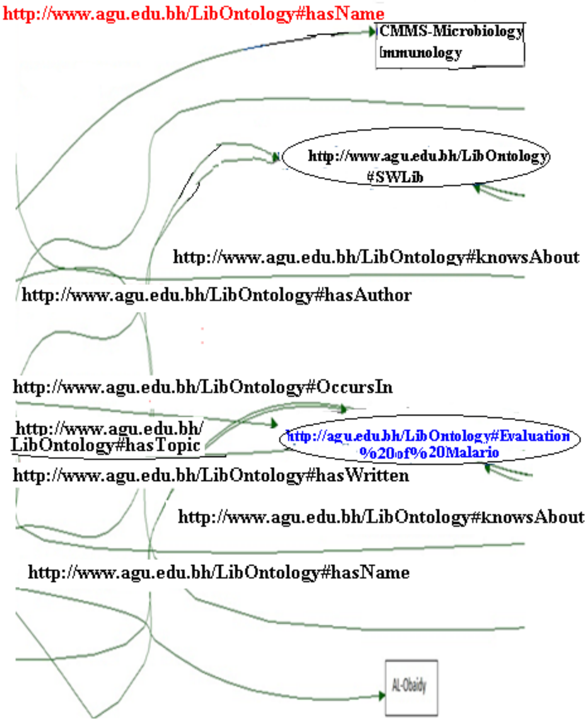

http://www.agu.edu.bh/LibOntology#hasName
CMMS-Microbiology Immunology
http://www.agu.edu.bh/LibOntology #SWLib
http://www.agu.edu.bh/LibOntology#knowsAbout
http://www.agu.edu.bh/LibOntology#hasAuthor
http://www.agu.edu.bh/LibOntology#OccursIn
http://www.agu.edu.bh/ LibOntology#hasTopic
http://agu.edu.bh/LibOntology#Evaluation %20of%20Malario
http://www.agu.edu.bh/LibOntology#hasWritten
http://www.agu.edu.bh/LibOntology#knowsAbout
http://www.agu.edu.bh/LibOntology#hasName
AL-Obaidy

Figure (7): The RDF graph.

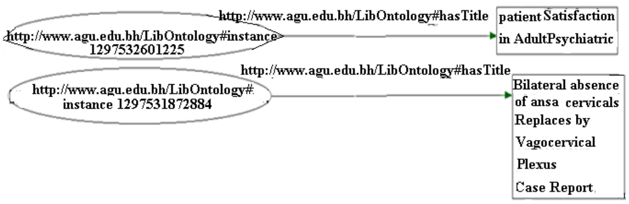

http://www.agu.edu.bh/LibOntology#hasTitle
http://www.agu.edu.bh/LibOntology#instance 1297532601225
patient Satisfaction in AdultPsychiatric
http://www.agu.edu.bh/LibOntology#hasTitle
http://www.agu.edu.bh/LibOntology# instance 1297531872884
Bilateral absence of ansa cervicals Replaces by Vagocervical Plexus Case Report

Figure (8): The RDF graph of Title concept.

9. Conclusion & Future directions

This paper introduces a proposal to develop ontology approach and show how to conduct this approach in the development of semantic academic portal. When building this ontology, it is very important to understand the knowledge base concept. This paper helps in identifying the proper ontology concepts, classes, subclasses, characterize the properties between them, shared all elements, describe the entities in those classes to describe the domain and the relationships between them and create a new knowledge from create related concepts.

This paper also presents the role of Ontologies in building Semantic Digital Libraries. Through the proposed OntoLib, the paper represents the Knowledge inside knowledge Base that can be used to build semantic infrastructure for Library 3.0. This library consists of a number of OntoLib for different domains; this can be lead to build integrated semantic libraries.

10. References

[1] Asuncion Gomez-Perez, Oscar Corcho, and Mariano Fernandez-Lopez, Ontological Engineering: With Examples from the Areas of Knowledge Management, E-commerce And the Semantic Web. First Edition (advanced Information and Knowledge Processing), 2004, ISBN: 1852335513, Springer-verlag.

[2] Berners-Lee, Tim; James Hendler and Ora Lassila, "The Semantic Web". Scientific American Magazine. http://www.sciam.com/article.cfm?id=the-semantic-web&print=true. Retrieved March 26, 2008.

[3] Berners-Lee, T., J. Hendler and O. Lassila : Semantic Web Scientific American, May 2000.

[4] Michael Science publications semantic C.Daconta, Leo J. Obrst, Kevin T.Smith,:Auide to Future of XML, Web Services and Knowledge Management, ISBN-13: 978-0471432579 2003.

[5] Nassim Dennouni, Ahmed Lehireche, Towards a spatiotemporel model for the interworking of the GIS, ICWIT 2010.

[6] T. R. Gruber. A translation approach to portable ontologies. Knowledge Acquisition, No 2, vol 5, 199-220, 1993.

[7] Nenad tojanovic, Alexander Maedche, Steffen Staab , Rudi Studer, York Sure. SEAL: a Framework for developing SEmantic PortALs, In K-CAP, pp. 155-162, 2001.What is Web 2.0: Design Patterns and Business Models for the next generation of software. O'Reilly website, 30th September 2005. O’Reilly Media Inc. Available online at: http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html [last accessed 17/01/07].

[8] Waralak Vongdoiwang Siricharoen Utilizing Ontologies Using Ontology Editor for Creating Initial Unified Modeling Language (UML) Object Model, , World Academy of Science, Engineering and Technology 64 2010.

[9] Viljanen, K., Tuominen, J., Hyv□onen, E.: Ontology libraries for production use: The Finnish ontology library service ONKI. In: Proceedings of the ESWC 2009, Heraklion, Greece, Springer.

[10] Kim Viljanen, Jouni Tuominen, Mikko Salonoja and Eero Hyvonen Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Gri_th, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., Musen, M.A.: Linked Open Ontology Services BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37(Web Server issue).

[11] Y Lei, An Instance Mapping Ontology for the Semantic Web. In Proceedings of the Third International Conference on Knowledge Capture, Banff, Canada, 2005.

[12] H. T. Al-feel, M. Koutb, and H. Suoror, web on scope: Anew architectural model for the semantic web 1, 2008.

A new approach for storing RDF triples based on ontology modularization

Meisam Booshehri [1], **Kamran Zamanifar** [2]**and Shahdad Shariatmadari** [3]
[1,2] Department of Computer Engineering, Najafabad Branch, Islamic Azad University, Najafabad, Iran.
[3] Department of Computer Engineering, Shiraz Branch, Islamic Azad University, Shiraz, Iran.

Abstract *- Managing RDF data in an effective manner is one of significant factors in realizing semantic web vision. Currently, there are two general approaches for storing RDF data which are column-oriented and row-oriented and each one has advantages and disadvantages. These two approaches are used in relational DBMSs. After reviewing different approaches and methods for storing RDF triples, we put forward a suitable bridging idea of ontology modularization that makes RDF data management systems outperform. The proposed approach will divide an ontology into several modules in order to reduce the size of information sets that we are working with in a specific moment at run time. We propose two new storing methods (which are table per ontology module method and vertically partitioned module method). We will analyze the effects of applying these methods on standard RDF repositories and figure out some directions that should be considered in designing the RDF repositories.*

Keywords- RDF repository, Ontology Modularization, RDF Triples, Column-oriented Approach, Row-oriented Approach.

1. Introduction

The content of annotation consists of some rich semantic information. These annotations are consumed both by human agents and software agents [1, 2, and 3]. Tim Burners Lee, the inventor of web, has presented a data model called “RDF Model” used for recourse annotation [1, 2, and 3]. This model covers three concepts including: resources, properties and statements. A resource is a thing you talk about it (can reference it). Every resource has a URI. RDF definitions are itself Resources [1, 2, and 3]. Properties, defines relationship to other resources or atomic values. Statements consist of resources, properties and values [3, 4]. Values can be resources or atomic XML data [3, 4].Using RDF Model is a common method for resource annotation. Using relational databases is one of common techniques for storing annotations. To date two types of DBMSs (row- oriented and column-oriented) have been constructed [10, 11, 12] for this purpose. And each type uses various approaches to form data tables.

In this paper, we propose two new storing methods based on the idea of ontology modularization. Ontology modularization [18] is used in ontology engineering as a way to structure ontologies, meaning that the construction of a large ontology should be based on the combination of self-contained, independent and reusable knowledge components.
In this paper we propose a new approach for storing RDF triples in relational databases based on ontology modularization.

The remainder of paper is organized as follows. In section 2 we talk about related work in which we review the most important existing storage methods. In section 3 we propose our new storage methods. In section 4 we discuss the new methods and compare them with corresponding methods. And in section 5 we conclude and talk about future work.

2. Related Work

It is important to select the best solution for storing and retrieving RDF information. [13] classifies state-of-the-art RDF storage and indexing schemes in two subcategories. First subcategory includes relational schemes that use RDBMSs for storing RDF data and the second subcategory includes native schemes that build RDF-specific stores and indexes from scratch. Although native schemes perform well because of their suitable design, relational schemes are preferred to use as a solution for RDF data management and this is because of the maturity, generality and scalability of relational databases [13]. In this context we must put forward a question. “How the table design should be for storing RDF triples?” By now some

[1] He is a Master Student in department of Computer Engineering, Najaf Abad Branch, Islamic Azad University, Najaf Abad, Iran. (email: m_booshehri@sco.iaun.ac.ir)

[2] He is an assistant professor in department of computer engineering, Najafabaad branch, Islamic Azad University, Najafabad, Iran. (email: Zamanifar@eng.ui.ac.ir)

[3] He is faculty member in department of computer engineering, shiraz branch, Islamic Azad University, Shiraz Iran. Also, he is PhD candidate in department of computer science and information technology, University Putra Malaysia. (email: shariatmadari@iaushiraz.ac.ir)

standard storage methods for row-oriented database systems has been proposed including Horizontal Table [5], Vertical Table [7, 10], Horizontal Class [8], Table per Property [8] and Hybrid approach [8]. Also with the birth of column-oriented database systems some other techniques have been proposed including vertically partitioned method [14] and sextuple indexing technique [13 , 15] etc. In the following sections we survey these methods.

2.1. Row-oriented database systems storage methods

Here we introduce storage methods which have been used in row-oriented RDBMSs and discuss the benefits and drawbacks of these methods.

2.1.1. Horizontal Table method

This storing method uses a universal table in the database for all existing ontologies [5]. Fields of this table includes instances' IDs, instance types and values of instance properties. You can see a representation [8] for horizontal table in Fig. 1. The domain of all columns in the table except the column "type" could be both URIs and literals.

Instance's ID	Type	Property_1	Property_2	…	Property_n
…#1	Class_A	Value_a	Value_b	…	Value_c
..#2	Class_B	Value_d	Value_e	…	Value_f

Figure 1. A representation for horizontal table

The advantages of this method come below:

1- Subject and predicate are stored once [5, 8].

2- Short query run time [5, 8].

3- Its structure is simple because every instance has one entry in the table [5, 8].

On the other hand we can mention five cases as disadvantages of this method which come below:

1- Abundance of columns: considering the size of knowledge-base and number of properties, we can say that the relational database system would be limited with number of columns [5, 8].

2- Limitation of property values (lack of support in the case of multi-valued properties): the said structure forces every property to have only one value; however, a large number of properties are naturally multi-valued [8].

3- Sparseness: it is obvious that every property has a corresponding field in the table despite the fact that many records might have no values for filling fields. Therefore many fields in the database may have null values [5, 8].

4- Difficulty of maintenance: whenever a new ontology is being incorporated into a system or a system changes an existing ontology, the said table needs to be reorganized. If the table is huge, such changes could be so expensive [5, 8].

5- Decrease in performance: this approach results in a large-size database. Also load time in this approach is longer in comparison to other ones [6, 8].

Another type of this approach is using a table for each ontology. It is obvious that this solution decreases the number of columns and cost of changes in every table, however, in the case of this type of horizontal table still limitations of properties exists and tables in this method have larger number of columns in comparison to other methods and there exists sparseness problem yet.

2.1.2. Vertical Table Method

In this method, we have a universal table [7]. The table has only three fields including subject, predicate and object. In predicate field we can store both property names and the string "type" which indicates that the current record defines the type of a resource. In subject field we can store the URI of resources and in object field we can store class names and values of properties. You can see a representation [8] for vertical table in Figure 2. Some benefits of this method are its simple structure, ease of maintenance and fixed number of columns. This method is the most straightforward relational method [13].

Predicate	Subject	Object
Type	…#1	Class_A
Property_1	…#1	Value_a
Property_2	…#1	Value_b
…	…	…

Figure 2. A representation for vertical table

It is very general that is every type of data could be stored in this format [13]. Also some disadvantages of this method are as follows:

1- High query run time [7, 8].

2- As mentioned in [8] "This design means that any query has to search the whole database and queries that involve joins will be especially expensive. In particular, queries about the members of a class will be particularly difficult, because there is no explicit treatment of the class hierarchy".

3-we have to save values in string format.

2.1.3. Horizontal class method

This method is similar to horizontal table method but with smaller tables. In this method we have a separable table for every class in ontology. A representation [8] for horizontal class table has been shown in Figure 3.

ID	Property_1	Property_2	…	Property_n
…#1	Value_a	Value_b	…	Value_c

Figure 3. A representation for horizontal class

The advantages of this method are as follows:

1- Less sparseness in comparison to horizontal table method [8].

2- the most important benefit of this method is that the query on the property of an instance or a set of instances is done effectively [8].

3- Lower query run time in comparison to horizontal table method [8].

One of the disadvantages of this method is that the properties which have no explicit domain should be repeated in all the tables [8].

2.1.4. Table per property method

This method was first used in row-oriented RDBMSs but it is more suitable for column-oriented RDBMSs**.** In this method every property has a corresponding table [5]. A representation of such tables has been shown in Figure 5. Instances of all classes are saved in a table named "Type" in which every record relates an instance to a class.

Subject	Object
…#1	Class_A
…#1	Class_B

Figure 4. A representation for the table "Type"

Subject	Object
…#1	Value_a
…#2	Value_d

Figure 5. A representation for the table of a property

A representation[8] for the table "Type" has been shown in Figure 4. Some advantages of this method are short time for simple queries and decrease in size of tables [8] .Some disadvantages of this method includes increase in number of tables and high run time for complicated queries due to lots of join operations that must be done [8].

2.1.5 Hybrid method

This method combines different storing methods therefore in comparison to other methods performance increases. Also there is less limitations and totally less drawbacks in comparison to the state in which we use every storing method independently [9]. This method has been used in many data repositories [8].

2.2. Column-oriented database systems storage methods

Here we talk about methods which are naturally suitable for and have been used in column-oriented RDBMSs. We discuss the advantages and disadvantages of these methods as well.

2.2.1. Vertically partitioned method

Property tables method did not fit well with the semi-structured nature of RDF data and SPARQL queries with unbound variables in the property positions. Thus to solve such limitations Abadi et. al. [22] proposed the vertically partitioned method. This method is the same table per property method with the difference that Abadi et. al. have integrated this method with a column-oriented RDBMS and some side techniques such as data compression have been used to improve the performance of this technique. Some advantages of this method are:

1- Support for multi-valued attributes [22].

2- Support for heterogeneous records [22].

3- Only those properties accessed by a query need to be read [22].

4- No clustering algorithms are needed [22].

Also one of the disadvantages of this method is that "if the property in a query is bound to a variable, then the rows returned from each property table must be union-ed. In the case where the property is not part of the result, then the union operator must also perform a duplicate elimination. Finally, since the data is not clustered on objects, a query which joins on objects, will not allow the use of a fast (linear) merge join" [15, 21]. However Abadi et. al. in [22] have mentioned that "although the vertically partitioned approach require more joins relative to the property table approach, properties are joined using simple, fast (linear) merge joins".

2.2.2. Sextuple indexing technique

The idea of sextuple indexing has been proposed in [15] for the first time. This technique points to the fact that a RDF triple is a three-dimensional entity and could be indexed in six ways. Thus using this approach allows fast merge-joins for any pair of two triple patterns [13, 15]. In spite of this fact neither Hexastore nor RDF-3X (as native schemes for storing RDF data) doesn't point to the ways that realize sextuple indexing technique in relational database systems[15, 23]. Therefore Xin Wang et al. in [13] propose a novel storage technique for RDF data in which sextuple indexing is applied to column-oriented RDBMSs. As mentioned in [13] "The experimental results shows that this method outperforms the row-oriented RDBMS approach by upto an order of magnitude, and is competitive to the best native RDF store RDF-3X". Also in our point of view applying sextuple indexing technique could be effective as we mention in the next sections.

3. Proposed approach

In our proposed method we try to reduce the size of information domain which we are working with in a specific moment at run time.

3.1. Ontology Modularization

Nowadays big size of existing ontologies is a crucial problem. Maintaining these ontologies which might have more than one hundred thousand concepts is difficult [16]. Also reusing the whole ontology is time-consuming and costly [17]. A solution to solve this problem is to divide ontology into some parts with a special subject. These parts are ontology modules and this process is called ontology modularization. There are various definitions for an ontology module. [18] defines ontology module as follows: "An ontology module is a reusable component of a larger or more complex ontology, which is self-contained but bears a definite association to other ontology modules, including the original ontology". Also as mentioned in [17] we say a module is self-contained while special reasoning tasks such as inclusion relation or query answering within a module are possible with no need to access other modules.

By now different approaches for ontology modularization have been proposed[24,25,26,27,28,29,30,31], including logic-based approaches[28,31] and Graph theory-based approaches[28] etc. We think that all these approaches could be useful in different places. But in this paper we don't use a specific approach or a special partitioning algorithm for the ontology modularization step in our proposed RDF storage approach.

The proposed approach in this paper will divide an ontology into several modules in order to reduce the size of information sets that we are working with in a specific moment at run time. So here we consider a two-column table named "Module_Class_Table" in which every record relates a class to the ontology module that the class belongs to. A representation of such table has been showed in the figure 6.

Class_Name	Module_ID
Class_A	…#10
Class_B	…#2

Figure 6. A representation for Module_Class_Table

In spite of previous storing methods which work on the whole ontology, our approach emphasizes on ontology modules as the base of designing database tables. Based on this new approach, we suggest two new methods w.r.t. the viewpoint of both row-oriented RDBMSs and column-oriented RDBMSs.

Before explaining these two methods we define two types of properties: intra-module properties and inter-module properties. Intra-module properties are those which are only related to concepts among an ontology module. And inter-module properties are those which connect couples of concepts from different modules. It is obvious that we may have both of these two types of properties within an ontology. Also with respect to this classification we can categorize queries into two subcategories which are *intra-module queries* and *inter-module queries*. An intra-module query work only on the information and concepts within a specific module (tables extracted from a specific ontology module), however, an inter-module query may work on the information and concepts within a specific ontology (especially among modules). Of course an inter-module query could be a combination of some intra-module queries and some inter-module queries.

Considering the classifications mentioned above, when a query is applied to a database system based on our approach, at first it must be recognized if the query is inter-module or intra-module. Such decision can be made by a simple query on the Module_Class_Table. Therefore for supporting our new approach, we think that a preprocessing unit should be embedded in RDF data repositories such as Jena and SW-store in which queries are analyzed to determine if a query is inter-module or intra-module.

3.2. Proposed method for row-oriented RDBMSs (Table per ontology method)

In this method every ontology module is stored into a separable table. Also in this method we use a universal vertical table which contains RDF triples that describe the inter-relations between concepts of ontology modules.

3.3. Proposed method for column-oriented RDBMSs (vertically partitioned module method)

Here we propose a new method named "vertically partitioned module method". In this method each intra-module property has a corresponding two-column property table. However, inter-module properties are mentioned in a universal vertical table which contains RDF triples that describe the inter-relations between concepts of ontology modules. Also intra-module property tables could be indexed by sextuple indexing technique and combined with other side techniques such as data compression techniques.

4. Discussion

As mentioned in [13] the most straightforward relational method for storing RDF triples is vertical table method. This method is very general that is every existing data type could be stored in this format [13]. The basic problem of this method is the problem of expensive self-joins over this vertical triples table which is possibly large [13]. These self-joins is because of SPARQL queries with multiple triple patterns [13]. Here we quit this method because of its basic problem and won't compare it with other methods.

Here we introduce a useful factor named "Working Information Set" or WIS. WIS is the smallest subset of information in a domain to which access probability is more than other subsets of information in the same domain in different time intervals. We use this relative concept to compare the performance of our methods with the others.

Another concept that we need it in this discussion is data fragmentation which is a topic discussed in distributed databases. It is classified into three subcategories: horizontal fragmentation, vertical fragmentation and hybrid fragmentation [19, 20].

Our approach emphasizes on ontology modules as the database design basis. It is obvious that the number of extracted tables from an ontology module is less than the number of extracted tables from the whole ontology. Moreover, existing data in the tables of a module is less than existing data in the corresponding tables of the whole ontology. It means that focusing on modules instead of the whole ontology, results in decrease in the size of WIS. And this subject causes lower load time and more performance. In our point of view this is one of the benefits that ontology modularization brings to column-oriented databases as well as row-oriented databases.

If we consider horizontal table as a reference table, horizontal class tables and property tables are produced by hybrid fragmentation of horizontal table. Also using ontology modules (as database design basis) instead of considering whole ontology is similar to data fragmentation techniques which we call it logical data fragmentation.

In this paper, we consider module extraction as a new type of data fragmentation in the context of RDF database systems. The more effective algorithms for module extraction we use the more suitable logical data fragmentation we have. As a result we can say that ontology module tables are also produced by hybrid fragmentation of horizontal table. Also vertically partitioned module tables are produced by hybrid fragmentation of each ontology module table.

It seems that separating ontologies into modules is a justifiable data fragmentation. Increasing the degree of concurrency and system throughput are two important benefits of data fragmentation in distributed databases [19, 20]. Therefore module extraction and use of ontology module (as database design basis) would make us closer to these two benefits.

On the other hand there are two important disadvantages for data fragmentation as follows:

1-If we have some requirements which are in conflict with data fragmentation, the performance would decrease. For instance it is costly to retrieve several different parts that must be joined or unioned [20].

2-.During data fragmentation some attributes that is related to an association relationship may be separated into several parts and located in distinct sites. This will cause the problem of difficulty in semantic control of data and difficulty in integrity control as well [20].

According to the self-contained feature of an ontology module, we can say that the problems mentioned above are not serious about ontology modules.

Based on the above discussions, it is better to compare *horizontal table* method, *horizontal class* method and *table per ontology module* method with each other because these three methods are naturally suitable for row-oriented RDBMSs. On the other hand it is better to compare *table per property* method, *vertically partitioned* method and *vertically partitioned module* method with each other since these methods are naturally suitable for column-oriented RDBMSs.

4.1. Discussion about Table per ontology module method

Generally this method is a middle method between horizontal table method and horizontal class method. If the ontology has only one module, this method is equivalent to horizontal table method. Also if we consider every ontology class as a module, this storing method is equivalent to horizontal class method. But in our point of view, because of strong associative relationships that some classes might have with each other, it seems that mapping a class to a module is not suitable.

We consider three general states for ontologies existing in a domain and then we compare it with previous methods. Suppose we have extracted 'n' ontologies.

State 1- None of 'n' ontologies has the capability of modularization. In this state there are n tables for storing RDF triples and the new method is exactly similar to the type of horizontal approach in which every ontology is stored in a separate table. So the advantages and disadvantages of this method would be the same as horizontal table method.

State 2- the number of extracted modules from existing ontologies is abundant and the number of classes of the modules is few. In this state the advantages and disadvantages of the new method are very similar to the advantages and disadvantages of horizontal class method.

State 3 – The existing ontologies have the capability of being modularized and also the number of extracted modules is average. It seems that this state is the best state for this method. In our point of view such state could show itself in multifaceted ontologies (ontologies that cover different domains of information). Some features of this method are as follows:

- This method has less sparseness in comparison to horizontal table method; however, its sparseness is more than horizontal class method.
- The number of table columns in this method is fewer than the number of table columns in horizontal table method and more than the number of table columns in horizontal class method.
- Comparing to horizontal table method, this method has fewer number of tables but in comparison to horizontal class method it has larger number of tables.
- Maintenance process in this method is easier than horizontal table method. Surely it comes from benefits of modules in software engineering.
- Less load time in comparison to horizontal table method and longer load time comparing with horizontal class method.

Performance of this method is related to the ontologies of the application which we want to design a database for it. If we have large ontologies that have mostly dependant components, we can say that this storing method has a better performance in comparison to other previous storing methods. But if we have ontologies with coherent components and there is no capability for transforming them into many modules, we can say that this storing method won't have a good performance.

Generally this new method could be useful with respect to features mentioned above. Composing this new method with other storing methods seems to be useful too.

4.2. Discussion about Vertically partitioned module method

Some row-oriented RDBMSs use table per property method combining with other methods. Actually vertically partitioned method is the improved version of table per property method. Also if we compare vertically partitioned method with vertically partitioned module method, we can see that superiority of vertically partitioned module method comes from its smaller WIS. This fact leads to the lower size of tables, lower load time and possibly higher performance. Generally we think that vertically partitioned module method improves the vertically partitioned method from this aspect .Of course it is obvious that separating ontologies into modules in a defective manner decreases performance.

5. Conclusion and future work

In this paper we proposed a new approach for storing RDF triples in relational databases based on ontology modularization and we presented two new storing methods which are *table per ontology module method* and *vertically partitioned module method*. The *table per ontology module* method seems to be naturally suitable for row-oriented RDBMSs. In this study, we show that in some cases this method is an appropriate alternative to horizontal table method and horizontal class method.

The second method is *vertically partitioned module* method. This method can be integrated with some techniques such as sextuple indexing method [15] to get a better performance. This method is naturally suitable for column-oriented RDBMSs.

Our approach is based on ontology modularization in which we divide an ontology into several modules in order to reduce the size of WIS.WIS is the smallest subset of information in a domain to which access probability is more than other subsets of information in the same domain in different time intervals. In this paper we have compared the *table per ontology module method* with the existing methods which are naturally suitable for row-oriented RDBMSs. Also we have compared *vertically partitioned module method* with the existing methods which are naturally suitable for column-oriented RDBMSs. Totally our preliminary study shows that our new methods results in smaller WIS in comparison to other previous corresponding methods. It is obvious that Smaller WIS leads to the lower load time and totally higher performance.

In our point of view ontology modularization is a type of data fragmentation in the context of RDF database systems. Therefore our approach results in increasing degree of concurrency and system throughput in distributed environments. Also according to the self-contained feature of an ontology module, we can say that the problems [20] of data fragmentation are not serious about ontology modules and totally our storing approach.

As future work we are going to implement our methods which have been explained in this paper. This is related to designing a new model for RDF data repositories which enables them to recognize modules in a specified ontology. It seems that it is better to consider a new unit in a data repository which pursues module extraction smartly and automatically. For designing this model the structure of data repositories must be examined precisely.

Also a preprocessing unit for analyzing queries should be embedded into data repositories to determine if a query is inter-module or intra-module. Therefore we must design this unit.

Moreover, we are planning to select and extend a suitable partitioning algorithm among existing ones [28] that modularize ontologies in an efficient way.

6. References

[1] The Semantic Web, Scientific American, May 2001, Tim Berners-Lee, James Hendler and Ora Lassila

[2] Tim Berners-Lee, Eric Miller, Jim Hendler, "Integrating applications on the Semantic Web", Journal IEEE Japan, 122(10):676-680, 2002.

[3] "Practical RDF". (2003) RDF/XML focused but well written and highly recommended http://www.oreilly.com/catalog/pracrdf/

[4] Michael C. Daconta et al., "The Semantic Web: A Guide to the Future of XML, Web Services and Knowledge Management" Complementary alternative http://www.amazon.co.uk/exec/obidos/ASIN/0471432571/

[5] R. Agrawal, A. Somani, and Y. Xu. Storage and Querying of E-Commerce Data. In Proc. of VLDB, 2001.

[6] D. Florescu and D. Kossman. A performance evaluation of alternative mapping schemes for storing XML data in a relational database. Technical report, INRIA, France, May 1999.

[7] D. Beckett and J. Grant, Mapping Semantic Web Data with RDBMSes. 2001, http://www.w3.org/2001/sw/Europe/reports/scalable_rdbms_mapping_report/

[8] Z. Pan and J. Heflin, "DLDB: Extending Relational Databases to Support Semantic Web Queries". Technical Report: LU-CSE-04-2006

[9] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis & K.Tolle, "On Storing Voluminous RDF Description: The case of Web Portal Catalogs", In Proc. of the 4th International Workshop on the Web and Databases (WebDB2001) in conjunction with ACM SIGMOD'01 Conference, 2001.

[10] Stonebraker et al. . "C-Store: A Column-Oriented DBMS." ,VLDB, 2005.

[11] Daniel J. Abadi, Samuel Madden, Nabil Hachem, "Column-stores vs. row-stores: how different are they really?", In Proc. SIGMOD Conference, 2008: 967-980

[12] Harizopoulos, Liang, Abadi, Madden . "Performance Tradeoffs in Read-Optimized Databases", In proc. VLDB, 2006.

[13] Xin Wang, Shuyi Wang, Pufeng Du and Zhiyong Feng. "Storing and Indexing RDF Data in a Column-Oriented DBMS". In proc. IEEE Conference, 2010.

[14] D. J. Abadi, A. Marcus, S. R. Madden, and K. Hollenbach, "Scalable semantic web data management using vertical partitioning," In *Proc. VLDB*, pp. 411–422, 2007.

[15] C. Weiss, P. Karras, and A. Bernstein, "Hexastore: sextuple indexing for semantic web data management", In Proc. VLDB, pp. 1008–1019, 2008.

[16] Schlicht, A, Stuckenschmidt H. . " A Flexible Partitioning Tool for Large Ontologies". In proc. International Conference on Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM , 2008.

[17]B. Konev, C. Lutz, D. Walther, and F. Wolter, "Logical Difference and Module Extraction with CEX and MEX", In Proc. Description Logics, 2008.

[18] Paul Doran, "Ontology reuse via ontology modularization", In proc. KnowledgeWeb PhD Symposium, 2006.

[19] A. Silberschatz, et al., "Database System Concepts", 5th edition, Mc Graw Hill, 2006.

[20] M. T. Ozsu, et al., "Principles of Distributed Database Systems", Prentice Hall, USA, 1999.

[21]Lefteris Sidirourgos, Romulo Goncalves, Martin Kersten, Niels Nes, Stefan Manegold. "Column Store Support for RDF Data Management: not all swans are white". In Proc. VLDB ,2008.

[22] Daniel J. Abadi, Adam Marcus, Samuel Madden, Kate Hollenbach: "SW-Store: a vertically partitioned DBMS for Semantic Web data management". VLDB J. 18(2): 385-406 , 2009.

[23] T. Neumann and G. Weikum, "RDF-3X: a RISC-style engine for RDF", In *Proc. VLDB*, pp. 647–659, 2008.

[24] H. Stuckenschmidt and M.C.A. Klein, "Structure-Based Partitioning of Large Concept Hierarchies", In Proc. International Semantic Web Conference, 2004, pp.289-303.

[25] Mathieu d'Aquin, Anne Schlicht, Heiner Stuckenschmidt, Marta Sabou, "Criteria and Evaluation for Ontology Modularization Techniques. Modular Ontologies" 2009: 67-89

[26] Jie Bao, Doina Caragea, Vasant Honavar, "Modular Ontologies - A Formal Investigation of Semantics and Expressivity", In proc. Asian Semantic Web Conference, 2006, pp.616-631

[27] Bernardo Cuenca Grau, Bijan Parsia, Evren Sirin, Aditya Kalyanpur, "Automatic Partitioning of OWL Ontologies Using E-Connections", in proc. Description Logics, 2005.

[28] Pathak J, Johnson TM, Chute CG. "A Survey of Modular Ontology Techniques and their Applications in the Biomedical Domain. International Journal of Integrated Computer-Aided Engineering", Vol. 16(3), 2009; pp. 225-242.

[29] Palmisano, I., Tamma, V., Payne, T. R., Doran, P.. "Task Oriented Evaluation of Module Extraction Techniques", In Proc. The Eighth International Semantic Web Conference (ISWC'09), October 25th-29th 2009.

[30] Julian Seidenberg, Alan Rector. "Web Ontology Segmentation: Analysis, Classification and Use" In Proc. the 15th international conference on World Wide Web, 2006, pp. 13-22.

[31] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler, "Extracting Modules from Ontologies: A Logic-based Approach", In Proceedings of OWLED, 2007.

SESSION

NOVEL APPLICATIONS + TOOLS + ONTOLOGY MAPPING

Chair(s)

Prof. Hamid R. Arabnia

Semantic Search Tool for Adverse Event Reports of Medical Devices

Lisham L Singh[1,2], Sithu D Sudarsan[1], Raoul P Jetley[1], Brian Fitzgerald[1], and Mariofanna Milanova[2]

[1]US Food and Drug Administration
[2]University of Arkansas at Little Rock
llsingh@ualr.edu, [{sithu.sudarsan, raoul.jetley, brian.fitzgerald}@fda.hhs.gov], mgmilanova@ualr.edu

Abstract

Signal detection is a critical activity carried out by US Food and Drug Administration (FDA) analysts as part of the agency's mission of public health protection, using large amount of data gathered in disparate formats. Much of this data is unstructured narrative text, limiting use of traditional data mining. Therefore, FDA analysts spend a significant time to locate appropriate documents before relevant information in them can be used. To address this, researchers at the Center for Devices and Radiological Health (CDRH) are developing a semantic search and retrieval framework (SARF) with a Semantic Search Tool (SST) to locate documents containing relevant information quickly.

SARF is capable of analyzing millions of regulatory documents to provide FDA analysts with an intuitive SST for signal detection and evaluation. SARF utilizes a range of techniques innovatively, including use of structured information available in the document for sorting and filtering, while utilizing the narrative unstructured information for context and semantics. Semantic analysis is achieved using, but not limited to, dictionaries, ontologies, and standards.

At FDA, while the value of enabling machine aided narrative text analysis is immense, the benefits of using structured data cannot be over looked. Therefore, SARF is innovatively architected and engineered to take advantage of both structured and unstructured information available with regulatory submissions.

SARF and SST, developed using open-source tools, have been tested with several millions of documents running into multiple terabytes. Yet, the time to query for specific narrative is in terms of milliseconds. A case study is presented to highlight the use of our tool.

1 INTRODUCTION

Decision making is one of the routine but critical activities performed in organizations. Such decisions include technical, scientific, economic and managerial ones. Potential public impact makes decision making more critical in regulatory organizations; hence, scientific and informed decision making is extremely important. For instance, if adverse events reported in respect of a medical device has safety implications then it may have to be recalled based on review by US Food and Drug Administration (FDA). Identifying such specific issue based on available information and evaluation is known as signal detection. Identification, evaluation and confirmation of a signal [1] is an essential ingredient of decision making. This paper outlines a *Semantic Search and Retrieval Framework* (SARF) [2] with a *Semantic Search Tool* (SST) that helps in signal detection.

Incorrect signals result in either *Type 1* or *Type 2* error. When a recall is not made where it should have been made then such an error is known as *Type 1* error; this results in allowing an unsafe device to continue in the market. When a recall is made where it should not have been made then such an error is known as *Type 2* error; this results in denying an acceptable medical device to the needy. Even though specific procedures are followed, sometimes errors do occur. One of the reasons for such errors is the inability to identify and analyze relevant documents and associated meta-data among millions of documents within the available time frame. These documents have accrued over time. SST helps the reviewers performing signal detection in reducing these errors.

Locating relevant information among millions of documents based on queries by the reviewer is a challenging task, due to several reasons. Few of them are: (i) as the size of the historical data grows, it becomes impractical to manually search for similar instances of the problem to make informed decisions within time constraints; (ii) the heterogeneity in the structure and format of data (emails, pdf's, xml, doc, txt so forth as so on) adds to the complexity of searching such data; (iii) many documents contain domain specific descriptions with specific abbreviations, acronyms or terminologies; (iv) diversity of reporting sources, including public, manufacturers, hospitals, laboratories, etc. results in the same or similar events being described in different ways; and (v) useful signals are contained in the narrative description of the event as it provides the context and requires semantic rather than just syntactic search making simple text search across these documents less effective.

At present, reviewers identify relevant document by filtering based on select structured data fields, e.g. date of report, and product code followed by manual analysis of narrative text. Given that over 200,000 devices related adverse events are received by FDA each year, and the number of relevant documents among them are few, the reviewer could be overwhelmed and potentially fatigued. Therefore, any tool that will speed up locating narrative text and reduce the fatigue by eliminating non-relevant documents for analysis enhances productivity. This makes SST a very useful tool for signal detection and evaluation.

In this paper we present SST that would aid reviewers in finding relevant documents by supporting various types of queries ranging from syntactic to semantic search with meta-data based filtering/sorting. Users have option to automate and customize the use of different dictionaries and reference tables while constructing the query to accommodate specific semantic requirements. Our approach enables text-mining over a huge document collection with document size ranging from few kilobytes to several gigabytes; this improves upon current large scale text mining solutions which expect specific document size, e.g., typical document size of few hundred kilobytes or about 5000 to 10000 words per document and so on. SARF supports multiple document corpora.

SST is a practical/scalable tool that facilitates efficient searching of relevant documents, using text mining techniques based on SARF. Our approach is generic and independent of the structure of data being searched. SARF allows automation as well as customization by query reformulation and expansion. It also provides resources to the users so that they can refer or look up as reference documents. The efficiency of the tool remains unaffected with increasing size of document collection, making it scalable.

The tool is designed to search across loosely-coupled corpora i.e. independent corpora, as SARF enables this feature without any significant compromise in response time. Even though we focus on medical device adverse event corpus, there are multiple corpora. This feature is very useful, for example, when an adverse event involves a device as well as, says a drug and reviewers need to locate documents from either of the corpus. Drug related adverse events form its own corpus.

Our implementation uses open source tools including Apache Lucene [3], which is a high-performance, open-source, information retrieval Java API library. This approach has made the solution to work across operating systems, while avoiding the reinventing of the wheel.

SST is web enabled and has a user friendly and interactive graphical presentation of the search result.

The rest of the paper is organized as follows, Section 2 discusses background and some related work. We outline the problem in Section 3. In Section 4, our approach is discussed. Section 5 presents a case study of the tool. Section 6 concludes the paper by summarizing the contributions proposed in this work.

2 BACKGROUND

A traditional search engine based approach could be useful in our case, but for its limitation on being just syntax based. Semantic search engine fits the requirements better as the descriptions are semantically similar. Syntax based searching is straightforward and it works on looking for documents that contain the terms or patterns specified in the query. Semantic search engines, on the other hand, aim to improve search accuracy by taking the contextual meaning of terms as they appear within the search space, to generate relevant results. However, developing such a tool is more challenging.

Though there are difficulties in developing accurate and powerful semantic search engines, the popularity of such applications are increasing and spreading in many areas. Some interesting research work on and related to semantic search are: Moldovan et al. in [4] have discussed about improving the search quality of traditional search engines by using WordNet, and later work by Guha et al. [5] mentioned how relationships of objects on the web documents can be established and exploited for semantic search. More recently, researchers in [6 - 8] have shown different approaches toward semantic search incorporating existing ontologies, taxonomies and natural language processing techniques. Lopez et al. in [6] have addressed a question-answering system which takes queries expressed in natural language and an ontology as input and returns answers drawn from the available semantic markup.

Existing semantic search applications are based on machine readable electronic documents. While electronic, our documents are not readily machine readable but mostly human readable. This makes available semantic web engines unsuitable for the task at hand and necessitates custom solution. We address handling of documents containing descriptions of medical device adverse events. These include the initial report, follow-up reports, and communications from FDA to the manufacturers and user facilities. As our tool deals with large number of documents, we use certain techniques to ensure quick response to queries while keeping it scalable. This section outlines few of them.

IR systems

Information Retrieval (IR) systems aim to identify documents that are relevant to a given query among the documents available for search, typically ranked in some order of relevance. Sometimes, they point out the location of query or its related terms within the document. To this end IR systems [9, 10] address issues concerning representation, search and manipulation of large collection of electronic documents. Here, we are concerned with documents containing narrative text. Some popular and widely used IR systems are web search engines like *Google*, *Bing*, and *Yahoo*. Additionally digital library based services enable researchers, academia and medical practitioners to learn about new research articles published in their respective areas. IR systems are not limited to web search engines and digital libraries, but extend to desktop searches to specially designed enterprise level search systems. A typical IR system can be used to address multiple concerns including *``document routing/filtering'', ``text clustering and categorization'', ``text summarization''* [11], *``information extraction''* [12], *``topic detection and tracking''* [13].

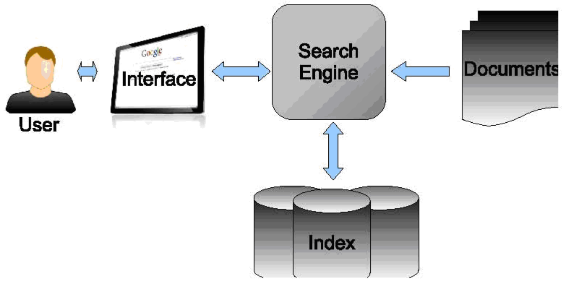


Figure 1: Components of an IR System

Core components of a typical IR system are shown in Figure 1. A search engine maintains an index of the document contents that need to be searched. Users issue queries to the IR system through user interface. Queries are typically made of terms (Note: We use “terms” instead of words due to the way indices are generated. For instance a term may be combination of number, date, wildcard characters etc.). The search engine processes the query and responds with a list of relevant documents that contain terms matching the query. The returned document list as the result of a query is ranked based on a ranking algorithm.

Inverted File Index

Searching the contents of actual documents each time a query is received is resource intensive and time consuming. Therefore IR systems typically maintain an index of documents to speed up the process. Among various indexing strategies, use of inverted file indexing [14] is well suited for large document collections. Queries are run against the inverted file index which tends to be much smaller than the documents themselves and hence results in quick identification of query terms in the index. Each term then points to documents that contain those terms. Additional information such as the term frequency helps in ranking the returned documents in specific order.

In its simplest form inverted file index maintains the mapping of terms and their location in a text collection. For instance a text document can be thought of as a collection of m words. It is made up of a sequence of n unique words such that $n<=m$. The number n is usually far less than m as most of the words is repeated while forming a document. For instance the word ‘the’ is repeated several times in this paper. The set of unique words within an index forms the "Term List" υ of the index. If a pointer (say numeric location) is associated with each word in υ to the location of that word in text document, the resultant data structure is a form of inverted file index. As the document collection grows, the number of documents matching a word in the index becomes sparser.

Text Mining and Semantic Search

The ability to retrieve related information from narrative texts enables performing more complex operations like text mining. The aim of text mining is uncovering hidden information from text documents [11]. Such hidden information could be discovered from contextual or semantic or ontological relationships among documents as reviewed in [15].

In addition to IR, techniques like Information Extraction (IE), Data Mining, Natural Language Processing (NLP) etc are used in text mining [16].

Semantic Search

A semantic search application takes user query and it returns top-k of the most conceptually relevant documents. The main phases of a semantic search could be summarized as (i) *Query Expansion*, which converts the searcher's query to a Semantic query. Some of the works on query expansion include Mitra et al. [17] and [18]. Such techniques help in increasing both recall and precision values. Techniques for fuzzy and proximity based searches increase recall but may reduce overall precision. (ii) *Search Space*: This is created during the indexing of the document collection. Techniques like stemming, or substitution of word using ontology or synonyms, domain specific tables are common during this phase. Some relevant works include [19, 7]. The cons of such method are that there is a chance of loosing the original context while replacing the word in the original document. (iii) *Searching and Ranking*: This phase depends on how the documents are modeled. Vector Space Model [20] is a common model and our application is also based on this model. (iv) *Presentation*: This phase is about how the search results are presented to the searcher and it depends on the requirement of the application. Popular display format used by Google, Yahoo etc are in one category while newer semantic search applications like Flamenco [21] may be considered as another category of display; however, the differences are blurring with time.

3 PROBLEM

Given a set of documents *D* and a set of query terms *Q* the problem is to select an ordered list of documents $L_{D'}$ such that the set of documents *D'* forming $L_{D'}$ is subset of *D*

$$D' \subseteq D$$

Furthermore documents in $L_{D'}$ should be ordered by rank(*R*) or decreasing order of relevancy with respect to *Q*. Since *R* is subjective to the specific needs of a user, we attempt to quantify *R* as a function *f* of term frequency (TF) and inverse document frequency (IDF) [22] of *Q* in the $L_{D'}$.

$$TF(d) = \frac{No.\ of\ times\ Q\ appears\ in\ d}{Total\ no.\ of\ terms\ in\ d}, \quad d \in D'$$

$$IDF = \frac{No\ of\ documents\ in\ D}{Total\ no.\ of\ documents\ containing\ Q}$$

$R(d) = f(TF, IDF)$, where *f* is a function on *TF* and *IDF*.

The problem is to select $L_{D'}$ such that following optimization requirements are met:

Precision Maximization: Given a set of documents (D_r) such that $D_r \subset D$ and D_r is the set of established relevant documents with respect to Q and D' is the set of documents selected by the system as relevant documents with respect to Q. Precision *P* is defined as

$$P = \frac{D^{'} \cap D_r}{D^{'}}$$

P quantifies the measure the fraction of documents in $L_{D'}$ that is relevant. The requirement is to maximize P

$$P \approx 1.$$

Recall Maximization: Given a set of documents (D_r) such $D_r \subseteq D$ and D_r is the set of established relevant documents with respect to Q and D' is the set of documents selected by the system as relevant documents with respect to Q. Recall *Rec* is defined as

$$\mathrm{Re}c = \frac{D_r \cap D'}{D_r}$$

Rec quantifies the measure the fraction of relevant documents that appear in result set $L_{D'}$. The requirement is to maximize *Rec*

$$\mathrm{Re}c \approx 1.$$

4 APPROACH

We now present our approach to semantic mining. Figure 2 shows a high level overview of SARF. *Indexing* and *Searching* form the core components of SARF.

Indexing module accepts documents and associated meta-data of a corpus to generate its index. Similarly, indices for related dictionaries, ontologies, and synonyms are also generated. These indices are stored in the *index repository*. Users can issue queries to the Searcher module via the *query interface* provided by the Semantic Search Tool (SST). Statistical information related to document repositories and their indices are generated and are made available through the user interface of SST.

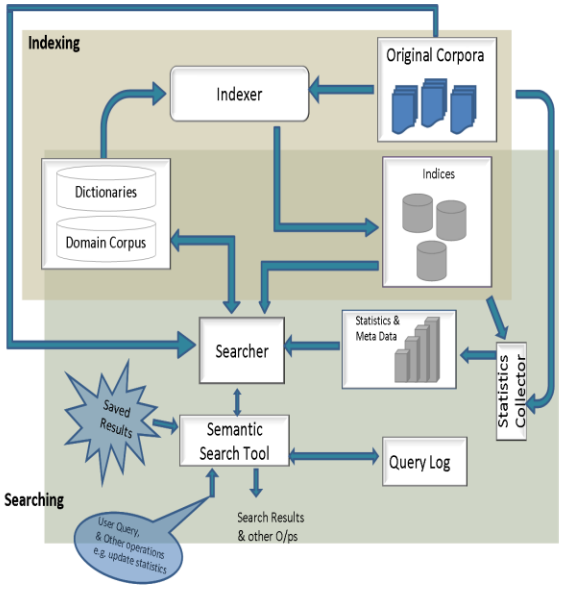


Figure 2: System Architecture

SARF is designed to handle multiple document repositories. As shown in Figure 2, indexing of each repository is done while taking into account relevant meta-data and it is done by extending and customizing Lucene APIs [3]. As new documents keep getting added in each of the repository, our indexer updates corresponding indices periodically to account for them. Thus those new documents are available for searching. The frequency of such updates is repository dependent.

Our searcher supports Lucene query syntax. Searcher has overloaded methods to handle a variety of search requests including filtering and sorting options. It is designed to search across several indices and is multithreaded for enhanced performance. Documents matching the query are ranked by the searcher using TF-IDF based relevance ranking algorithm.

SST supports varying levels of query complexity from simple pattern matching queries to complex ones requiring query expansion, boosting, sorting as well as filtering. For example, user could expand the query to include synonyms using say WordNet. SST accepts the results and displays them to the user. SST has the ability to present the results in multiple ways. For example users may choose to view results chronologically rather than using relevance ranking. The result set presented displays select segments from the narrative text containing query terms. After going through the result set users have the option to view the entire document using a clickable link. On clicking the document link, the query interface fetches the document from the document repository for viewing.

Specific information like the earliest and latest date is used as optional search options for filtering purposes. Documents do contain multiple narrative fields. For example in MedWatch 3500 form [23], which is for reporting medical device related adverse events, the narrative field "Event Description" is perhaps the most important ones for text mining. In case of follow-up reports the narrative field "Manufacturer Narrative" could be more important than other narratives. Thus ranking of the documents can be dynamic according to the type of the document. This concept is extended to all the corpora.

SST also has components for logging user queries, exporting search results and user account/access management. Online help and documentation are available. All user queries are maintained in a querylog. Querylog is designed to serve two purposes - firstly it enables 'autosuggestion' feature and secondly enables understanding of usage pattern to help optimize user interface in future. Account management module addresses management of user accounts and access control issues of the framework.

In summary, SARF with SST is a customizable, scalable and web-enabled system addressing the needs of handling different types of documents for text mining.

5 CASE STUDY

Reviewers at the Center for Devices and Radiological Health spend days, going through narratives of medical device adverse event reports, to identify reports relevant to specific issue being looked into. Typically, even among the filtered reports based on structured information like date range, and product code, only less than ten percent of them are relevant. With our tool, it is now possible to identify those ten percent relevant documents within minutes, and the reviewer needs to look only into those documents. In a specific instance, a reviewer had to go through over 2000 documents to look for specific information. The reviewer found 203 documents that had the relevant information, in 4 days. By using the same criteria used by the reviewer to identify the relevant documents, our tool returned the same 203 documents in couple of minutes. Most of the time was in getting the query right and few seconds for getting the search results. Use of SARF could save as much as 90% of time, which is spent in locating relevant documents. We believe that reviewers responsible for signal detection and adverse event analysis would benefit the most.

6 DISCUSSION AND CONCLUSIONS

In this paper, we presented a generic, practical and scalable approach to assist decision makers/regulators in searching for relevant textual information from large scale data repository (in terabytes). Our approach is based on text mining and it is independent of the type of document.

Our SST with SARF is capable of analyzing millions of regulatory documents to provide FDA analysts with an intuitive web-based tool for signal detection and evaluation. Our approach utilizes a range of techniques innovatively, including use of structured information available in the document for sorting and filtering, while utilizing the narrative unstructured information for context and semantics. Semantic analysis is achieved using, but not limited to, dictionaries, ontologies, and standards.

Our tool is powerful and it provides a wide range of queries ranging from simple to very strict. Strictness is obtained using meta-data information or query syntax or in combined. Historical user queries are maintained and used as suggestions to future users. User can export the selected search results and view details later by importing them using the application. The tool also provides dictionaries and look-up table which are specific to repository domain to assist users in constructing appropriate queries. Adding new dictionary or look up table is very easy. Most frequent words in the top-k search results are shown to the users which in turn help in reformulating new queries. Our application serves common trend of querying i.e. starts with a naive query and narrow down the search with the help of the meta-data or with using those frequent words. Search results can be displayed in interactive time series and relevance scores graphs. These features are very help to the regulators.

While automatic query modification has its advantages but it also suffers from the problem of synonyms not controlled well. It means that by letting allowing the system to select automatic modification lets using incorrect concepts/semantic and that reduces the precision. But regulators are domain experts in most of times, hence allowing them selecting the appropriate synonyms would be more helpful thereby reducing the unwanted results i.e. increasing the precision. This is exactly our application provides. Hence user has the option to start with automatic and rejects the unwanted concepts system suggested from the query. Searching across multiple corpuses is also allowed in our approach but the result for this part is now mentioned in the paper due to confidentiality of the data.

ACKNOWLEDGEMENT

This work is supported in part by an appointment to the Research Participation Program for the US Food and Drug Administration, Center for Devices and Radiological Health, Office of Science and Engineering Laboratories, administered by the Oak Ridge Institute of Science and Education and an agreement between the Department of Energy and FDA.

REFERENCES

[1] Weekly Epidemiological Record, 7 July 2006 http://www.who.int/wer/2006/wer8127.pdf

[2] S D Sudarsan, R P Jetley, B Fitzgerald and S Ramaswamy, "Implementing an information retrieval for an organizational repository" ApacheCon US 2009, 2-6 November 2009

[3] Apache Lucene, http://lucene.apache.org/java/docs/index.html

[4] D I Moldovan, and R Mihalcea, "A WordNet-Based Interface to Internet Search Engines" In Proc of the Eleventh Intl. Florida Artificial Intelligence Research Society Conference FLAIRS-98, pp. 275-279

[5] R Guha, R McCool and E Miller, "Semantic Search" In Proc of the Twelfth Intl Conf on World Wide Web, WWW'03, pp.700-709, ACM Press 2003

[6] V Lopez, M Pasin and E Motta, "Aqualog: An ontology-portable question answering system for the semantic web" In Proc of the 2nd European Semantic Web Conference, Greece 2005

[7] P M Kruze, A Naujoks, D Roesner and M Kunze, "Clever search: A wordnet based wrapper for internet search engines" Journal of Computing Research Repository - CORR, vol. abs/cs/050, 2005

[8] H Zhao, W Meng, Z Wu, V Raghavan and C Yu, "Fully automatic wrapper generation for search engines" In proc of the 14th Intl Conf on World Wide Web WWW'05, pp.66-75

[9] E S de Moura, G Navarro, N Ziviani and R Beaza Yates, "Fast searching on compressed text allowing errors" In proc of the 21st annual intl ACM SIGIR conf on Research and Development in information retrieval SIGIR'98, pp.298-306

[10] R A Baeza Yates and B Ribeiro Neto, "Modern Information Retrieval" Addison-Wesley Longman Publishing Co., Inc., Boston, MA USA 1999

[11] M A Hearst, "Untangling text data mining" In proc of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics" ACL'99 pp.3-10

[12] C H Chang, M Kayed, M R Girgis and K F Shaalan, "A survey of web information extraction systems"

IEEE Trans on Knowledge and Data Eng., 18:1411-1428, Oct'06

[13] J Allan, editor, "Topic detection and tracking: event-based information organization" Kluwer Academic Publishers, Norwell MA USA 2002

[14] Moffat, Alistair, and Z Justin, ”Self-indexing inverted files for fast text retrieval” ACM Trans. Inf. Syst. Vol. 14, issue 4 October 1996

[15] S Anna, A Periklis and N Nicolas, “Overview and semantic issues of text mining” SIGMOD Rec, vol.36, issue 3, September 2007 ACM

[16] Mooney, R Bunescu, “Mining knowledge from text using information extraction” SIGKDD Explorations, 2005, pp.3-10

[17] M Mitra, A Singhal, C Buckley, “Improving automatic query expansion” Proc. of the 21st annual intl ACM SIGIR conf on Research and development in information retrieval, SIGIR '98 pp.206-214

[18] C Mangold, “A survey and classification of semantic search approaches” Intl. J. Metadata Semant. Ontologies, vol. 2, issue 1, September, 2007, pp.23-34

[19] W Dakka, P Ipeirotis and KR Wood, “Automatic construction of multifaceted browsing interfaces” Proc of the 14th ACM intl conf on Information and knowledge management, CIKM '05, pp.768-775

[20] Vector Space Model (VSM) http://en.wikipedia.org/wiki/Vector_space_model

[21] Flamenco http://flamenco.berkeley.edu

[22] J Sparck Karen, “A statistical interpretation of term specificity and its application in retrieval” Document retrieval systems 1988 pp.132-142

[23] MedWatch 3500 Form http://www.fda.gov/Safety/MedWatch/HowToReport/DownloadForms/default.htm

On using database techniques for generating ontology mappings

Carlos R. Rivero
University of Sevilla, Spain
carlosrivero@us.es

Inma Hernández
University of Sevilla, Spain
inmahernandez@us.es

David Ruiz
University of Sevilla, Spain
druiz@us.es

Rafael Corchuelo
University of Sevilla, Spain
corchu@us.es

***Abstract*—In the Semantic Web, there are a variety of ontologies, which motivates the need for integrating them. Integration tasks rely on the use of relationships amongst the integrated ontologies, known as mappings. The literature reports on a number of techniques to automatically generate such mappings, unfortunately, the results are not suitable to perform integration tasks since they can produce incoherent data. The database community has devised techniques that automatically generate integration-enabling mappings, however, these techniques are not directly applicable to the semantic web context due to the inherent differences between ontologies and database models. In this paper, we present the differences between semantic web ontologies (RDF, RDFS and OWL) and nested relational schemata, which argue to develop new techniques to generate integration-enabling mappings. Furthermore, we analise the requirements to generate integration-enabling mappings in the context of the Semantic Web.**

***Index Terms*—Database technologies for semantic web; Ontology mediation.**

I. Introduction

Ontologies are suitable to solve semantic heterogeneity problems that can arise when two or more user communities share their knowledge [37]. However, in a distributed, evolving and open-world environment such as the Semantic Web, there are a variety of ontologies. This implies that using ontologies per se does not avoid semantic heterogeneity problems since they appear again when user communities share their different ontologies [12]. In this paper, we focus on semantic web ontologies, i.e., ontologies that are specified using RDF, RDFS or OWL [3].

As a result, the Semantic Web comprises heterogeneous and distributed ontologies, and there is a need to integrating them [4], [25]. Without an exception, integration tasks rely on the use of mappings, which are formulae represented in some logical formalism that relate the integrated ontologies [12]. Integration tasks include data translation and data integration. On the one hand, the data translation task is the process of moving the data that is stored at a number of source ontologies to a target ontology [9], [13], [14], [32]. On the other hand, the data integration task is the process of answering a query over a target ontology using only the data stored at the sources [17], [28], [40], [42]. In this paper, we focus on the data translation task.

Specifying hand-crafted mappings is tedious and error-prone since it leads the user to a frustrating trial-and-error loop: the user specifies the mappings and tests if their behaviour is correct; otherwise, the user has to rewrite the mappings and starts the loop again [30]. Note also that the costs of maintaining hand-crafted mappings are very high [43]. Therefore, automatic mapping generation relieves users from the burden of specifying hand-crafted mappings and reduces integration costs [23], [43]. There exists a large amount of literature that study how to generate these mappings automatically [8], [12], [20], [34]. Unfortunately, the mappings these techniques generate are not suitable to perform integration tasks since they can produce incoherent target data when considered in isolation or restrictions are not taken into account [23], [14], [32]. To solve these problems, the database community has devised techniques that automatically generate integration-enabling mappings based on correspondences, which are simple relationships amongst elements in the integrated database models [14], [32]. Note that our research focuses on nested relational schemata, which is a common abstraction for relational, XML and other hierarchical, set-oriented models [14]. Furthermore, nested relational techniques cannot be directly applied to the ontologies due to the inherent differences between ontologies and database models [16], [24], [26], [41].

In this paper, our goal is to support that techniques to generate integration-enabling mappings for nested relational schemata are not directly applicable to the semantic web case, due to inherent differences between database models and ontologies. As a result, we present a number of requirements for generating integration-enabling mappings in the semantic web context, and we analise them based on the inherent differences.

The paper is structured as follows: Section II describes the related work. Section III presents preliminaries regarding nested relational schemata and ontologies. In Section IV, we describe a number of inherent differences between nested relational schemata and semantic web ontologies. In Section V, we present a motivating example to discuss about why mapping generation techniques for nested relational are not directly applicable to semantic web ontologies, and the requirements for generating integration-enabling mappings in the context of the Sematic Web. Finally, Section VI summarises our main

Supported by the European Commission (FEDER), the Spanish and the Andalusian R&D&I programmes (grants TIN2007-64119, P07-TIC-2602, P08-TIC-4100, TIN2008-04718-E, TIN2010-21744, TIN2010-09809-E, TIN2010-10811-E, and TIN2010-09988-E).

conclusions.

II. Related work

In the database context, correspondences are represented in multiple ways [32], [34]. Correspondences relate the most simple entities in the source and target models, e.g., a column in the relational model or an attribute in the nested relational model [34]. Furthermore, correspondences can relate one or more entities in the source with one or more entities in the target, so, they are of four types, namely: 1:1, n:1, 1:n, n:m.

Popa et al. [32] used one of the most simple form of correspondences: a logic equality between an attribute in the source and an attribute in the target (1:1 correspondences). The mapping system developed by Mecca et al. [21] handled a more general form of correspondences: they relate a number of source attributes with a target attribute via a transformation function (n:1 correspondences). Their mapping system also allows to restrict the way the correspondences must be combined. Raffio et al. [33] proposed correspondences not only between attributes but also between nodes.

Integration-enabling mapping generation techniques for nested relational schemata produce three different types of mappings, namely: basic, nested and laconic/core. The technique devised by Popa et al. [32] generates basic mappings, and it combines logically related correspondences in which referential integrity constraints and nesting are made explicit. Fuxman et al. [14] proposed to use nested mappings to improve the results of basic mappings, which have a number of problems such as the inefficiency in their execution or the redundancy in their specification. They generate basic mappings and compose nested mappings by rewriting them; this is more efficient since they factor out common expressions and reduce the number of basic mappings used to translate the data. Basic and nested mappings can produce redundant target instances, which motivated the research on the generation of laconic/core mappings that produce target instances with no redundancy [21], [38]. These techniques generate basic mappings and rewrite them to produce laconic/core mappings.

Basic, nested and laconic/core mappings are translated into a suitable query language, and the resulting mappings are known as query mappings. The data translation task consists of executing the query mappings over the source to produce instances of the target. The benefits of using query mappings are that the data translation process is simplified, making it more efficient and flexible: instead of relying on complex, ad-hoc programs that are difficult to create and maintain, thanks to query mappings, the database management system is used as the transformation engine [23]. Furthermore, database management systems incorporate a vast knowledge on query manipulation, from which it is derived that query mappings can be automatically optimised and paralelised so that data translation can perform as good as possible.

Finally, Alexe et al. [2] developed a benchmark for comparing and evaluating mapping generation techniques. The evaluation criteria include the scalability of the generated query mappings and the support of various typical scenarios.

Regarding ontologies, there are a number of proposals to represent the correspondences that are based on ontologies and rules. Ontology-based proposals describe them by means of a populated ontology [10], [19], e.g., the Semantic Bridge Ontology [19] allows to relate classes, properties, and individuals. Rule-based approaches describe the correspondences as bridge rules in a specific-purpose language [5], [27], e.g., C-OWL [5] allows bridge rules that relate classes in five ways: equivalent, into, onto, incompatible or compatible.

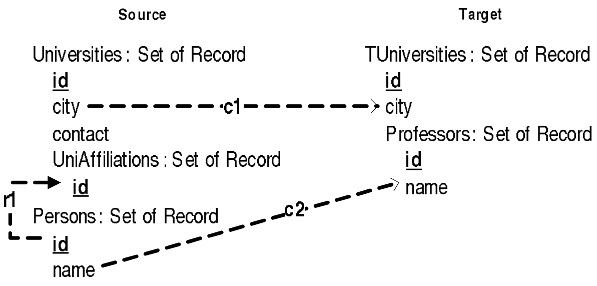
Maedche et al. [19] described a number of dimensions (aspects) for correspondences in the ontology context, namely: entity, cardinality, constraint, transformation and structural. The entity dimension establishes which entities are related, e.g., concepts, relations or attributes. The cardinality dimension covers how many entities are related ranging from 1:1 to n:m, however, the authors have found that 1:n and n:1 correspondences are sufficient in most general cases. The constraint dimension are conditions that must hold to perform an ontology data translation. The transformation dimension specifies how the values have to be translated during an ontology data translation. Finally, the structural dimension deals with how the correspondences are combined; these combinations include specialisation, abstraction, composition and alternative.

Regarding ontology mapping generation techniques, Choi et al. and Euzenat and Shvaiko [8], [12] surveyed a number of approaches to automatically generate ontology mappings. However, these ontology mappings are not suitable to perform integration tasks since they do not incorporate the model restrictions (e.g., domain or range or cardinality of properties) and do not combine logically related mappings. Early attempts that work with ontology mappings are Observer and Momis [7], [22], however, these systems focus on integration tasks and assume that mappings are available beforehand.

Regarding ontology data translation, current proposals rely on the use of ad-hoc techniques [19], reasoners [9], [36] or SPARQL query mappings [11], [29]. Regarding ad-hoc techniques, Maedche et al. [19], [18] built an execution engine in which the instances of the Semantic Bridge Ontology are evaluated to perform the data translation process. This engine is their first step towards developing a general translation technique for instances of the Semantic Bridge Ontology.

Regarding proposals based on reasoners, Dou et al. [9] perform data translation by means of a reasoner that needs a merged ontology that covers the source and target models. In this approach, the user specifies the mappings by using a first-order logic language and the reasoner is a first-order theorem prover that has been optimised for the data translation task. Serafini and Tamilin [36] work with correspondences between two classes, and their data translation process consists of reclassifying source instances into the target.

Regarding SPARQL queries, Euzenat et al. [11] presented preliminary ideas on the use of SPARQL queries to perform the data translation process. They focus on the lacks of SPARQL to work as a mapping query language [31]. Parreiras et al. [29] proposed a model-driven framework to


Source
Target
Universities : Set of Record
id
city
c1
contact
UniAffiliations : Set of Record
id
r1
Persons : Set of Record
id
name
TUniversities : Set of Record
id
city
Professors : Set of Record
id
name
c2

c1: Universities.city = TUniversities.city
c2: Persons.name = Professors.name
r1: Persons.id → UniAffiliations.id

m1: $\forall$ u · u $\in$ Universities $\Rightarrow$
$\exists$ tu · tu $\in$ TUniversities $\wedge$ tu.city = u.city
m2: $\forall$ p, u, ua · p $\in$ Persons $\wedge$ u $\in$ Universities $\wedge$
ua $\in$ u.UniAffiliations $\wedge$ ua.id = p.id $\Rightarrow$
$\exists$ tu, prof · tu $\in$ TUniversities $\wedge$
prof $\in$ tu.Professors $\wedge$
tu.city = u.city $\wedge$ prof.name = p.name

```
qm1:   <TUniversities>
       {
          for $u in $doc/Universities
          return
             <University>
                <id>generateId($u/city)</id>
                <city>$u/city</city>
             </University>
       }
       </TUniversities>
qm2: for $p in $doc/Persons,
. . .
```

Fig. 1. Examples of integration-enabling mappings in nested relational schemata

solve the automatic generation of query mappings: their pilot implementation translates hand-crafted mappings specified in OCL into SPARQL query mappings.

III. Preliminaries

In this section, we provide an example of nested relational schemata and semantic web ontologies. Furthermore, we provide examples of the concepts that we use throughout this paper.

In Figure 1, we present an example of two nested relational schemata, correspondences and mappings. The source schema comprises three nodes: $Universities$, $UniAffiliations$ and $Persons$. These nodes have a number of attributes, e.g., $city$ of $Universities$ or $name$ in $Persons$. Note that $UniAffiliations$ is nested into the $Universities$ node, furthermore, $r1$ is a referential integrity constraint that relates the id of $Persons$ and $UniAffiliations$. The target schema has $TUniversities$ and $Professors$ nodes, and $Professors$ is nested into $TUniversities$.

The relationships established between the source and the target schemata can be of three conceptual levels [15], namely:

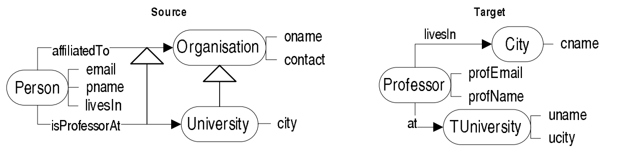


Fig. 2. Example of semantic web ontologies

- Correspondences are simple relationships amongst elements in the source and the target [32]. For example, in Figure 1, $c1$ and $c2$ are two correspondences relating $city$ and $name$ attributes in the source and target schemata. Correspondences can be given by means of an automatic matching tool [12], [34], by the user with the help of a graphical tool [1], [33], or using patterns [35].
- Integration-enabling mappings are formulae represented in some logical formalism, which combine logically related correspondences in which the inherent restrictions entailed by the modelling language are made explicit [15]. For example, in Figure 1, $r1$ is a referential integrity restriction of the model and $m1$ and $m2$ are two integration-enabling mappings, which are created based on correspondences $c1$ and $c2$, and restriction $r1$.
- Query mappings are the translation of integration-enabling mappings into a suitable query language such as SQL, XQuery or SPARQL [11], [14], [43]. Examples of query mappings include $qm1$ and $qm2$ in Figure 1.

In Figure 2, we present two semantic web ontologies that are intended to represent the same concepts as the nested relational schemata shown in Figure 1. Languages to describe semantic web ontologies allow to specify classes and properties that relate them. In our example, the classes are represented as oval shapes, e.g., in Figure 2, $Person$ or $Organisation$ are classes of the source ontology. The classes are similar in spirit to nodes in nested relational schemata.

Regarding properties, they have a domain and a range: the domain consists of a set of classes (possibly one), and the range can be either a set of classes (possibly one) or a basic data type. OWL distinguishes two types of properties: object and data properties. Object properties relate two classes, e.g., in the source ontology in Figure 2, $affiliatedTo$ means that a $Person$ (domain) is affiliated to one or more $Organisations$ (range). Data properties relate a class with a constant of a simple type, e.g., in the source ontology in Figure 2, $pname$ is the name of the $Person$ (domain), which is a String (range) but we omit it for the sake of simplicity.

Note that, when a semantic web ontology is populated, the data does not reflect all the implicit implications entailed by the modelling language. For instance, in the source ontology in Figure 2, if an instance of $University$ exists, this instance has implicitly the $Organisation$ type. Reasoners are used to make this knowledge explicit for a populated ontology [39]. This task is mandatory for some applications, e.g, when SPARQL is used to query an ontology, it is mandatory to make the knowledge explicit (in the query or in the model) because SPARQL only deals with RDF and does not implement RDFS/OWL semantics. However, there are other query languages that implement RDFS/OWL semantics, e.g., SeRQL [6].

IV. Nested relational schemata versus semantic web ontologies

In this section, we present the inherent differences we have found between the nested relational schemata and semantic web ontologies, namely:

D1 Structure and data: data, in nested relational schemata, cannot exist before a model is devised for this data, e.g., in the source schema in Figure 1, it is not possible to have an instance of $Universities$ without the schema. However, in the Semantic Web, data exists before it is modeled since data in this context amounts to pre-existing web resources, e.g., it is possible to have an URI that represents a person, but this data has no type until it is explicitly modeled to be of type $Person$ in the source ontology in Figure 2. Furthermore, several (even conflicting) models can exist for the same data.

D2 Weak existential relations: in nested relational schemata, the semantics of an attribute is that if a node instance exists, then there must exist an attribute instance for each attribute in the schema; if the value is not known, then it is null. However, in semantic web ontologies, data properties are similar to attributes but, due to the weak existential relations, if the value of a data property instance is unknown, then the data property instance does not exist. For instance, attribute $city$ of node $Universities$ in the source schema in Figure 1 has to exist and, if its value is unknown, then it is null. However, in the source ontology in Figure 2, an instance of data property $oname$ that is related to an instance of $Organisation$ only exists if the data property instance has a known value.

Furthermore, in nested relational schemata, referential integrity constraints and nesting are the mechanisms to relate nodes, which implies related nodes must exist. For example, in the source schema in Figure 1, the nesting of $UniAffiliations$ in $Universities$ implies that an instance of $UniAffiliations$ exists if and only if and instance of $Universities$ exists (but not conversely). A similar behaviour occurs with the referential integrity constraint $r1$, which implies that, if an instance of $Persons$ exists, then an instance of $UniAffiliations$ must exist (but not conversely).

In the semantic web context, object properties can be modelled as referential integrity constraints or nesting but, due to the weak existential relations, instances of object properties do not enforces the mandatory existence of instances of their domains or ranges. For example, we cannot assume that an instance of $Person$ and an instance of $University$ in the source ontology in Figure 2 are related by property $isProfessorAt$, since isolated instances of $Person$ and $University$ are allowed without no object properties instances involved.

D3 Subclasses and subproperties: in semantic web ontologies, when a class is specialised into a subclass, an instance of the subclass has also the type of the superclass, i.e., the final instance has both the type of the class and the subclass. In the source ontology in Figure 2, the specialisation of classes is represented as a white triangle with a line connecting the subclass, e.g., $University$ is subclass of $Organisation$, which means that any instance of $University$ is also an instance of $Organisation$.

When a property is specialised into a subproperty, the domain and range classes of the property are implicitly added to the domain and the range of the subproperty. In the instance level, two instances related by the subproperty are automatically related by the superproperty. In the source ontology in Figure 2, the specialisation of properties is represented with a white triangle that relates two properties, e.g., $isProfessorAt$ is a subproperty of $affiliatedTo$. If an instance of $Person$ and an instance of $University$ are related by $isProfessorAt$, they are also related by the $affiliatedTo$ property. Furthermore, $affiliatedTo$ has $Person$ as its domain, and its range is $Organisation$ and $University$.

Note that, in nested relational schemata, there is no equivalent to subclasses and subproperties. There are a number of extensions to database models that take classes and subclasses into account. However, none of them deals with subproperties or multi-type instances (see below).

D4 Instances of multiple types: in semantic web ontologies, one instance can be of multiple types without a relationship amongst them, e.g., the source ontology in Figure 2 can be populated with an instance that is both a $Person$ and $University$. Note that this is not a mistake since a type is a classifier for an existing web resource. For instance, http://en.wikipedia.org/wiki/Edsger_W._Dijkstra can be classified as both a $Person$ resource and $University$ resource.

In nested relational schemata, an instance is a set of attribute values that specify a concrete node in the schema, e.g., in the source schema in Figure 1, $Universities$ is specified by the values of id, $city$ and $contact$ attributes. Therefore, in nested relational schemata, it is impossible to have multiple types since the concept of multiple types is not supported.

D5 Properties are globally defined: in semantic web ontologies, properties are not local to a certain class but they are global to the whole ontology, e.g., assume that in the source ontology in Figure 2, we change the $pname$ and $oname$ properties by a new one called

$name$, this entails that data property instances of $name$ can have instances of $Person$, $Organisation$ or both as domain.

In nested relational schemata, attributes are local to each node, e.g., in the source schema in Figure 1, each instance of $Persons$ has an attribute id, and each instance of $Universities$ has another attribute id but they are totally different, without any relation between them.

D6 URIs: in semantic web ontologies, every class, property or instance is identified by an URI, i.e., a Uniform Resource Identifier, which can be an URL (web address) or some other type of unique identifier. In nested relational schemata, unique identifiers have to be made explicit by means of primary keys. For example, in the source schema in Figure 1, the id attributes in $Universities$, $UniAffiliations$ and $Persons$ are different.

V. Discussion

In this section, we argue that existing techniques to generate integration-enabling mappings for nested relational schemata are not applicable to the semantic web context. Furthermore, we present the requirements that such a technique should fulfill.

Assume that we use the ontologies in Figure 2 as a data translation scenario. Regarding weak existential relations, assume that we establish two correspondences that relate the source and the target of the type data property to data property (data to data), which are similar to attribute correspondences in database models:

dd1 (data to data): pname = profName
dd2 (data to data): city = ucity

We argue that data to data correspondences are not sufficient to generate ontology mappings. Recall that integration-enabling mapping generation techniques for nested relational models are based on correspondences between attributes and the inherent restrictions of these models, i.e., referential integrity constraints and nesting (cf. Figure 1). However, ontology languages do not impose any inherent restrictions since ontology properties are weak existential relations. Therefore, if a mapping generation technique for nested relational schemata is applied to our example in Figure 2 with $dd1$ and $dd2$ correspondences, then, mappings that include object properties $isProfessorAt$ and at will not be generated since the existence of instances of these object properties is not mandatory.

To solve this problem, Maedche et al. [19] introduced two new types of correspondences: object property to object property and class to class. What follows is an example of these correspondences:

oo1 (object to object): isProfessorAt = at
cc1 (class to class): Person = Professor

Another solution is to restrict the cardinality of the properties, which can be seen as enforcing the existence of an instance property. For example, if we establish that the cardinality of property $isProfessorAt$ in our source ontology is exactly one (by means of the "cardinality" construct in the OWL language), this implies that every instance of $Person$ is associated with one instance of $University$ by $isProfessorAt$.

Note that cardinality restrictions are only allowed by the OWL language but not by the RDFS language. This motivates the need to explicitly indicate that a property in an RDFS ontology is mandatory, i.e., the existence of an instance of the property is mandatory only when an instance of its domain or range exists. In our example, we make property $isProfessorAt$ mandatory by enforcing the existence of an instance of $University$, i.e., we only consider instances of $Person$ that are related to one or more instances of $University$ by $isProfessorAt$.

Regarding subclasses and subproperties and the instances of multiple types, nested relational models do not support neither of them, therefore, it is not possible to apply mapping generation techniques over ontologies with subclasses, subproperties or instances of multiple types.

After this discussion, we identify a number of requirements for the generation of integration-enabling mappings, namely:

R1 New types of correspondences: data property to data property correspondences are not sufficient to generate integration-enabling mappings because properties are weak existential relations (cf. difference D2). It is necessary at least three types of correspondences: class to class, data property to data property, and object property to object property.

R2 Explicit mandatory properties: in semantic web ontologies, properties are optional if cardinality is not restricted (cf. difference D2). However, in a concrete scenario, we may wish to have into account a property as mandatory.

R3 Subclasses and subproperties: every class and property can be specialised into a subclass or a subproperty (cf. difference D3). An integration-enabling mapping generation technique must deal with these specialisations.

R4 Multiple types, domains and ranges: instances can be of multiple types (cf. difference D4). It is also possible that the properties are multi-domain and/or -range, which is related to the global definition of the properties (cf. differences D5).

VI. Conclusions

In the bibliography, there are several techniques to automatically generate ontology mappings, which relieves users from the burden of specifying hand-crafted mappings and helps reduce integration costs. However, these resulting mappings are not suitable to perform data translation since they can lead to incoherent data. Mappings that are suitable to perform integration tasks are known as integration-enabling mappings.

The database community has devised several techniques to automatically generate integration-enabling mappings, which, as far as we know, have not been studied yet in the semantic web context. Furthermore, database techniques to generate integration-enabling mappings are not directly applicable to the semantic web context due to the inherent differences between database models and ontologies.

In this paper, we argue that more research on the generation of integration-enabling mappings is required in the semantic web context. We analise a number of differences between nested relational schemata and semantic web ontologies. Finally, we present a number of requirements to generate integration-enabling mappings.

References

[1] B. Alexe, L. Chiticariu, R. J. Miller, D. Pepper, and W. C. Tan. Muse: a system for understanding and designing mappings. In *SIGMOD Conference*, pages 1281–1284, 2008.

[2] B. Alexe, W. C. Tan, and Y. Velegrakis. STBenchmark: towards a benchmark for mapping systems. *PVLDB*, 1(1):230–244, 2008.

[3] G. Antoniou and F. van Harmelen. *A Semantic Web Primer, 2nd Edition*. The MIT Press, 2008.

[4] P. A. Bernstein and L. M. Haas. Information integration in the enterprise. *Commun. ACM*, 51(9):72–79, 2008.

[5] P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. Contextualizing ontologies. *J. Web Sem.*, 1(4):325–343, 2004.

[6] J. Broekstra, A. Kampman, and F. van Harmelen. Sesame: A generic architecture for storing and querying RDF and RDF Schema. In *International Semantic Web Conference*, pages 54–68, 2002.

[7] G. Cabri, F. Guerra, M. Vincini, S. Bergamaschi, L. Leonardi, and F. Zambonelli. Momis: Exploiting agents to support information integration. *Int. J. Cooperative Inf. Syst.*, 11(3):293–314, 2002.

[8] N. Choi, I.-Y. Song, and H. Han. A survey on ontology mapping. *SIGMOD Record*, 35(3):34–41, 2006.

[9] D. Dou, D. V. McDermott, and P. Qi. Ontology translation on the Semantic Web. *J. Data Semantics*, 2:35–57, 2005.

[10] J. Euzenat. An API for ontology alignment. In *International Semantic Web Conference*, pages 698–712, 2004.

[11] J. Euzenat, A. Polleres, and F. Scharffe. Processing ontology alignments with SPARQL. In *CISIS*, pages 913–917, 2008.

[12] J. Euzenat and P. Shvaiko. *Ontology matching*. Springer-Verlag, 2007.

[13] R. Fagin, P. G. Kolaitis, R. J. Miller, and L. Popa. Data exchange: semantics and query answering. *Theor. Comput. Sci.*, 336(1):89–124, 2005.

[14] A. Fuxman, M. A. Hernández, C. T. H. Ho, R. J. Miller, L. Popa, and P. Papotti. Nested mappings: Schema mapping reloaded. In *VLDB*, pages 67–78, 2006.

[15] L. M. Haas, M. A. Hernández, H. Ho, L. Popa, and M. Roth. Clio grows up: from research prototype to industrial tool. In *SIGMOD Conference*, pages 805–810, 2005.

[16] G. Karvounarakis, A. Magkanaraki, M. Scholl, D. Plexousakis, S. Alexaki, V. Christophides, and K. Tolle. Querying the Semantic Web with RQL. *Computer Networks*, 42(5):617–640, 2003.

[17] M. Lenzerini. Data integration: A theoretical perspective. In *PODS*, pages 233–246, 2002.

[18] A. Maedche, B. Motik, and L. Stojanovic. Managing multiple and distributed ontologies on the Semantic Web. *VLDB J.*, 12(4):286–302, 2003.

[19] A. Maedche, B. Motik, R. Volz, and N. Silva. Mafra: A mapping framework for distributed ontologies. In *EKAW*, pages 235–250, 2002.

[20] M. Mao, Y. Peng, and M. Spring. An adaptive ontology mapping approach with neural network based constraint satisfaction. *J. Web Sem.*, 2010.

[21] G. Mecca, P. Papotti, and S. Raunich. Core schema mappings. In *SIGMOD Conference*, pages 655–668, 2009.

[22] E. Mena, A. Illarramendi, V. Kashyap, and A. P. Sheth. Observer: An approach for query processing in global information systems based on interoperation across pre-existing ontologies. *Distributed and Parallel Databases*, 8(2):223–271, 2000.

[23] R. J. Miller, L. M. Haas, and M. A. Hernández. Schema mapping as query discovery. In *VLDB*, pages 77–88, 2000.

[24] B. Motik, I. Horrocks, and U. Sattler. Bridging the gap between OWL and relational databases. *J. Web Sem.*, 7(2):74–89, 2009.

[25] N. F. Noy. Semantic integration: A survey of ontology-based approaches. *SIGMOD Record*, 33(4):65–70, 2004.

[26] N. F. Noy and M. C. A. Klein. Ontology evolution: Not the same as schema evolution. *Knowl. Inf. Syst.*, 6(4):428–440, 2004.

[27] M. J. O'Connor, H. Knublauch, B. N. Grosof, S. W. Tu, W. E. Grosso, M. Dean, and M. A. Musen. Supporting rule system interoperability on the Semantic Web with SWRL. In *International Semantic Web Conference*, pages 974–986, 2005.

[28] C. R. Osuna, D. Ruiz, R. Corchuelo, and J. L. Arjona. SPARQL Query Splitter: query translation between different contexts. In *JISBD*, pages 320–323, 2009.

[29] F. S. Parreiras, S. Staab, A. Winter, and S. Schenk. Model driven specification of ontology translations. In *ER*, pages 484–497, 2008.

[30] M. Petropoulos, A. Deutsch, Y. Papakonstantinou, and Y. Katsis. Exporting and interactively querying web service-accessed sources: The CLIDE system. *ACM Trans. Database Syst.*, 32(4), 2007.

[31] A. Polleres, F. Scharffe, and R. Schindlauer. SPARQL++ for mapping between RDF vocabularies. In *OTM Conferences (1)*, pages 878–896, 2007.

[32] L. Popa, Y. Velegrakis, R. J. Miller, R. Fagin, and M. A. Hernández. Translating web data. In *VLDB*, pages 598–609, 2002.

[33] A. Raffio, D. Braga, S. Ceri, P. Papotti, and M. A. Hernández. Clip: a tool for mapping hierarchical schemas. In *SIGMOD Conference*, pages 1271–1274, 2008.

[34] E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *VLDB J.*, 10(4):334–350, 2001.

[35] F. Scharffe, J. Euzenat, and D. Fensel. Towards design patterns for ontology alignment. In *SAC*, pages 2321–2325, 2008.

[36] L. Serafini and A. Tamilin. Instance migration in heterogeneous ontology environments. In *ISWC/ASWC*, pages 452–465, 2007.

[37] N. Shadbolt, T. Berners-Lee, and W. Hall. The Semantic Web revisited. *IEEE Intelligent Systems*, 21(3):96–101, 2006.

[38] B. ten Cate, L. Chiticariu, P. G. Kolaitis, and W. C. Tan. Laconic schema mappings: Computing the core with sql queries. *PVLDB*, 2(1):1006–1017, 2009.

[39] H. J. ter Horst. Completeness, decidability and complexity of entailment for RDF Schema and a semantic extension involving the OWL vocabulary. *J. Web Sem.*, 3(2-3):79–115, 2005.

[40] S. Thakkar, J. L. Ambite, and C. A. Knoblock. Composing, optimizing, and executing plans for bioinformatics web services. *VLDB J.*, 14(3):330–353, 2005.

[41] M. Uschold and M. Grüninger. Ontologies and semantics for seamless connectivity. *SIGMOD Record*, 33(4):58–64, 2004.

[42] C. Yu and L. Popa. Constraint-based XML query rewriting for data integration. In *SIGMOD Conference*, pages 371–382, 2004.

[43] C. Yu and L. Popa. Semantic adaptation of schema mappings when schemas evolve. In *VLDB*, pages 1006–1017, 2005.

JAKARA: Developing Smart Dynamic Packaging Based Linked E-tourism Data

Afaf MERAZI[1]

[1]EEDIS Laboratory, Department Of Computer Science, Djillali Liabès University of Sidi Bel-Abbès, Algeria

Abstract— *E-Tourism has become the world's largest industry, composing of numerous enterprises such as airlines, hoteliers, car rentals, leisure suppliers, and travel agencies. It involves the buying and selling of tourism products and services via internet and includes many processes (e.g., dynamic packaging) which provide useful new options for consumers. Therefore, Intermediaries needs e-Business solutions to offer the growing demand of personalised products and services. Nowadays, The Semantic Web extends the existing Web, adding a multitude of language standards and software components to give humans and machines direct access to data such as GoodRelations. This paper describes a semantic e-Business platform for tourism package intermediaries that allow adaptation and customisation of tourism products and services to the desires and restrictions of the tourist at any time. We present JAKARA, a system designed to support convenient GoodRelations annotation and intelligent querying of linked Semantic Web resources oriented e-tourism Data.*

Keywords: Semantic Web, E-Tourism, GoodRelations Ontology, Tourism Products and Services, Semantic Query

1. Introduction

E-Tourism is a very successful and dynamic industry in the world business. In recent years, with many countries turning to tourism to supplement their economies, there has been a massive expansion of tourism vendor offerings. As more travel arrangements are made online, pressure is put on e-Tourism website developers to provide efficient and easy to use interfaces and intelligent services.

The Tourism industry is an information intensive business, and the amount of information is increasing rapidly; however, efficient access to this information is becoming a challenge. Due to the deluge of tourism vendor offerings and ever increasing numbers of travelers, the limitations associated with conventional e-Tourism websites are coming to the fore.

In general, e-tourism consists of travel planning and booking, which is the mainly successful business model on the World Wide Web. However, the task of planning an individual trip on the Web still consumes a time and seems to a complicated endeavor. Most of the huge number of tourism websites provides isolated and unified information about accommodation (hotel,...), transportation (flight,...), services (rental car,...), events (concert,...) or they relate that information in a very limited manner letting the consumer the heavy task of putting all the pieces together. There exists currently no integrated service for arranging personalized trips called to any desired destination, relying on distributed information sources which have to be reasonably combined.

Currently, with most tourism information systems, consumers need to visit multiple independent websites to plan their trip, register their personal information multiple times, spend hours or days waiting for response or confirmation, and make multiple payments by credit card. Consumers are discouraged by the lack of functionality. Dynamic packaging applications are emerging in response to these limitations and have caught the attention of major universal online travel agencies.

E-tourism is a perfect application area for Semantic Web technologies since information integration, dissemination, and exchange are the key backbones of the travel industry. Therefore, the Semantic Web can considerably improve e-tourism applications. The aim of this paper is to describe JAKARA system, which is designed to support convenient GoodRelations annotation and intelligent querying of Semantic Web resources. GoodRelations Annotator software is used by a website owner to generate RDF description describing the content of their website. The RDF description is essentially instance data that conforms to OWL e-tourism ontology. Query functions are facilitated by CPSPARQL query engine developed by [3]. The semantic query facility is accessed remotely via a Web-based GUI and provides the consumer with a number of search options. Once a query is submitted, a list of matching results is displayed to the consumer.

2. Dynamic Packaging

Dynamic Packaging [6]: "An industry buzzword for enabling the consumer (or booking agent) to build a customized itinerary by assembling multiple components of their choices and complete the transaction in real-time".

Dynamic Packaging is very different from prepackaged travel (see Figure 1). Prepackaged travel relies on selling to the customer a complete package that includes usually transportation, accommodations, car rental, activities and other services... These packages are made usually months in advance and published in brochures or sold online. These packages allow the different actors of the travel industry from producers to resellers to offer "mass-market" products and to operate relatively simple business processes that allow them to have higher margins.

These "mass packages" offer: Inflexible dates, Fixed itineraries, Very limited options. But as they are made months in advance, they also often hinder the optimization of revenues through yield-management techniques that are based on adjusting price and availability to demand in real-time. In Dynamic Packaging, the process is different even if the result could seem to be the same to the end customer: here, the components are "drawn from the inventories of the travel producers and combined to satisfy a particular customer requirement, which is collected during an interactive dialog".

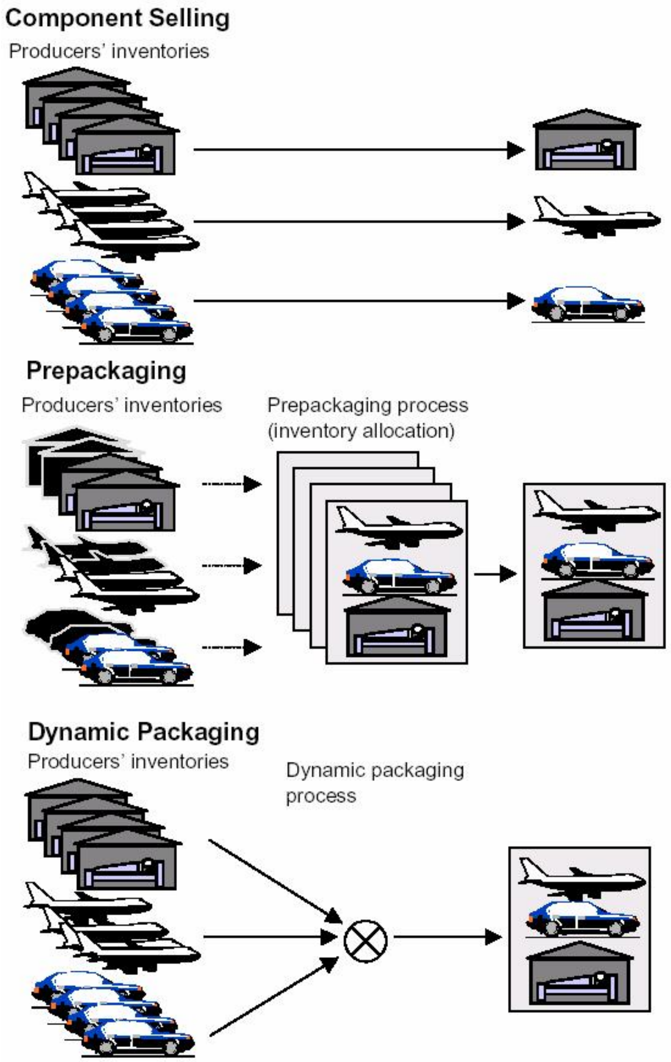


Fig. 1: Difference between dynamic packaging and component selling

Smart Dynamic Packaging is a collection of several services together in real-time according to availability, product-specific rules, especially requests (preferences) of clients (operators, tour operator). It usually includes transportation, accommodation and food, but may also include tours, excursions and other services. Any exploration of how to create smart dynamic packages must begin with an evaluation of available travel web resources; times that activities are possible and the logistics of combining different vacation components. Good questions to explore are:

- **What** can a tourist do in my area?
- **Where** are located the interesting places to see and visit?
- **When** can the tourist visit a particular place?
- **How** can the tourist get to its destination to see or do an activity?
- **What local resources** can I combine to create attractive packages?

3. Summary of E-tourism ontologies

In tourism domain, there may already exist different in-house taxonomies and catalogues which are designed and used internally by tourism agents to help them for managing heterogeneous tourism data. Efforts are made to generate global standards to facilitate inter and intra tourism information exchange (e.g., by the WTO "World Tourism Organization" which consisting of collection and collation of statistical information on international tourism). Ontologies play an important role to facilitate semantic integration of heterogeneous data. In this section, we identify several publicly available formal tourism ontologies which show the current status of the efforts and may serve as a basis for problem specific tourism ontologies (see Figure 2).

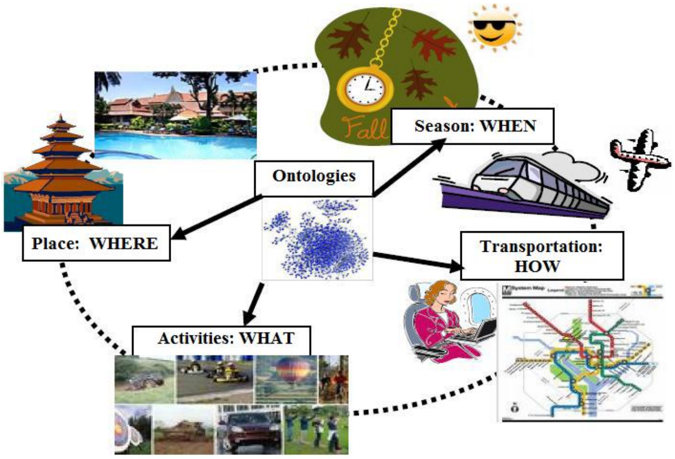


Fig. 2: The e-tourism ontologies development methodology

3.1 Harmonise Ontology

The Harmonise[1] Ontology was created within the EU Tourism Harmonisation Network (THN) established by eC-TRL (E-Commerce and Tourism Research Laboratory[2]), IFITT (International Federation for IT and Travel & Tourism[3]), and others. It is specialized to address interoperability problems in the area of e-tourism focusing on data exchange. The goal of the ontology is to allow tourism organizations to exchange data and information without changing their local data structures and information systems. Harmonise is an ontology-based mediation by mapping different tourism ontologies. The used ontology language is RDF/RDFS and the ontology itself contains about 200 concepts mainly dealing with accommodation and events.

3.2 Mondeca Tourism Ontology

Mondeca[4] Tourism Ontology includes important concepts of the tourism domain which are defined in the WTO thesaurus[5] managed by the WTO (World Tourism Organization). These concepts include terms for tourism object profiling, tourism and cultural objects (place, museum, restaurant, housing, transportation, events, etc...), tourism packages and tourism multimedia content. The used ontology language is OWL and the ontology itself contains about 1000 concepts.

3.3 OnTour Ontology

The OnTour Ontology[6] is an ontology created especially for the tourism domain and was developed by DERI (Digital Enterprise Research Institute). In addition to normal tourism concepts (location, accommodation) it also includes concepts that describe leisure activities and geographic data. The used ontology language is OWL-DL[7].

3.4 OpenTravel Alliance (OTA)

The OTA (Open Travel Alliance)[8] members are organizations that represent all segments of the travel industry, along with key technology and service suppliers. The OTA Specification defines XML Message Sets packages that contain about 140 XML Schema documents corresponding to events and activities in various travel sectors (booking, availability checking, reservation and cancelling of flights, hotels and rental cars).

3.5 Other Ontologies

A Tourism Ontology[9] developed by the University of Karlsruhe contains four different sub-ontologies for the tourism domain defining about 300 concepts and more than 100 relations. The EON Travelling Ontology[10] is mainly designed for the travel domain developed by INA (Institut National de l'Audiovisuel - France). The TAGA Travel Ontology[11] is another travel focused ontology that provides typical concepts of travelling combined with concepts describing typical tourism activities such as itineraries, customers, travel services, and service reservations.

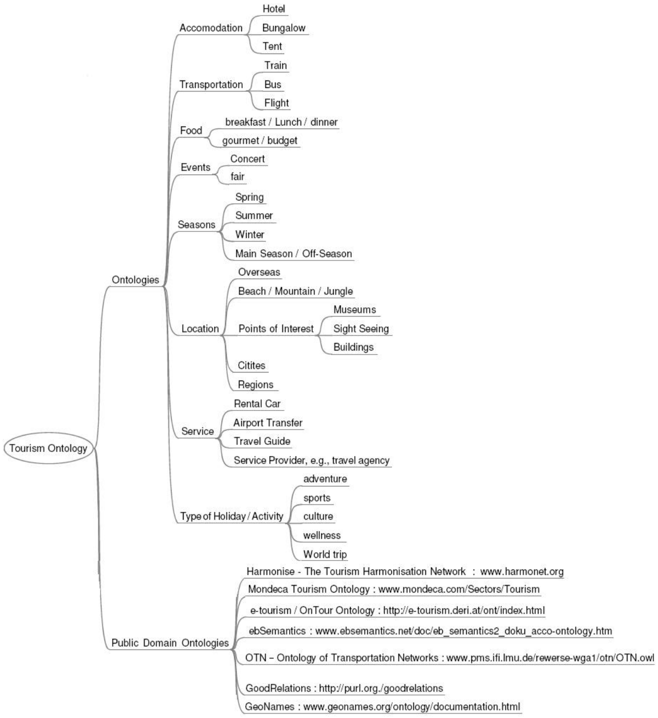

Tourism Ontology
Ontologies
Accomodation
Hotel
Bungalow
Tent
Transportation
Train
Bus
Flight
Food
breakfast / Lunch / dinner
gourmet / budget
Events
Concert
fair
Seasons
Spring
Summer
Winter
Main Season / Off-Season
Location
Overseas
Beach / Mountain / Jungle
Points of Interest
Museums
Sight Seeing
Buildings
Citites
Regions
Service
Rental Car
Airport Transfer
Travel Guide
Service Provider, e.g., travel agency
Type of Holiday / Activity
adventure
sports
culture
wellness
World trip
Public Domain Ontologies
Harmonise - The Tourism Harmonisation Network : www.harmonet.org
Mondeca Tourism Ontology : www.mondeca.com/Sectors/Tourism
e-tourism / OnTour Ontology : http://e-tourism.deri.at/ont/index.html
ebSemantics : www.ebsemantics.net/doc/eb_semantics2_doku_acco-ontology.htm
OTN – Ontology of Transportation Networks : www.pms.ifi.lmu.de/rewerse-wga1/otn/OTN.owl
GoodRelations : http://purl.org./goodrelations
GeoNames : www.geonames.org/ontology/documentation.html

Fig. 3: Tourism ontology dimensions

The following code is showing an example of e-tourism ontology which includes classes Accommodations, Guestroom, etc. and where the Accommodations classes have object property called hasRoom.

```
<owl:Ontology rdf:about=""/>
<owl:Class rdf:ID="Accomodation">
<rdfs:subClassOf>
<owl:Restriction>
<owl:onProperty>
<owl:ObjectProperty rdf:ID="hasRoom"/>
</owl:onProperty>
<owl:someValuesFrom>
<owl:Class rdf:ID="Guestroom"/>
</owl:someValuesFrom>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf....
```

[1] http://www.harmonise.org

[2] http://ertrl.itc.it:8080/home/index.jsp

[3] http://www.ifitt.org

[4] www.mondeca.com

[5] The WTO Thesaurus includes information and definitions of the topic tourism and leisure activities. www.world-tourism.org

[6] http://ontour.deri.org/ontology/ontour-02.owl

[7] http://e-tourism.deri.at/ont/index.html

[8] http://www.opentravel.org/

[9] http://ontobroker.semanticweb.org/ontos/comparing.html

[10] http://opales.ina.fr/public/ontologies/EON-TravellingOntology-v0.2.daml

[11] http://taga.sourceforge.net/owl/travel.owl

4. GoodRelations: An Ontology for Describing Web Offers

Semantic Web + E-Commerce = GoodRelations

Representing the semantics of data is a need that has recently emerged from the Semantic Web area which intends to bring meaning to the unprecedented and tremendous amount of data published over the Web. Tim Berners-Lee presents the Semantic Web [5] as "not a separate Web, but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation". He also conditions its success to an automated reasoning: "computers must have access to a structured collections of information and sets of inference rules that they can use to conduct automated reasoning [7]". The Semantic Web is based on Ontologies - languages to formally describe domain specific concepts and their relations - coupled to reasoning engines, to perform domain fact deductions. A promising application domain for Semantic Web technology is the annotation of products and services offers on the Web so that consumers and enterprises can search for suitable suppliers using products and services ontologies. Martin Hepp introduces GoodRelations Ontology to allow for each company who possesses a website to describe and represent all their products and services offering for intelligent search and deterministic response End user. We give an overview of some definitions about this domain application:

"The GoodRelations ontology[12] provides a conceptual model for a consolidated view on commerce data on the Web, e.g.: companies, store locations, offers, product descriptions, pricing, payment, shipment, and warranty information. It is provides the vocabulary for annotating e-commerce offerings (see Figure 4)." [9]

"The GoodRelations ontology provides a generic yet lightweight vocabulary for describing in a machine-readable way the details of offers made on the Web. This allows vendors to encode their offers so that Semantic Web search engines can find such Web resources precisely. It empowers them to return exactly matching offers for a given need."

"GoodRelations is a lightweight yet sophisticated vocabulary that allows manufacturers and show operators to express the exact meaning of their offers made on the Web in a machine-readable way. This empowers search engines to support more precise search, and partners in the value chain to automate their content integration tasks. (1) to sell, lease, repair, dispose, and maintain commodity products and (2) to provide commodity services. GoodRelations allow describing the relationship between (1) Web resources, (2) offerings made by those Web resources, (3) legal entities, (4) prices, (5) terms and conditions, and the aforementioned ontologies for products and services[13] (6)." [8]

In the following, we define an overview of the relevant conceptual entities by giving an informal definition and additional explanations in the Table 1.

Table 1: The most important conceptual elements of GoodRelations.

Feature	Description	Reference
Business Entity.	present the company Web site (mailing address and contact details).	*gr:BusinessEntity*
Products and Offerings.	describe the actual products or services that are being offered, and the offering, the actual business function (sell, repair, dispose, etc.) and other commercial properties.	*gr:ProductOrServiceModel*
Eligible Customers and Regions.	The types of customers for which the given Offering is valid.	*gr:eligibleCustomerTypes*
Price Specifications.	information on the prices fixed by vendors.	*gr:hasPriceSpecification*
Delivery Options and Delivery Charge Specifications.	shipment charge.	*gr:availableDeliveryMethods*
Payment Options and Payment Charge Specifications.	procedure for transferring the monetary amount for a purchase	*gr:acceptedPaymentMethods*
Warranty Promises.	offer includes a bundle of services in case of defects or malfunction.	*gr:hasWarrantyPromise*
Bundles.	placeholder instance for unknown instances of a mass-produces commodity.	*gr:ProductOrServicesSome InstancesPlaceholder*
Services and Value Ranges.	numerical interval that represents the range of a certain quantitative Product or Service Property in terms of the lower and upper bounds for one particular Product Or Service.	*gr:QuantitativeValueFloat* *gr:TypeAndQuantityNode*
Shop Locations and Opening Hours.	availability of a particular Offering at or from and defines the opening hours for a given DayOfWeek.	*gr:availableAtOrFrom* *gr:hasOpeningHoursSpecification*

Every object called resource in RDF has a global identifier in the form of a URI. The design of Linked Data [4] provides recommendations on how to publish both kinds of resources, and it is also an initiative on publishing data from various closed databases in RDF. Linked e-tourism Data is

[12] http://www.heppnetz.de/projects/goodrelations/primer/20080808/

[13] We are talking about real services, not Semantic Web services or Web services.

an example of such a Linked Data database of information coming from a set of GoodRelations ontologies including people, places, organizations, agencies, etc. Each of them is one non information resource with a unique URI and its information representation. It is recommended to reuse resources from such knowledge bases instead of creating new individuals.

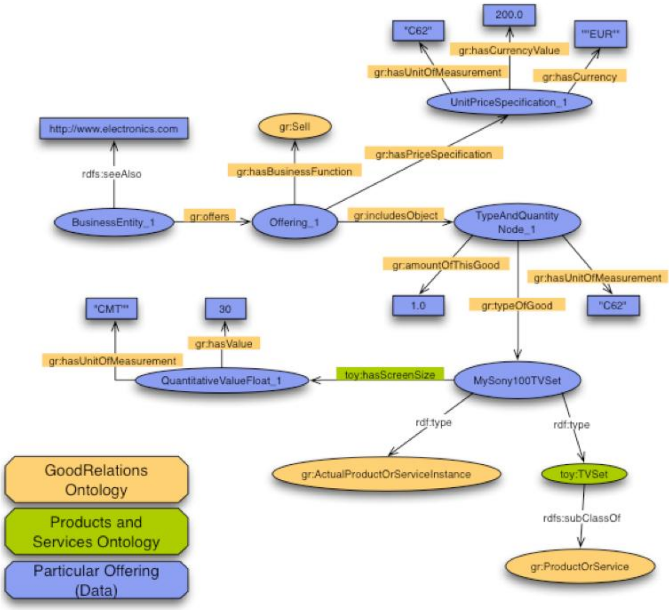


Fig. 4: RDF graph of the minimal example

5. Framework Architecture & Overview

Nowadays, E-tourism is a leading industry in e-business. Furthermore, it is viewed as information intensive industry with a long value chain where information plays an important role for decision and action making. There are many challenges to solve in the e-tourism distribution; the most important is to enable interoperability between information systems, allowing seamless information exchange between tourism organizations. Semantic technologies provide methods and concepts facilitating integration of tourism information originating from various sources on top of so-called ontologies (formal domain conceptualisations). Semantic technologies offer a significant potential for better cross-system integration and a more versatile linkage of available multimedia tourism data based on ontology references and other types of semantic mark-up (such as geo-referencing of data).

Generally, The interoperability problem between cooperating systems can be differentiated into:

1) **Information-level interoperability**, which addresses clashes between different data representations and their meaning, and
2) **Service-level interoperability**, when different processes should cooperate to automatically perform business between enterprises.

Our framework concentrates at the information level interoperability problem and proposes solution for this level only. Information heterogeneity between different systems can be categorized as follows:

- Semantic clashes: addressing different meaning of concepts of different systems, and
- Representational clashes: caused by different data representations formats With the respect to the impact on the data exchange, structuring conflicts can be differentiated into:
- Fully mappable: all clashes can be resolved without any loss of information, and
- Partially or nonmappable: covering the structural conflicts for which any conceivable transformation will cause a loss of information.

In this paper, we analyze the complexity of product description in the tourism distribution and propose to use the GoodRelations ontology that covers the representational needs of typical e-commerce scenarios in the commodity segment. In this section, we develop the requirements on a e-tourism products and commodity services offered on the Web.

- Due to the heterogeneity of the e-tourism product, *tourism agency consultant or a person who is planning the trip itself must have access to multiple sources of information.*
- Most e-tourism products (e.g., hotel rooms, flight tickets) are time-constrained and can't be stocked/stored. To support this, *basic products must have well-defined interfaces with respect to consumer needs, prices or distribution channels and an abstract model of the product (e.g., its description).*
- E-tourism product is also immaterial, meaning that traveler cannot see or touch the tourism product before the trip. *That is why reliable information about destination, accommodation options and other parts of the tourism product is extremely important for both people working in tourism industry and tourists themselves. Tourism product cannot be stored in storage*.
- If a hotel room or a seat in an airplane remains empty today, this is lost revenue for the tourism company. This is a reason why *effective distribution and inventory management are key factors* in the tourism business.

The purpose of JAKARA is to create a system that provides a tangible benefit over existing e-tourism Web portals by allowing tourism customers to search the underlying products or services of a website, thus producing results that more closely match the customer's needs and preferences denoting "smart dynamic packaging". This is achieved by using GoodRelations ontology "a promising application domain for Semantic Web technology" to infer knowledge about resorts and seamlessly integrating that knowledge so that it can be used by a tourism customer when searching for suitable travel.

The system is limited to the GoodRelations'annotation and querying of a specific Country travel & tourism websites. Only websites annotated consistent with the e-tourism ontology (OnTour Ontology viewed in section 3.3) employed are included in the system. Such techniques are recognized as important for the integration of tourism & travel information, but fall outside the scope of what the system aims to demonstrate. Web-page annotations are conceptually consistent products and services offers instance data of the e-tourism ontology, and are queried by the GUI using a database looks-up from a Jena back-end knowledge base. We present the percentage of websites that contain at least a sufficient amount of information in the various categories is visualized in Figure 5.

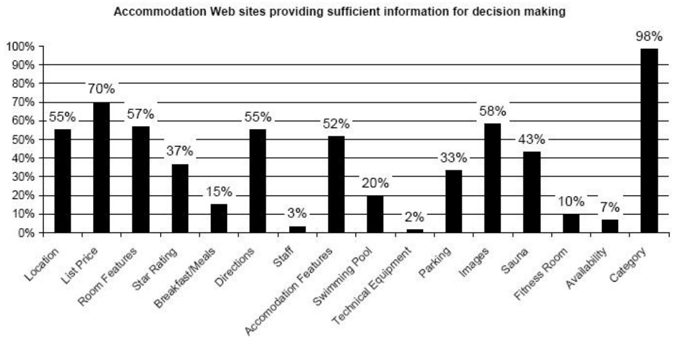


Fig. 5: Percentage of e-tourism Web sites

The JAKARA architecture (see Figure 6) is designed to support convenient annotation based product and service offers application domain and intelligent querying of Semantic Web resources.

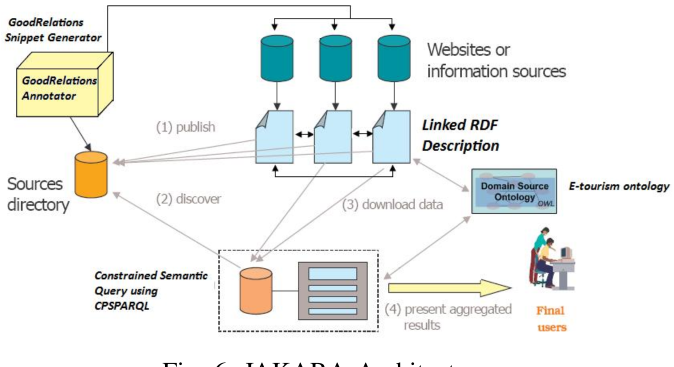

GoodRelations Snippet Generator
GoodRelations Annotator
Websites or information sources
(1) publish
Linked RDF Description
Sources directory
(2) discover
(3) download data
Domain Source Ontology OWL
E-tourism ontology
Constrained Semantic Query using CPSPARQL
(4) present aggregated results
Final users

Fig. 6: JAKARA Architecture

5.1 GoodRelations Annotator based e-tourism ontology

Annotation software is used by a website owner to generate RDF descriptions describing the content of their website, it is a form-based tool that helps owners of any business in the world produces a basic yet detailed description of its range of products or services called GoodRelations Annotator [10] "Inspired by the impact of the FOAF-a-Matic tool on the availability of FOAF data on the Web, we developed a form-based tool that can be used by any business in the world to create a basic yet detailed description of its range of products, payment and delivery options, store locations, opening hours, and eligible customer types and eligible regions[14] Internally, it uses a HTML form with a substantial amount of JavaScript for validating the input, and a Python server-side component that handles the conversion and returns RDF/XML (in the future also RDFa snippets [1]) plus instructions on how to publish the data."

The RDF description is essentially instance data that conforms to an OWL e-tourism ontology. Query functions are facilitated by a Jena based CPSPARQL[15] query engine that uses a Pellet reasoner and the linked GoodRelations ontologies to infer knowledge about the e-tourism domain. The query facility is accessed remotely via a Web-based GUI and provides the consumer. Once a query is submitted, a list of matching results is displayed to the consumer.

5.2 Querying GoodRelations offers using CPSPARQL

"SPARQL [11] can be used to express queries across diverse data sources, whether the data are stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be result sets or RDF graphs." There are several proposed extension to SPARQL:

- PSPARQL [2] which allows query graph patterns involving regular expressions. This provides a lot of flexibility in expressing graph patterns that can be matched. However, PSPARQL supports only a pattern matching query paradigm.
- CPSPARQL [3] that further extends (P)SPARQL by allowing, for example, complex constraints on nodes and edges of traversed paths.

[14] The tool is available at: http://www.ebusiness-unibw.org/tools/goodrelations-annotator/

[15] available at: http://psparql.inrialpes.fr/

We decide to implement the CPSPARQL query language to describe the preferences and needs of the consumer furthermore to create a Smart dynamic Packaging which must correspond to products and services offering requirement. In the following, we give an example in CPSPARQL on how to query respective product and services data on the Semantic Web. For example, the price of each direct trip is no more than 500:

```
PREFIX gr: <http://purl.org/goodrelations/v1#>
PREFIX ex: <http://www.heppnetz.de/ontologies/goodrelations
/examples#>
SELECT ?offering ?uri ?maxprice ?currency
WHERE { CONSTRAINT const1 ]ALL ?Trip]:
?offering gr:includesObject ?TypeAndQuantityNode .
?TypeAndQuantityNode gr:typeOfGood ?something .
?something rdf:type ex:trip .
?offering gr:hasBusinessFunction gr:Sell.
?offering rdfs:seeAlso ?uri .
?offering gr:hasPriceSpecification ?priceSpecification .
?priceSpecification rdf:type gr:UnitPriceSpecification .
?priceSpecification gr:hasCurrency ?currency .
?priceSpecification gr:hasMaxCurrencyValue ?maxprice .
FILTER (?maxprice < 500)
}
```

6. Conclusion

The Semantic Web tools and technologies are useful in developing the next generation of e-Tourism websites. Some of the benefits would be, more flexible searching of appropriate websites, less precise input from users in requesting what they want and a standard way of packaging intelligence for a suite of e-Tourism applications. Some limitations in applying the Semantic Web are due to the nature of the Internet as it stands today. For example, how much automation is it safe to apply on the Internet, given that security remain a problem?. This paper presented an overview of the Semantic Web, and described in detail a framework for developing e-Tourism applications based on the technologies of the Semantic Web.

Hence, one big challenge to successfully develop dynamic packaging applications is finding a solution to cope and integrate the nonstandard way of defining e-tourism products and services. We therefore suggested Web services and semantics as emerging technologies that can be used to deal with the lack of standard and enable data intention. These two technologies have already been recognized in the Semantic Web and can considerably improve e-tourism systems.

References

[1] B. Adida, M. Birbeck, S. McCarron, and S. Pemberton, *RDFa in XHTML: Syntax and Processing Ű A collection of attributes and processing rules for extending XHTML to support RDF*, W3C Recommendation, http://www.w3.org/TR/2008/REC-rdfa-syntax-20081014/, 2008.

[2] F. Alkhateeb, J. F. Baget, and J. Euzenat, *Extending SPARQL with regular expression patterns (for querying RDF), Journal of Web Semantics*, 7(2):57–73, Elsevier Science Publishers B. V. 2009.

[3] F. Alkhateeb, J. F. Baget, and J. Euzenat, *Constrained regular expressions in SPARQL*, in: Hamid R. Arabnia and Andy Marsh (eds), Proc. international conference on semantic web and web services (SWWS), Las Vegas (NV US), pp 91–99, 2008.

[4] T. Berners-Lee, *Linked Data*, in Design Issues, http://www.w3.org/DesignIssues/LinkedData.html, 2006.

[5] T. Berners-Lee, J. Hendler, and L. Ora, *The Semantic Web*, Scientific American, 2001.

[6] J. Cardoso and C. Lange, *A Framework for Assessing Strategies and Technologies for Dynamic Packaging Applications*. Journal of Information Technology & Tourism, 9(1), 27–44, 2007.

[7] A. Gaignard, and J. Montagnat, *Survey on semantic data stores and reasoning engines* VIP Project - ANR-09-COSI-03 - Milestone 1.2.1 pp 1–31. Version 1.1, September 14, 2010.

[8] M. Hepp. *GoodRelations: An Ontology for Describing Products and Services Offers on the Web*, Proceedings of the 16th International Conference on Knowledge Engineering and Knowledge Management (EKAW2008), September 29 - October 3, 2008 (forthcoming), Acitrezza, Italy, Springer LNCS, Vol. 5268, pp. 329–346, 2008.

[9] M. Hepp. *Product Variety, Consumer Preferences, and Web Technology: Can the Web of Data Reduce Price Competition and Increase Customer Satisfaction?*, in: T. Di Noia and F. Buccafurri (Eds.): EC-Web 2009, LNCS 5692, p. 144. Springer-Verlag, Berlin Heidelberg, 2009.

[10] M. Hepp, A. Radinger, A. Wechselberger, A. Stolz, D. Bingel, T. Irmscher, M. Mattern, and T. Ostheim, *GoodRelations Tools and Applications*, in: Poster and Demo Proceedings of the 8th International Semantic Web Conference (ISWC 2009), Washington, DC, USA, October 25–29, 2009.

[11] E. Prud'hommeaux and A. Seaborne (ed.), *SPARQL Query Language for RDF*. http://www.w3.org/TR/rdf-sparql-query/, 2008.

A Web Service Selection Mechanism based on User’s Preferences

Chin-Chih Chang and Wen-Wei Tsai
Department of Computer Science and Information Engineering
Chung Hua University
Hsinchu City, Taiwan

***Abstract*—With the fast development of Web services more and more Web services are available. The conventional issues of Information/knowledge discovery has evolved into service discovery. There is a need to discover, select and compose suitable services that meet user's requirements.**

The existing service discovery technologies are mainly keyword-based search and less emphasize on user’s preferences. We believe user’s preferences are the key factor for the decision on the service selection. In this paper we present a Web service selection mechanism based on user’s preference. The mechanism enhances the existing Web service registry (UDDI) with the sorting, ranking, and feedback capability. In each step the user's preferences are taken into consideration.

A system is built based on the mechanism and the feasibility of the mechanism is validated by a travel plan example. The performance of the mechanism is also analyzed. It shows the system is feasible and effective.

Keywords-Web Service; UDDI; Service Discovery; Service Selection; Quality of Service

I. Introduction

Web Services are currently the de facto technology for service-oriented computing (SOC) which is widely adopted in academic and industrial software development. With the fast development of Web services more and more Web services are available. There is a trend that the data-oriented Web is migrating to the Service Web [1]. This migration has led to the need for Web services discovery and selection. To search and select a suitable Web service that can meet user's requirements from the numerous Web services has become an important research topic.

The existing service discovery technologies are mainly keyword-based search which cannot fit user’s need well [2]. In a lot of business operations, users' preferences are often the decision factor. In order to offer Web services that satisfy users' preferences we present a user-centric mechanism for Web service selection. We use the existing Web service registry and integrate with the sorting, ranking, and feedback mechanisms which we proposed to construct a Web service system with emphasize on user’s preferences. When a user wants to search and select a web service, she or he can access the system interface to accomplish the task. In the ranking mechanisms, we adopt two algorithms: the rule-based and weighting methods, to select a suitable Web Service and recommend to the user. Finally, we use the feedback mechanism to adjust the selection weighting for services. When a user is using or after using the system, the system will collect the rating values for the service from the user. These feedback values are used to adjust the weighting of Web Services dynamically to provide the user with the adaptable Web Services.

We use the tourist service as an example to validate the feasibility of our system. We will show how a user utilizes our system to plan a trip according to user's preferences. Finally, we will analyze the algorithms that we present, and compare their advantages and disadvantages. And we also analyze our system and compare with other research work.

II. Service Discovery and Service Selection

Web service discovery and selection is a complex process. There is still no clear distinction between discovery and selection. Service selection begins with discovery. In this paper we adopt the view that discovery is referred as the activities related to identifying the functional properties of user’s requests and selection as nonfunctional properties [3]. Crasso et al. further enumerate four functional and four nonfunctional criteria [4]. The discovery process is mainly keyword-based search which returns a list of candidate services and the selection process is a refinement of the discovery process [5]. For the recent survey on service discovery and selection interested readers can refer to [4].

Functional criteria are mainly used to match against the specifications offered by the service provider. QoS (quality of service) is commonly used to indicate nonfunctional criteria [6]. In this paper, we focus on nonfunctional QoS because they are often the decision factor. With the development of Web services research, various QoS for Web services have been identified and can be classified into the following categories: runtime-related, transaction support related, configuration management and cost-related QoS, security-related QoS, and user-related QoS [7, 8]. From our point of view, user-related QoS is the most important factor in these nonfunctional characteristics because users are always the ones who make the final decision. Hence, we focus on Web service selection in terms of user’s preferences.

A. *Web service discovry*

The goal of Web service discovery is to find appropriate Web services that match user’s functional requirements. Web service discovery through UDDI (Universal Description, Discovery, and Integration) is the most basic discovery method. UDDI provides only a category-based browsing and keyword-based matching discovery service [8]. Discovery by UDDI is simply a process of matching the WSDL (Web Service Description Language) based on keyword and category. After matching keyword the possible services are replied but the most appropriate service is not retrieved now and then. Therefore, quite a number of approaches have been proposed to enhance UDDI. Four main categories of these approaches have been specified: information retrieval-based, QoS-aware, semantic-based, and highly scalable and available [4]. The discovery process usually finds a list of candidate services. To further identify the most suitable service for the requestor selection process is conducted.

B. Web service composition

One of powerful and highly expected capabilities is service composition. Especially for a complex goal if a single service cannot fulfill the request, the service composition is activated. Usually, the service composition is undertaken by decomposing the complex goal into a sequence of simple sub-goals. Each sub-goal can be fulfilled by a single service. The main issue of the service composition is how to decompose the task and then select suitable services to complete the task effectively and efficiently. For example, the task of booking a hotel in a foreign country can be decomposed into two tasks: exchange rate conversion and hotel booking as shown in Figure 1. Once two tasks are fulfilled, the complete task is completed.

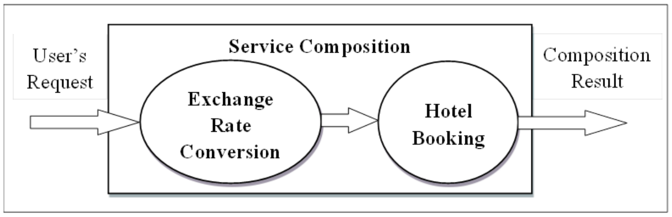


Figure 1. Example of service compostion

Basically, the service composition strategies are classified as static and dynamic composition based on the time of service composition [9]. Static composition takes place during design time while dynamic composition takes place at run time. A composition mechanism must satisfy four requirements: connectivity, nonfunctional QoS properties, correctness, and scalability [10]. The effective dynamic Web service composition is still a highly complex and challenging task [11]. Most current solutions are either too theoretic or only suitable to some specific situations. Much effort is continuously devoted to Web service composition. For a recent survey interested readers can refer to [12].

C. Web Service selection

Web service selection is a process to select the most suitable service from a list of candidate services after service discovery or composition mainly based on nonfunctional properties. Singh and Huhns pointed out three categories of Web service selection strategies: semantic service selection, social service selection, and economic service selection [13]. Semantic service selection finds a match based on the semantic description. Social service selection uses social rating such as reputation, recommendation, referrals, etc. to select a service. Economic service selection uses cost related information to choose a service. The main activities of service selection contain (1) matching nonfunctional service request, (2) evaluation of service offerings, and (3) result aggregation [8]. After discovery a list of candidate services is presented. First, services are matched against the nonfunctional specifications. Secondly, service offering is evaluated. Lastly, the result is aggregated.

Though quite amount of work has been dedicated to Web service selection, a solution that is more practical, less costly, and more universal is still now there. Web service selection is still a research topic for the time being. For a recent development interested reader can refer to [14].

Most selection practice is a composition of semantic, social, and economical strategies. From our point of views social service selection is the key factor. In this paper we will focus on user's preferences in social selection because the user is the final one who makes the decision.

III. System Architechture

This section introduces the system architecture and its modules. The system architecture as shown in Fig. 2 contains four modules: query interface, UDDI repository, discovery and selection module, provider interface, and as follows.

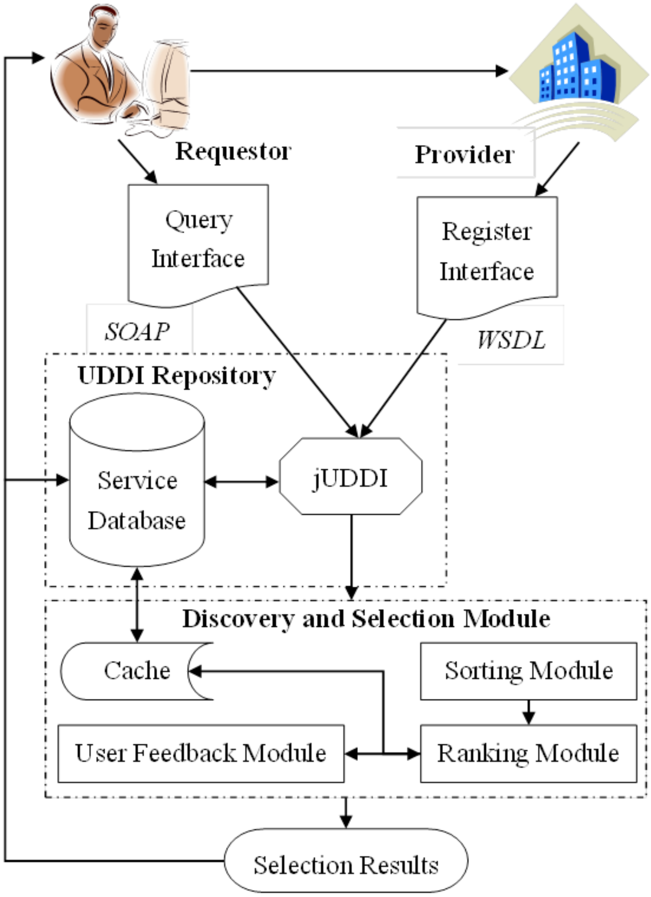


Figure 2. System architecture

- Query Interface: A Web-based interface for a user to query and invoke the service.
- UDDI repository: This module contains a UDDI registry and a database as the service repository. We implement the registry using jUDDI which is an open source Java UDDI implementation.
- Discovery and selection module: This module enhances UDDI with sorting, ranking, and feedback capability. Sorting filters out those services whose attributes are quite different from those of the request. Ranking ranks those services based on similarity to user's request. The feedback module collects user's rating for each service features, analyzes the ratings, and adjusts the weighting of services. The details of the mechanism of this module will be illustrated in the next section. This is the core contribution of this work.
- Provider Interface: A Web-based interface that facilitates service providers to register their services.

IV. Methodology of Service Selection

The core part of the proposed system is the discovery and selection module which is composed of three modules: sorting, ranking, and feedback. As shown in Figure 3 after the user's request from the query interface is collected, the user's selected features and preferences are analyzed. After the sorting and ranking process the results are produced. The user is asked if she or he is satisfied with the result. If the answer is yes, the service is invoked. Otherwise, the ranking process will be rerun based on either adjusting the priority of selection rules or the weighting of selected services. The process will repeat a number of times until the user is satisfied.

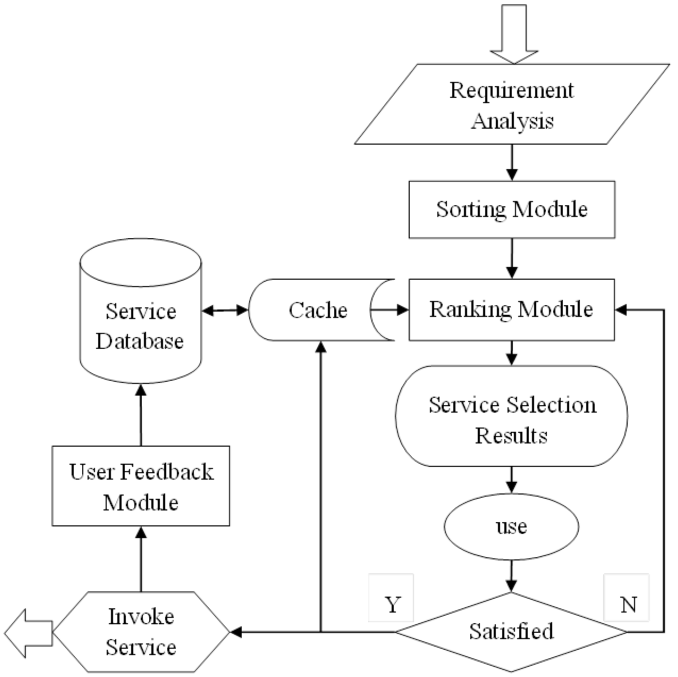


Figure 3. Flowchart of the service discovery and selection.

The major work of the sorting module is to sort all Web services in the service repository by user's preferences and select those more suitable services and filter out those suitable services. In this module user's request (U_{req}), all available services (WS_{all}), and service attributes (WS_{i_attr}) are the input.

When the data (UA_{pre}) of requirement analysis is received, user's preference weighting is acquired. The services will be sorted by the level of how the service meets the user's requirement from high to low. The lower ones are excluded from the next step of refinement, ranking. The algorithm is shown in Figure 4.

```
Sorting Algorithm
$U_{req} \leftarrow \{ur_1, ur_2, ur_3, \dots, ur_n\}$ // User's requests
$WS_{all} \leftarrow \{ws_1, ws_2, ws_3, \dots, ws_m\}$ // All services
$WS_{i\_attr} \leftarrow \{wsc_1, wsc_2, wsc_3, \dots, wsc_l\}$ // Service attribute
$i \leftarrow 0$
for each $ur_i$
        $UA_{pre}[] \leftarrow analyze(ur_i)$
end
$i \leftarrow 0$ , $j \leftarrow 0$
//Send to Sorting Module
for each $ur_i$ in $UA_{pre}$
    for each $wsc_j$ in $WS_{i\_attr}$
        if $ur_i = wsc_j$ then $WS_{all}[] \leftarrow sort(wsc_j)$
        end
    end
end
//Dismiss low level of $WS_{sort}$
$m \leftarrow b$
$WS_{sort} \leftarrow WS_{all}$
//Output of Sorting Module
Return $WS_{sort}$
```

Figure 4. Sorting Algorithm

After sorting the services are further ranked. We proposed a priority-based method which ranks sorted services based on the attribute priority. First a list of services with higher score on the attribute which the user designates as the first priority is selected. Then the similar process will go through the next attribute until the last attribute. In this way the sorted service in the previous step will be ranked. The ranking process is shown in Figure 5 and the algorithm is shown in Figure 6.

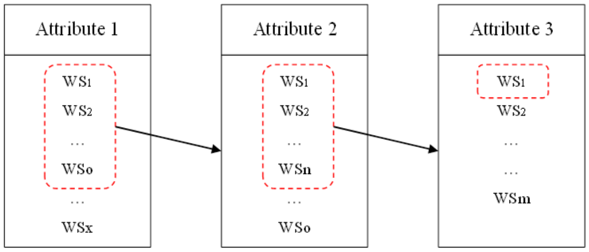


Figure 5. Proity-based ranking process

The last step is the feedback module. After the service is used, an interface is provided to the user. The user can specify the satisfaction rating for each service attribute. These rating is collected by the feedback module and used for adjusting the service selection weighting. In this way the selection weighting can be dynamically adjusted and will produce the results that fit the user's requirements better. The algorithm is shown is Figure 7.

```
Priority-based Ranking Algorithm
WS_sort = {ws_1, ws_2, ws_3, ..., ws_m} // Sorting results
U_req = {ur_1, ur_2, ur_3, ..., ur_n} // User's requests
WS_i_attr = {wsc_1, wsc_2, wsc_3, ..., wsc_l} // Service attributes
i ← 0
for each ur_i
      UB_pre[] ← analyze(ur_i)
end
//Send to Ranking Module
 i ← 0 , j ← 0 , k ← 0
 for each ur_i in UB_pre
     for each ws_j in WS_sort
         for each wsc_k in WS_i_attr
             if wsc_k = ur_i then WS_rank[] ← ws_j
         end
     end
end
//Select high level WS in WS_rank
 WS_rank ← WS_sort
//Output of Ranking Model
Return WS_rank
```

Figure 6. Proity-based ranking algorithm

```
Feedback Algorithm
WS_i_attr = {wsc_1, wsc_2, wsc_3, ..., wsc_l}  // Selected attribute
 //User fills the feedback interface system to get U_ws
Get U_ws = {rws_1, rws_2, rws_3, ..., rws_n}  // User's feedback
 for each wsc_i in WS_i_attr
    // Recalculating service attribute
   average ← compute(U_ws , WS_i_attr)
   WS_attr ← average        // Updating the attribute in database
end
Cache ← ws_i                // Storing in Cache
```

Figure 7. Feedback algorithmExperiments and results

V. Experiment and Results

This section describes the system development, service provision, and experimental results.

A. *System and service development and nterfaces*

This system is built on an Intel Core Quad Q8300 2.5GHz CPU, 4GB DDR RAM with Microsoft Windows Server 2008. JSP is selected as the programming language execute on Glassfish v3 Web server.

To verify the system we develop a travel plan example which includes three categories of services: accommodation, meal, and activity. We register 53 accommodation, 40 meal, and 57 activity services as shown in Table 1, 2, and 3 respectively. There are total 150 services.

Table 1. Categories of accomodation service

Categories	Room capacity	Location	Price	Service Features (Attributes)
Luxury Hotel Motel Log cabin Resort Bed & Breakfast Camping	1 2 4	Kenting Hualian/ Taitung Green Island Surrounding Islands	Based on the listed price	Service Cleanliness Room quality Fire safety Convenience Popularity Total satisfaction Combined average

Table 2. Categories of meal service

Categories	Location	Price	Service Features (Attributes)
Barbecue Quick meal Hot pot Exotic meal Highly exquisite Local specialty Dish set	Breakfast Lunch Dinner	Based on listed price	Service Cleanliness Atmosphere Meal quality Popularity Total satisfaction Combined average

Table 3. Activity classification

Categories	Service Features
Leisure Sightseeing Nautical activity Excitement Backpacking	Popularity Total satisfaction Combined average

The system main functions include activity planning, travel information enquiry, and customer feedback. A user can either queries a single function or a set of functions which can fulfill the user's request. When the system is asked to organize a trip, the activities, accommodation, and meals will be queried in the same time. The results from each query join together to form a travel plan for the user as shown in Figure 8.

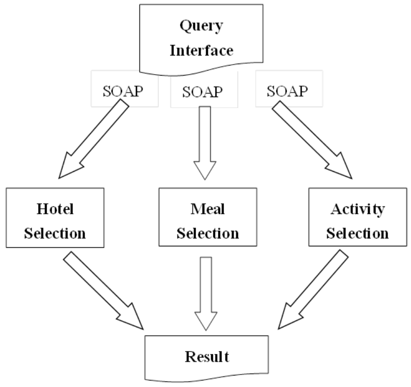
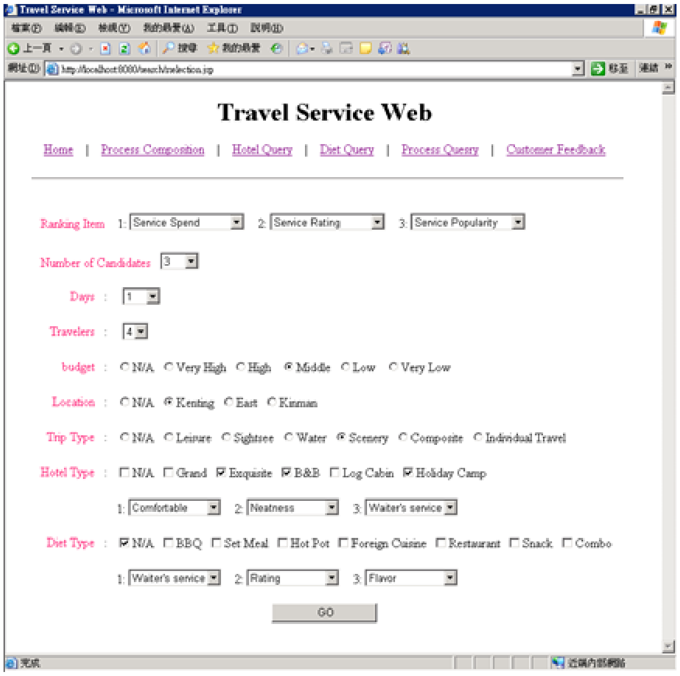


Figure 8. User query process

Figure 9 shows the Web service selection interface. A user can specify their preferences through this interface (Web page). These data are sent to the system for analysis and produce the result. Figure 10 shows the results after going through our proposed mechansim.


Travel Service Web - Microsoft Internet Explorer
檔案(F) 編輯(E) 檢視(V) 我的最愛(A) 工具(T) 說明(H)
上一頁
搜尋
我的最愛
網址(D)
移至
連結
完成
近端內部網路

http://localhost:8080/search/selection.jsp

Travel Service Web

Home | Process Composition | Hotel Query | Diet Query | Process Query | Customer Feedback

Ranking Item 1: Service Spend 2: Service Rating 3: Service Popularity

Number of Candidates 3

Days : 1

Travelers : 4

budget : ○ N/A ○ Very High ○ High ● Middle ○ Low ○ Very Low

Location : ○ N/A ● Kenting ○ East ○ Kinman

Trip Type : ○ N/A ○ Leisure ○ Sightsee ○ Water ● Scenery ○ Composite ○ Individual Travel

Hotel Type : □ N/A □ Grand ☑ Exquisite ☑ B&B □ Log Cabin ☑ Holiday Camp

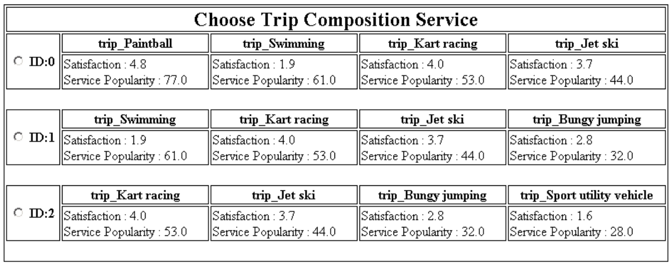
1: Comfortable 2: Neatness 3: Waiter's service

Diet Type : ☑ N/A □ BBQ □ Set Meal □ Hot Pot □ Foreign Cuisine □ Restaurant □ Snack □ Combo

1: Waiter's service 2: Rating 3: Flavor

GO

Figure 9. Web service selection interface



Choose Trip Composition Service

○ ID:0	**trip_Paintball** Satisfaction : 4.8 Service Popularity : 77.0	**trip_Swimming** Satisfaction : 1.9 Service Popularity : 61.0	**trip_Kart racing** Satisfaction : 4.0 Service Popularity : 53.0	**trip_Jet ski** Satisfaction : 3.7 Service Popularity : 44.0
○ ID:1	**trip_Swimming** Satisfaction : 1.9 Service Popularity : 61.0	**trip_Kart racing** Satisfaction : 4.0 Service Popularity : 53.0	**trip_Jet ski** Satisfaction : 3.7 Service Popularity : 44.0	**trip_Bungy jumping** Satisfaction : 2.8 Service Popularity : 32.0
○ ID:2	**trip_Kart racing** Satisfaction : 4.0 Service Popularity : 53.0	**trip_Jet ski** Satisfaction : 3.7 Service Popularity : 44.0	**trip_Bungy jumping** Satisfaction : 2.8 Service Popularity : 32.0	**trip_Sport utility vehicle** Satisfaction : 1.6 Service Popularity : 28.0

Figure 10. Query results

B. *Experiment analysis and discussion*

In this experiment we use MAP (mean average precision) as a measure to evaluate the effectiveness of the system. The basic idea is that a service with a higher rank will have a higher MAP value. The equation is shown as follows:

$\text{MAP} = \frac{\sum_{i=1}^{N} \frac{i}{\text{Rank i}}}{N}$ where N is the total number of selected services and Rank_i is the rank of the ith service.

For example, if a user requests three services and the system finds three services located in the rank first, second, and fifth. The MAP value is 0.866 where $(\frac{1}{1} + \frac{2}{2} + \frac{3}{5})/3 = 0.866$.

We used 150 Web services as shown in Table 1, 2, and 3 to verify our system. We invited 30 students as participants. To minimize the subjective problem, we exclude the 5 values that are too low or too high. We let the participants try the selection with our proposed mechanism and then the original UDDI mechanism which is without sorting or ranking. In such a sequence the effect of the user's familiarity with the system is minimized. The result is shown as in Figure 11 and Figure 12. It shows our proposed mechanism it better than the original mechanism.

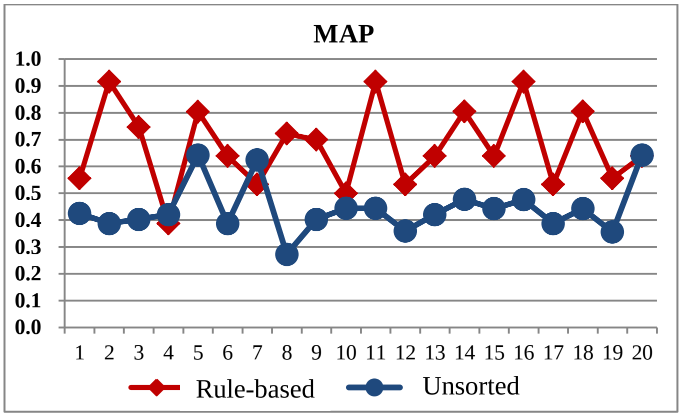


Figure 11. MAP comparison between selection with rule-based ranking and without sorting

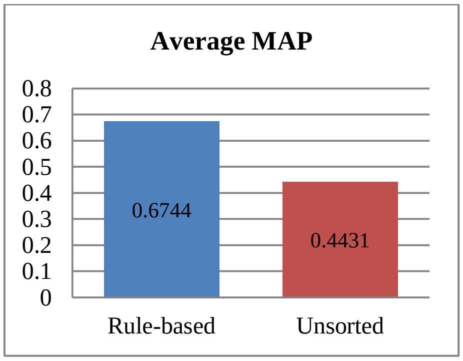


Figure 12. Average MAP comparison between selection with rule-based ranking and without sorting

VI. Conclusion and Future Work

In this paper we present a method for Web service selection that focuses users' preferences. The main objective is to devise a mechanism that takes user's preferences into consideration in every step of the service selection process. In addition, the users' feedback is collected to adjust the selection weighting of the services. To verify our mechanism a system is built and a travel plan example is analyzed. The result shows our mechanism is functional and feasible.

Currently, we are increasing the number of services and enhancing mechanism with fuzzy logics or neural networks. We are also conducting a comparison with other similar system. The other issue is to address the preferences for each user rather than the average of all users' preferences.

Acknowledgment

This work is supported by the National Science Council of Taiwan under the grant number NSC 99-2218-E-216-001.

References

[1] C. Petrie and C. Bussler, “Service Agents and Virtual Enterprises: A Survey”, IEEE Internet Computing, vol. 7, no. 4, pp. 68-78, 2003.

[2] Qi Yu and A. Bouguettaya, “Guest Editorial: Special Section on Query Models and Efficient Selection of Web Services”, IEEE Transactions on Services Computing, vol. 3, no. 1, pp. 161-162, 2010

[3] A. Carenini, D. Cerizza, M. Comerio, Emanuele D. Valle, F. De Paoli, A. Maurino, M. Palmonari, M. Sassi, and A. Turati, “Semantic Web Service Discovery and Selection: a Test Bed Scenario”, Proceedings of the 6th International Workshop on Evaluation of Ontology-based Tools and the Semantic Web Service Challenge (EON-SWSC-2008), Tenerife, Spain, June 1-2, 2008.

[4] Marco Crasso, Alejandro Zunino, Marcelo Campo: A Survey of Approaches to Web Service Discovery in Service-Oriented Architectures. J. Database Manag. Vol. 22 no. 1, pp. 102-132, 2011

[5] Wolf-Tilo Balke and Matt Wagner, “Towards Personalized Selection of Web Services,” WWW (Alternate Paper Tracks) 2003, Budapest, Hungary, 20-24 May 2003.

[6] Hei-Chia Wang, C-S Lee, and Tsung-Hsien Ho, “Combining Subjective and Objective QoS Factors for Personalized Web Service Selection,” Expert Systems with Applications, vol. 32, no. 2, pp. 571-584, 2007

[7] S.P. Ran, “A Model for Web Services Discovery with QoS. ACM SIGecom Exchanges”, vol. 4 no. 1, pp. 1–10, 2003.

[8] H.Q. Yu and S. Reiff-Marganiec, "Non-functional Property based service selection: A survey and classification of approaches", Non Functional Properties and Service Level Agreements in Service Oriented Computing Workshop co-located with The 6th IEEE European Conference on Web Services , November 12, 2008, Ireland, Dublin.

[9] W. Schreiner and S. Dustdar, “A Survey on Web Services Composition,” International Journal of Web and Grid Services, vol. 1, no. 1, 2005, pp. 1–30.

[10] N. Milanovic and M. Malek, "Current solutions for Web service composition," Internet Computing, IEEE , vol.8, no.6, pp. 51- 59, Nov.-Dec. 2004.

[11] J. Rao and X. Su. "A Survey of Automated Web Service Composition Methods". In Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition, SWSWPC 2004, San Diego, California, USA, July 6th, 2004.

[12] D. A. D’Mello, V.S. Ananthanarayana, and S. Salian, “A Review of Dynamic Web Service Composition Techniques,” Communications in Computer and Information Science, Vol. 133, Part 2, pp. 85-97, 2011.

[13] M. P. Singh and M. N. Huhns, “Service-Oriented Computing: Semantics, Processes, Agents,” Wiley, 2005.

[14] Qi Yu and Athman Bouguettaya, “Foundations for Efficient Web Service Selection,” Springer, 2010