



**HAL**  
open science

## Semantic web service adaptation model for a pervasive learning scenario

B. Y. Simon Lau, Cuong Pham Nguyen, C. S. Lee, Serge Garlatti

► **To cite this version:**

B. Y. Simon Lau, Cuong Pham Nguyen, C. S. Lee, Serge Garlatti. Semantic web service adaptation model for a pervasive learning scenario. CITISIA'08: IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, Jul 2008, Cyberjaya, Malaysia. pp.98 - 103, 10.1109/CITISIA.2008.4607342 . hal-02282366

**HAL Id: hal-02282366**

**<https://hal.science/hal-02282366>**

Submitted on 16 Sep 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Semantic Web Service Adaptation Model for a Pervasive Learning Scenario

B-Y-S. Lau, C. Pham-Nguyen, C-S. Lee, S. Garlatti

**Abstract**—With the proliferation of mobile devices, pervasive learning has become a new wave in technology-enhanced learning (TEL). One of the key problems to solve in this area is to adapt learning content and services according to a learner's needs and wants to different learning contexts at the workplace. We propose using semantic web services as a solution to context-adaptive pervasive learning. Another problem lies in web service search, which may return many services that do not match the requirements and context of a learner. This wastes resources and poses more problems to the formulation of just-in-time learning activities. In this paper, we define a service requirement specification to model a service request and a semantic description metadata schema to sufficiently annotate web service functionalities and behavior. On top of that, we propose an adaptation model to match and adapt relevant web services. The technique and algorithm presented in this paper are aimed at improving the efficiency and accuracy in selecting the right web service for a scenario at the workplace.

## I. INTRODUCTION

Mobile or pervasive learning is the new wave of technology-enhanced learning (TEL) with the proliferation of mobile device ownership such as PDAs and mobile phones. Truly “pervasive” learning envisages the combination of formal (at school) and informal (outdoor, at home, at workplace etc.) learning, integration of mobile devices in broader educational scenarios, context-as-construct and seamless learning across different contexts [1], [2]. As such, learning activities conducted through this new learning paradigm requires the delivery of learning content and service tailored to the learner's current situation. Two other compelling reasons for adapting appropriate content and services are limited computing resources of most mobile devices in terms of processing and displays as well as the complex requirement of engaging the learner with more fruitful learning experiences and outcomes [3].

Once a learning activity is determined based on the previous learning history and the learner's current situation,

the adaptation mechanism of learning content, navigation and services [4] for a particular learning context may be applied at two levels of granularities, i.e. 1) adaptation of the learning scenarios and 2) adaptation of services as resources to fulfill the requirements of a learning task.

With regards to the first type of adaptation, substantial work has been done such as adaptations through hierarchical tasks/the method model [5]. Generally, adaptation is aimed at achieving an activity according to the current situation. In other words, the learning system has to select dynamically the relevant way to achieve the different tasks included in a scenario. The system thus needs to carry out just-in-time searching and matching of available web services that may provide such desired functionalities.

Our paper addresses the second type of adaptation, i.e. service-adaptation. We propose a solution to manage context-adaptive semantic web services. The main contribution of this paper is a service matching and adaptation model that serves as a solid foundation for service discovery, matching and adaptation. The model is built with the support of the service requirement specification identified by a given task and the service descriptor indexing web services. All of the above will be illustrated with a context-aware learning scenario for the workplace.

The organization of this paper is as follows: First, we illustrate our context modeling technique on a simulated example of context-aware learning scenario for the workplace. Second, we present a service requirement specification and a service description metadata model that facilitate service retrieval and adaptation. Third, we describe a detailed adaptation process to rank services retrieved for a learning task in a scenario. Finally, we conclude with a summary, lessons learned and item for future work.

## II. RELATED WORK

At present, there have been numerous efforts to annotate web services with semantic information, such as WSMO [6], WSDL-S [7], METEOR-S [8] and OWL-S [9]. However, much of the efforts are on the definition and implementation of semantic web services in e-commerce and business process integration domains. Even though in recent times, there are some efforts to explore the implementation techniques for context-aware pervasive learning [5], [10], [11], efforts to adapt semantically annotated web services [12]-[15]

Manuscript received April 30, 2008.

B. Y. Simon Lau, is with the Faculty of Engineering, Multimedia University, Cyberjaya, 63100 Selangor, Malaysia (e-mail: [simon@mmu.edu.my](mailto:simon@mmu.edu.my)).

Cuong Pham-Nguyen is with the Computer Science Department, Telecom-Bretagne, CS 83818, 29238 Brest Cedex3, France (e-mail: [cuong.nguyen@telecom-bretagne.eu](mailto:cuong.nguyen@telecom-bretagne.eu)).

C. S. Lee, is with the Faculty of Information Technology, Multimedia University, Cyberjaya, 63100 Selangor, Malaysia (e-mail: [cslee@mmu.edu.my](mailto:cslee@mmu.edu.my)).

Serge Garlatti, is with the Computer Science Department, Telecom-Bretagne, CS 83818, 29238 Brest Cedex3, France (e-mail: [serge.garlatti@telecom-bretagne.eu](mailto:serge.garlatti@telecom-bretagne.eu)).

discovery and ranking mechanism for learning scenarios are still mainly not implemented on a large scale at this stage.

### III. CONTEXT-AWARE LEARNING SCENARIO FOR WORKPLACE

We will take a portion of an e-Retail learning scenario at the workplace shown in Fig. 1 below as the basis of our service adaptation model. This learning scenario is adapted from the p-LearNet project done in [5]. A learning situation at the workplace is composed of the physical environment, learning and commercial setting of the user's current work situation. Pervasive learning scenarios at the workplace are modeled by a hierarchical structure of tasks/methods representing learning and working activities. For a complete presentation and execution of the scenario, another category is needed: mixed activities. Thus, working and learning activities are represented by working, learning and mixed tasks. In Fig. 1, the context descriptor and the control structure of some methods are represented for different categories of tasks. The context descriptor of certain methods of a task describes the current situation of the learner. Adaptation modeling based on context is to be articulated in the sections that follow.

As mentioned in Section I, our focus is on service adaptation, an adaptation mechanism dedicated to refine the service retrieval process to realize an atomic task: how to discover available services and adapt them according to the current situation for realizing a learning task. In [5], a learning situation can be sufficiently described by a set of context features belonging to various context dimensions such as the scenario (a hierarchal task model having a task/method paradigm), the user (a user can be a learner, a teacher, a salesman, a customer, etc.) with sub-dimensions: the role, previous knowledge, know-how, preferences, loyalty card, purchase intentions, intention of use), the retail shop, the

TABLE I

DESCRIPTION OF PEDAGOGICAL AND PROFESSIONAL CONTEXT VIEW

Context views	Dimensions or features
Pedagogical	{user(role, knowledge, how-know); learning collaborator; scenario (learning activity); pedagogical tool; learning objects; learning location (coordinate, place); time; device (device type: PTA, RFID tag reader, large LCD screen); network (network type, bandwidth); system resource; physical environment (temperature, noise level)}
Professional	{user(role, preference, loyalty, purchase intentions, intentions for use); organisation unit; professional collaborator; scenario (working activity); resources; working location (coordinate, place); time; device (device type: PTA, RFID tag reader, large LCD screen); network (network type, bandwidth); system resource; physical environment (temperature, noise level)}

device, the location, the time, the pedagogical tools, the network, the physical environment and the resource (learning object, services, media resource, system resource, etc.). Specific metadata are associated to situation properties and context dimensions to enable adaptation. Context features are either permanent or transitory. Permanent context features are *a priori* knowledge about the scenario and the learner and consistent across different situations. Transitory context features are dynamic, and their values can only be gathered during runtime. It is necessary to analyse each context feature which contributes to service adaptation.

We extend the work in [5] by creating *context views* to define different viewpoints about the context usage. It is done by re-grouping context features to form a subset of all the context features of the context space. The context model is not prescriptive in nature as we do not want to restrict it to certain scenarios only. The context viewpoint can be easily tailored to

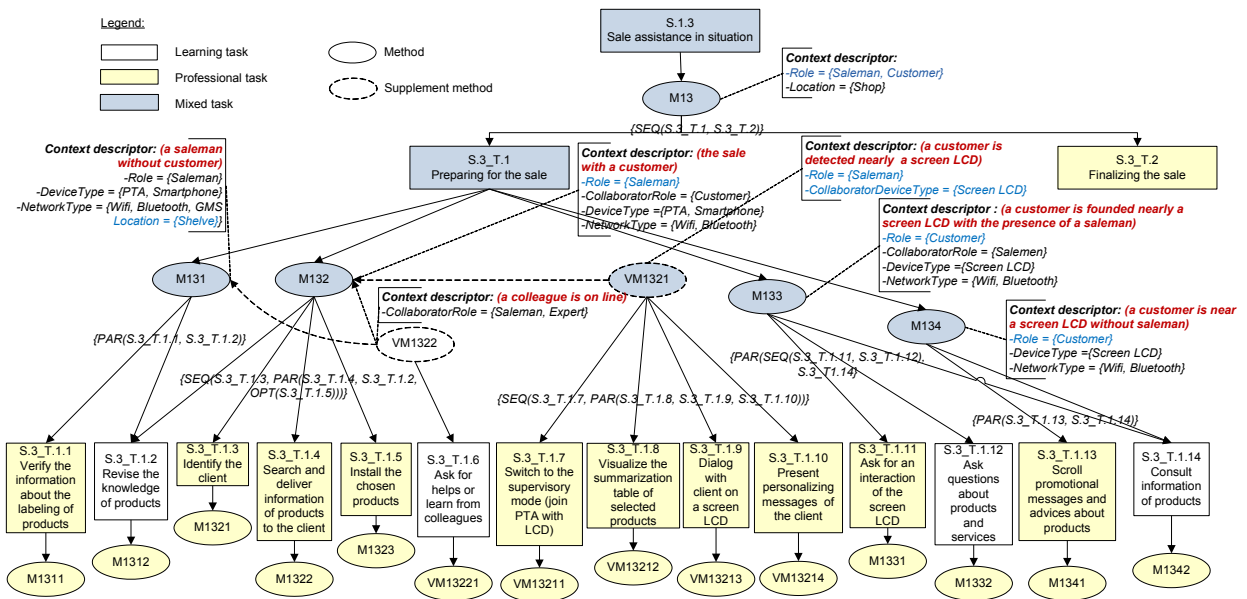


Fig. 1 A small portion of the e-Retail scenario

suit different domain types. At present, we have identified the pedagogical and professional context views to cater to the learning scenario for the workplace as illustrated in Fig. 1 below. They are shown in Table I in the previous page.

#### IV. SERVICE REQUIREMENTS SPECIFICATION

In a learning scenario, an activity may be realized by either one or more web services. Web services can be internal or external. The problem to solve is to search for a service or a set of services that may be composed in a certain order to fulfill the requirements of the current task. In the context of this paper, the task is an atomic task on a task/method hierarchy.

In systems without context adaptation, web services are statically bound to the achievement descriptor of a method [5] during design time. In this regard, the problem is just to bind, invoke and execute these web services in a known order. Take for instance Task “S.3\_1.4 – Search and deliver information of products to the client” in Fig. 1 in the previous page (web service A can be bound to this task at design time for searching product information and content objects (images, sounds, documents, etc.). At runtime, logics are built into web service A to search for relevant content objects annotated with description metadata from repositories or from the database and deliver them to the client which made the request.

Even though the ultimate long term vision of semantic web services is fully automatic composition of web services for a user-defined requirement, we argue that it is not practicable yet at this moment in time due to problems such as non-determinism [16]-[19], partial observability [18], [20] and semantic heterogeneity [21].

The solution that we propose in this paper is *semi dynamic service adaptation* (discovery and matching). In our proposition, the relevant services can be a single matched service or a set of matched services. For a set of matched services, the system searches for all the relevant services provided by service providers according to the current context descriptor and the service requirement specification (to be presented in this section) in the achievement descriptor. This is done based on an adaptation process (to be presented later) applied for services. For example, taking Task “S.3\_1.4” above again, our solution consists of four phases.

i) *Phase 1*: a search of web services, which provide information of a selected product or a product type required by the Service Requirement Specification of the method “M1322”, on a service repository is first carried out according to the current situation.

ii) *Phase 2*: All relevant web services of all the suppliers are organized, invoked and executed for searching of product information and resources.

iii) *Phase 3*: The found resources are classified according to the current situation. They are filtered and ranked according to their degree of relevance.

iv) *Phase 4*: Learning/working activities are delivered with the learning content to the targeted peripheral devices.

Characteristics and functionalities of a Service required by a relevant task or activity have to be specified semantically as

TABLE II  
AN EXAMPLE ILLUSTRATING THE SERVICE REQUIREMENT SPECIFICATION FOR THE METHOD “M1322”

Service Request	SRS_M1322	
Functional Requirement	( input	(ProductType ?pt) (ProductInfoType ?pi) (DeliveredResourceType ?rt) (ProductResource ?pr)
	output	(ProductResource ?pr)
	pre-condition	(isNotNull ?pt)
	post-condition	(isNotNull ?rt)
	)	
Non-functional Requirement	(	(DesiredQualityFeatures ?qf)
	)	
Content Requirement		(Product, ProductType, ProductPrice, ProductSupplier)

a signature to facilitate accurate and efficient discovery and matching of the right services. The primary goal of a Service Requirement Specification is to describe how a service is to be “desired”. It is a request issued by the system wishing to interact with a Service provider in order that a task is performed on behalf of the learner in the current context. By our definition, a service satisfies a service requirement by providing a set of desired output parameters for a desired goal (or “effect”) with a set of input parameters and situational context features.

Our proposed service requirement specification is summarized in as follows:

1) *Functional requirement*: describes the capabilities of web services desired by a user. It is characterized by input parameters (i.e. a set of triples of *name*, *type* and *value*), target output parameters (i.e. a set of triples of *name*, *type* and *value*), pre-conditions (i.e. a set of declarative predicates using the context features taken from context view) and expected post-conditions (i.e. a set of declarative predicates with another set of features describing the learner and the scenario after execution);

2) *Non-functional requirement*: includes the identity to annotate a service (e.g. name, owner, type etc.) as well as performance related parameters, such as Quality of Service (QoS), security, availability etc;

3) *Content requirement*: this specifies a list of domain concepts or a query identifying the content objects (e.g. Price, ProductModel, learning objects etc.).

For easy transfer and processing between different systems, a service requirement specification is manifested in OWL. The OWL representations are discussed in another paper.

Table II above shows an example of a service requirement specification “SRS\_M1322” for the method “M1322” which realizes the task “S.3\_T.1.4 – Search and delivery information of products to clients”. This specification consists of three main parts. In the functional requirement part, the input is composed of three parameters (ProductType ?pt, ProductInfoType ?pi and DeliveredResourceType ?rt) describing desired capabilities of the service for retrieving the desired resource (e.g. document, media, voice file, pages, database, etc.). The pre-condition is a predicate that verifies,

for example a mandatory presence of a *ProductType*. The output describes a list of relevant resources to be delivered to the client. The non-functional requirements portion represents QoS related context features that can be used for ranking purpose (see the section IV, subsection: ranking). Finally, the content requirement presents a list of domain concepts that covers a sub-domain semantic (ontology of products) for query refinement. These concepts are used to “compare” with the DomainContent category of a service (see Table III above) through the domain ontology.

## V. SEMANTIC SERVICE DESCRIPTION

Towards semantic discovery and matching of the right services, web services have to be semantically annotated with a semantic service descriptor. The fundamental consideration in describing a web service to support accurate and efficient service discovery and matching is to fulfill a three-part ontology [22]: *function*, *behavior* and *interface*. The *interface* dictates how the service can be invoked for a task/method by a calling learning system. The syntax aspect (message format and protocol details) of the Service has been specified in UDDI and WSDL [23]-[25].

TABLE III  
SERVICE FEATURE SET

Categories	Feature Set
General	{name, description, language, owner, type (name, taxonomy, value), entityType}
Meta-metadata	{metadataCreator, metadataValidator, creationDate, validationDate, language, format}
Life-cycle	{creator, dateCreated, version, status, contributor, publisher, dateUpdated, extentOfValidity}
Right	{IP, accessRight, signature, provenance, dateCreated, dateUpdated}
Technical	{URI, resource, resourceURI, resourceFormat, replacedBy, realisation, modeOfInteraction}
ServiceRequirement	{input(name, type, value, ontologyURI), output(name, type, value, ontologyURI), expectedEffect}
DomainContent	{listDomainConcepts}
Context	{roleModels, location(coordinate, spatialLocation, locationRelativity), physical(deliveryChannel, deliverySystem, deviceModel, tool), informaticResource(hardware, software), temporal(temporalCoverage, frequencyRequirement)}
Quality	{qualityRating, trustRating, qualityGuarantee, networkedQoS, accuracy, performance, reliability, robustness, scalability, security, availability, stability}
Financial	{cost, currency, chargingStyle, settlementModel, settlementContract, paymentObligation, paymentInstrument}

In this paper, we are interested in specifying a Service’s functionality and behavior to facilitate easy matching with the service requirement. It is contained in a signature-like Service descriptor feature set that can be specified in the

*ModelReference* extended elements/attributes of WSDL. Table III above enlists features which are generally applicable to most web services.

The features in Table III above can be extended to include features which describe more specific characteristics of web services such as pedagogy. The features on a feature list can be either *mandatory* or *optional*. For example, features such as name, description, language etc. in the *General* category are compulsory for all web services while features in the *meta-metadata*, *life-cycle*, *context* etc. can be optional depending on the scenario. They serve to index a service for either *filtering* or *ranking* purposes. For instance, the *location* feature when tested using an adaptation rule using the *location* context feature of the service requirement specification can be used to determine whether this service is to be relevant or not. A *Quality* feature such as *qualityRating* is used to rate and rank the service in relation to other service provisions in the web service search result list.

## VI. SERVICE MATCHING AND ADAPTATION

Service discovery, matching and adaptation involve the problem of matching the service requirement specification to the service descriptor. At present, we are investigating and designing a schema mapping mechanism to map features on the service description specification to the service descriptor for filtering and ranking purposes.

In this section, we elaborate on the adaptation principle carried out by the Service Matching and Adaptation Framework as shown in Fig. 2 below. The inputs to the matching subsystem are the Service requirement specification identified by a given task and the service descriptor indexing web services. The global process consists of two main phases executed sequentially: Service retrieval and Service adaptation. The former is aimed at searching relevant services in the repository based on the Service description to fulfill the Service request for a task. This is done by querying the Service descriptor metadata repository (matching particularly the ServiceRequirement and DomainContent of Services with functional, non-functional and content requirements of a request). The relevant Services serve as the input for the adaptation phase. The latter is aimed at refining the relevant Services according to the current situation and user’s interest. The result of the adaptation will be utilized by the Service Invocation, Orchestration, Choreography and Execution Subsystem which are parts of a global Pervasive Learning System.

The adaptation process consists of three stages: evaluation/classification, filtering and ranking (see Fig. 2). It is specified by a function  $f^a$  :

$$f^a = \{f^c, f^f, f^r\} \quad (1)$$

i) **Classification:** input services are classified according to the current situation in several equivalence classes: two classes {“Good”, “Bad”} for each transitory feature and up to five equivalence classes for all permanent situation features, together. Services belong to an equivalent class if they satisfy

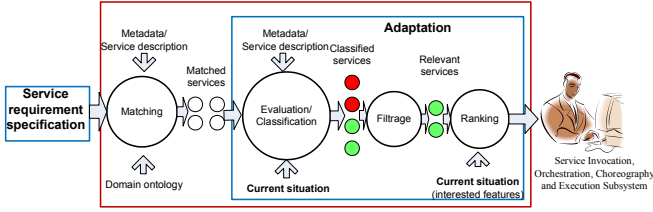


Fig. 2 Service Matching and Adaptation Framework

its membership rules. In a formal way, the classification function  $f^c$  is defined as follows:

$$f^c = f_1^c * f_2^c * \dots * f_n^c \quad (2)$$

where  $n$  is the number of context features for evaluation.

The function  $f_i^c$  is the classification function for each feature  $i$  and is specified by:

$$f_i^c : Sr \rightarrow Sr \times EC_i \quad (3)$$

where  $Sr$  denotes a set of services,  $EC_i$  is a set of equivalent classes of the feature  $i$ . At present, all transitory context features have two equivalent classes {"Good", "Bad"}.

Let  $(s, ec^i) = f_i^c(s)$ . This means that the classification function  $f_i^c$  classifies the Services into an equivalent class  $ec^i \in EC_i$ . The  $*$  operator is specified as follows:

$$(f_i^c * f_j^c)(s) = f_i^c(s) * f_j^c(s) = \langle s, \{ec^i\} \cup \{ec^j\} \rangle = \langle s, \{ec^i, ec^j\} \rangle$$

Let  $s^c = f^c(s)$ , by applying the functions (2) and (3), we achieve the classified service  $s: s^c = \langle s, \{ec^1, ec^2, \dots, ec^n\} \rangle$

For example,  $Sr = \{srv_1, srv_2, srv_3\}$  are matched for a service request of a task. In this case, there are four context features that are taken into account for the classification: *deviceModel*, *Location*, *deliveryChannel* and *performance*. The equivalent classes correspondent are  $EC_{dev} = EC_{loc} = EC_{del} = EC_{per} = \{"Bad", "Good"\}$ . The result by applying the classification function for  $Sr$  is described as follows:

$$f(srv_1) = \langle srv_1, \{"Good", "Good", "Bad", "Bad"\} \rangle$$

$$f(srv_2) = \langle srv_2, \{"Bad", "Good", "Good", "Good"\} \rangle$$

$$f(srv_3) = \langle srv_3, \{"Good", "Good", "Good", null\} \rangle$$

Note in the example above, service  $srv_3$  does not have information for the evaluation of the *performance* feature.

ii) **Filtering**: all services belonging to "Bad" classes according to a mandatory feature are filtered out (mandatory features are subset of transitory features). In other words, these services are considered irrelevant and discarded. For example, with the *deviceModel* feature, the class "Good" is considered as relevant while the class "Bad" is not. So, the system will eliminate all services that belong to the class "Bad". The filtering function  $f^f$  is formally specified as:

$$\forall s^c = \langle s, \{ec^1, ec^2, \dots, ec^n\} \rangle \in Sr^c, \quad (4)$$

$$(\exists ec \in \{ec^1, ec^2, \dots, ec^n\} \wedge ec \in EC^{ir}) \Rightarrow Filter(s^c)$$

where  $Sr^c$  denotes a set of classified services,  $EC^{ir}$  is the set of irrelevant equivalent classes ("Bad") for all transitory

features. The function  $Filter(s^c)$  eliminates the service  $s^c$  from  $Sr^c$ .

For instance, taking the set of three services in the example above, the mandatory features for filtering are *deviceModel* and *location*, thus the irrelevant equivalent classes  $EC^{ir}$  is {"Bad", "Bad"}. By applying the filtering algorithm (4) for  $Sr^c = \{srv_1, srv_2, srv_3\}$ , the service  $srv_2$  will be eliminate because it belongs to the class "Bad" of the *deviceModel* feature. Thus, the services  $Sr^f$  remain after filtering step are  $\{srv_1, srv_3\}$ .

iii) **Ranking**: some ranking algorithms have been proposed for web services [12]-[15]. In WSR [13], a ranking process will reduce the burden of finding the most relevant web services from the returned result set based on domain ontology. The ranking algorithm considers the semantics of each Service, and only recommends the most crucial ones to a request. In a pervasive environment, the ranking process is mainly to refine matched services according to the current context. Some context features from the Context, Quality, Financial, etc. categories are taken into account for the service ranking process. In this regard, the user can decide intentionally what context features are more important than others from a feature list proposed by the system. For example, users can choose the *qualityRating* of services as with high priority while the others are more interested in *performance*. Moreover, ranking requirements also depend on the type of services. With services for voice communications, the performance is very important while with services for financial transactions, the security must be highly assured. Therefore, the user can build an "interested" feature priority list by annotating each feature with an "interested" level (from very low to very high). This information will be memorized in the user's profile for later reuses and be part of the current situation information for this user. Based on the list of "interested" features, the system can rank all matched services. The ranking process  $f^r$  is presented formally by a quadruple as follows:

$$f^r = (I^u, W^u, Sr^f, \leq_{relevant}) \quad (5)$$

where  $I^u = \{i_1, i_2, \dots, i_l\}$  denotes the interested feature set of the current user  $U$ .  $W^u = \{w_1, w_2, \dots, w_l\}$  is the correspondent set of mapped weights, where  $w_i \in \{0, 1, 2, 3, 4\}$  relating to *{very low, low, average, high, very high}*.

$Sr^f = \{\langle sr_1, c_1 \rangle, \langle sr_2, c_2 \rangle, \dots, \langle sr_k, c_k \rangle\}$  is the set of relevant services that serves as the input. Each  $\langle sr_i, c_i \rangle$  is obtained by a projection of the classified service  $s_i \in Sr^f$  on the interested features set  $I^u$ .  $c_i = \{ec_{i1}, ec_{i2}, \dots, ec_{il}\}$ ,  $i = 1, k$  is a  $l$ -array representing a classification of a service  $sr_i$  with all interested features.

The ranking algorithm ranks relevant services according to their degree of importance as determined by the weightage assigned to each equivalent class. The degree of importance is calculated by a function described below:

$$f_{imp}(\langle sr_i, c_i \rangle) = \sum_{j=1}^{|I^u|} f_{weight}(ec_{ij}) \quad (6)$$

$f_{weight}(ec_j)$  returns a correspondent interested weight for the equivalent class  $ec_j$ . It is specified as follows:

$$f_{weight}(ec_{ij}) = \begin{cases} 0 & \text{if } ec_{ij} = \text{"Bad"} \text{ or } ec_{ij} = \text{null} \\ w_j & \text{if } ec_{ij} = \text{"Good"} \end{cases} \quad (7)$$

$\leq_{relevant}$  is the total relevant order defined on  $Sr^f$  for ranking purpose,  $srv_i \leq_{relevant} srv_j$  implies the service  $srv_j$  is more relevant than the service  $srv_i$  if  $f_{imp}(srv_j) \geq f_{imp}(srv_i)$ .

For example, take the aforementioned example again. The interested features  $I^u$  for ranking are  $\{location, deliveryChanel, performance\}$  and the corresponding mapped weights  $W^u$  are  $\{2, 3, 4\}$  relating to  $\{average, high, very high\}$ . After applying a projection of  $Sr^f$  on  $I^u$ . We obtain  $Sr^f = \{<srv_1, \{\text{"Good"}, \text{"Bad"}, \text{"Bad"}\}\rangle, \langle srv_3, \{\text{"Good"}, \text{"Good"}, \text{null}\}\rangle\}$ . The important point of each relevant service calculated by (6) and (7) is:

$$f_{imp}(srv_1) = 2 + 0 + 0 = 2$$

$$f_{imp}(srv_3) = 2 + 3 + 0 = 5$$

Because  $f_{imp}(srv_3) \geq f_{imp}(srv_1)$ , consequently, the ranking order is specified as:  $srv_1 \leq_{relevant} srv_3$ . In other words, the service  $srv_3$  is the most relevant according to the current context.

## VII. CONCLUSION

In this paper, we have described a context-aware learning scenario for the workplace based on a hierarchical task/method model, a service requirement specification, a service description metadata model and a service adaptation model. They serve as a solution to manage dynamic content and learning activity delivery in a dynamic pervasive learning system using context adaptive semantic web services. The proof of concept system is currently very much a work in progress with specific learning scenarios rigorously being defined and refined. The results will be presented in our future publication.

## REFERENCES

- [1] M. Sharples, "Big Issues in Mobile Learning: Report of a workshop by the Kaleidoscope Network of Excellence Mobile Learning Initiative," LSRI, University of Nottingham 2006.
- [2] N. Balacheff, "10 issues to think about the future of research on TEL," Les Cahiers Leibniz, Kaleidoscope Research Report, 2006.
- [3] R. Beale and P. Lonsdale, "Mobile Context Aware Systems: the intelligence to support tasks and effectively utilise resources," in *Mobile Human-Computer Interaction (MobileHCI 2004)*, 2004, pp. 240-251.
- [4] B. Bomsdorf, "Adaptation of Learning Spaces: Supporting Ubiquitous Learning in Higher Distance Education," in *Mobile Computing and Ambient Intelligence: The Challenge of Multimedia, Dagstuhl Seminar Proceedings*, 2005.
- [5] C. Pham-Nguyen and S. Garlatti, "Pervasive Learning System based on a Context-Aware Hierarchical Task Model", *IEEE RIVF'08*, to be published.
- [6] (Online Source) J. de Bruijn et al. (2005). *Web Service Modeling Ontology (WSMO)* [W3C Member Submission]. Available: <http://www.w3.org/Submission/WSMO/>
- [7] (Online Source) R. Akkiraju et al. (2005). *Web Service Semantics - WSDL-S*. Available: <http://www.w3.org/Submission/WSDL-S/>

- [8] A. Patil, S. Oundhakar, A. Sheth and K. Verma, "METEOR-S Web service Annotation Framework," in *Proceeding of the World Wide Web Conference*, July 2004, pp. 553-562.
- [9] (Online Source) D. Martin et al. (2004). *OWL-S: Semantic Markup for Web Services*. Available: <http://www.w3.org/Submission/OWL-S/>
- [10] G. Zhao and Z-K. Yang, "Learning Resource Adaptation and Delivery Framework for Mobile Learning," in *Proceedings 35th IEEE Annual Conference on Frontiers in Education, (FIE '05)*, 2005, pp. F1H-18 - F1H-24.
- [11] Y-K Wang. "Context Awareness and Adaptation in Mobile Learning," in *2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'04)*, 2004, pp. 154-158.
- [12] A-M, Eyhab and M. H. Qusay, "Relevancy Ranking of Web Services", in *ISIC. IEEE International Conference on Systems, Man and Cybernetics*, October, 2007, pp. 783-788.
- [13] H. Lu, "Semantic Web Services Discovery and Ranking," in *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05)*, France, pp. 157 - 160.
- [14] N. Ahmadi and W. Binder, "Flexible Matching and Ranking of Web Service Advertisements," In *MW4SOC Workshop*, Newport Beach, CA, USA, 2007.
- [15] H.-E. Lu, "Ranking Web services based on ontology semantics," in *Proceedings of 2005 International Conference on Machine Learning and Cybernetics*, 4(4), 2005, pp. 2161-2165.
- [16] (Online Source) S. S. Sahoo (2004). *Web Services Composition (An AI-based Semantic Approach)* [Power Point slides]. Available: <http://lsdis.cs.uga.edu/~satya/Presentation/Web%20Services%20Composition.ppt>
- [17] M. Vukovic and P. Robinson, "Adaptive, Planning Based, Web Service Composition For Context Awareness," In *International Conference on Pervasive Computing*, Vienna, April 2004.
- [18] M. Pistore, F. Barbon, P. Bertoli, D. Shaparau and P. Traverso, "Planning and Monitoring Web Service Composition", in *Proceedings of the 14th International Conference on Automated Planning and Scheduling (ICAPS 2004)*, 2004, pp.106-115.
- [19] B. Srivastava and J. Koehler, "Web Service Composition - Current Solutions and Open Problems," in *ICAPS 2003 Workshop on Planning for Web Services*, 2004, pp. 28 - 35.
- [20] Y. Charif and N. Sabouret, "An Overview of Semantic Web Services Composition Approaches," in *Electronic Notes in Theoretical Computer Science Volume 146 Issue 1, Proceedings of the First International Workshop on Context for Web Services (CWS 2005)*, 2005, pp. 33-41.
- [21] M. Nagarajan, K. Verma, A. P. Sheth, J. Miller and J. Lathem, "Semantic Interoperability of Web Services - Challenges and Experiences," in *IEEE International Conference on Web Services (ICWS'06)*, 2006, pp. 373-382.
- [22] N. Milanovic and M. Malek, "Current Solutions for Web Service Composition," in *IEEE Internet Computing* vol. 8, no. 6, 2004, pp. 51-59.
- [23] (Online Source) K. Januszewski and E. Mooney (2002). *UDDI Features List (3)*. Available : [http://uddi.org/pubs/uddi\\_v3\\_features.htm](http://uddi.org/pubs/uddi_v3_features.htm)
- [24] (Online Source) J. Colgrave, K. Januszewski, F. Curbera, D. Ehnebuske, and D. Rogers (2002). *Using WSDL in a UDDI Registry, (1.08)*. Available: <http://www.oasis-open.org/committees/uddi-spec/doc/bp/uddi-spec-tc-bp-using-wsdl-v108-20021110.htm>
- [25] (Online Source) E. Christensen, F. Curbera, G. Meredith and S. Weerawarana (2002). *Web Services Description Language (WSDL) 1.1*. Available: <http://www.w3.org/TR/wsdl>