Pervasive Learning System Based on a Scenario Model Integrating Web Service Retrieval and Orchestration

C. Pham-Nguyen[1], B. Y. S. Lau[2], B. Barbry[3], T. Vantroys[3] and S. Garlatti[1]

[1] TELECOM- Bretagne, Brest, France, [2] Multimedia University, Cyberjaya, Malaysia
[3] University of Sciences and Technologies of Lille, Villeneuve d'Ascq, France

Abstract—**We are interested in learning and working scenarios integrating web service retrieval and orchestration in pervasive TEL systems in a learning situation at workplace. This paper proposes a context-aware model of corporate learning and working scenarios in e-retail environment such as shops and hypermarkets. This scenario model enables us to select how to achieve activities according to the current situation. We outline the semantic description of web services to enable the selection, composition and execution of web services to achieve objectives specified by learning and working activities. We propose a context-aware and adaptive model for pervasive learning systems. This model enables the selection of the relevant methods or services to realize activities according to the current situation. Moreover, we also build and develop an Intelligent Selling Space (ISS) architecture that serves as an infrastructure for service management and execution in e-retail environment.**

Index Terms—**Adaptation, Context-aware, Hierarchical task model, Learning and working situation, Pervasive learning, Service description, Service requirement, Task/method paradigm.**

I. INTRODUCTION

Nowadays, technology-enhanced learning (TEL) systems must have the capability to reuse learning resources from large repositories, to take into account the context and to allow dynamic adaptation to different learners based on substantial advances in pedagogical theories and knowledge models [1]. It is particularly true in mobile learning. Thus, several expressions are used: pervasive, ubiquitous, mobile, ambient and nomadic learning systems. It is necessary to clarify their meaning. We consider that these expressions are synonymous. They are used for highlighting a specific property in learning systems [2-5]. Generally, such types of learning systems have all the properties of mobile, pervasive and ubiquitous computing. Many definitions of pervasive learning are given in literature [2, 4, 6, 7]. We can cite the following one: *"Pervasive learning environment is a context (or state) for mediating learning in a physical environment enriched with additional site-specific and situation dependent elements – be it plain data, graphics, information -, knowledge -, and learning objects, or, ultimately, audio-visually enhanced virtual layers"* [2]. Virtual Learning Environments can be built based on a Service Oriented Architecture approach. It facilitates the deployment of adapted learning environment based on the aggregation and orchestration of the services needed by an organization. This approach can be effective for pervasive learning environments if one considers a continuous adaptation based on the available services and other contextual information.

We are interested in the following issues: combination of formal (at school) and informal (outdoor, at home, at the workplace etc.) learning, integration of mobile devices in broader educational scenarios, context-as-construct and seamless learning across different contexts [1, 8]. Moreover, the p-LearNet project [9] has to integrate context-aware corporate learning and working activities in a particular framework: the e-retail framework (retail activities through shops and hypermarkets). Scenarios are used to describe the learning, working and tutoring activities to acquire some domain knowledge and know-how or to solve a particular problem or to support working activities. Scenario analysis reveals that learning and working situations can be modeled by an explicit task model because working and learning activities are well structured and stable. In pervasive learning systems, activities, represented by tasks, can be achieved in a different way according to the current situation. Methods associated to tasks, enable us to provide different ways for carrying out tasks. Activities need to have access to resources or web services to be performed. Thus, a context-aware and adaptive mechanism is necessary to select relevant methods associated to a task, resources and web services.

The main contribution of this paper is an adaptive and context-aware model of a scenario integrating web service retrieval and orchestration based on an interdisciplinary approach (education, computer science, social sciences, and business) for a pervasive learning system supporting working and learning activities. The scenario model is based on a hierarchical task model having the task/method paradigm. An activity, represented by a task, may have several associated methods. A method represents a way to perform a task in a particular situation. The context-aware and adaptive mechanism can be viewed as the selection of the relevant content for a given task according to the current working and/or learning situation. This mechanism is based on the matching between content description and the current situation for filtering, annotation and ranking purposes. Content (methods, resources and services) and situation need to have corresponding features for adaptation purpose. Methods are described by contextual

features while resources and services are described by metadata. Situations are described according to a context model. For managing web services, we also define the service requirement specification for web service retrieval. Besides, a generic ISS infrastructure based on Open Services Gateway initiatives (OSGi) and Universal Plug and Play (UPnP) is also proposed for web service orchestration and execution in workplace environment.

First of all, the p-LearNet project and e-retail system goals are presented. Secondly, a context management model (organization, features) is detailed and linked to a situation. Thirdly, we present our scenario model and its relationships with the context management model. Fourthly, the context management, i.e. the detection and creation of new situations are explained. Fifthly, the web service requirements and specifications are presented and also the semantic metadata schema describing the web services. Sixthly, we present our service execution architecture that serves as a platform of service management in e-retail systems. Seventhly, the adaptation process which links the context model, the scenario model, the metadata schema and the context management is presented. Finally, the conclusion highlights the main results of this study and some perspectives on the results.

II. P-LEARNET PROJECT AND E-RETAIL SYSTEMS

At first look it seems that the traditional retail industries are not heavily impacted by Information & Communications Technologies (ICTs) in their relationships with the customers. Obviously ICTs were found in back-offices of the retail places, i.e. in the Supply Chain Management (SCM) systems, or in the Point of Sales [10] for the payment phase for more than two decades. However, observations made at the different retail places show that there is more and more presence of ICTs in direct interaction with the customer, with a large diffusion of electronic displays of different formats: from small monochrome LCD used as price tags to large ones used for dynamic advertising. But these ICT devices are not truly interactive. We also notice other more interactive systems allowing customers to check product prices, or to listen to the content of music CDs, by scanning barcodes, or even selecting the right ink cartridges for ink-jet printers by interacting with tactile displays placed across related shelves. In fact the diffusion of ICTs within future retail places is becoming more and more pervasive.

In such a framework, the main issues of the p-LearNet project are: work-integrated learning and customer learning support, continuous professional training at the workplace, professional learning whatever the place, the time, the organisational and technological contexts of the individual or collective learning and working processes, context-as-construct and seamless learning. In this project, our corporate partners are international retail companies having chains of shops and hypermarkets. The design and engineering of pervasive learning systems must be considered as an interdisciplinary problem requiring the integration of different scientific approaches from computer science, education, commerce, social sciences, etc. Learning focuses mainly on how to support individual and group learning processes through pedagogical guidance and how to enhance the learner's knowledge. Corporate partners at retail workplaces identify the problems and requirements about quality and efficiency of information and services to increase market share and the

corresponding learning goals. In such a framework, several innovative scenarios have been set up according to two main learning and/or working situations for a seller and a customer as learners: i) Seller or customer, outside the shop counters: seller in the back office or storage areas, customer at home or elsewhere; ii) Seller in his department, alone or with a customer having resources from the Smart Spaces (large LCD screen, printers, RFID, etc.) surrounding them [11].

From a technical point of view, we define the seller's workplace as an Intelligent Selling Space (ISS). It appears that ICTs have allowed the development of a new form of commerce called e-commerce with several variants enabled by new technologies, such as Web 2.0, Semantic Web and mobile communication and nomad objects which enrich the interactions and relationships with the client, and augment his/her experience as a user, and finally allow more sophisticated marketing strategies such as mass One-to-One relationships.

III. CONTEXT MODELING

There have been numerous attempts to define the notion of context in different fields. We mainly focus on definitions giving the relationships between the physical world and users' activities. We claim that activities embedded in a particular physical world (or environment) are key issues to give us intention and meaning according to different situations and finally to determine the relevant features describing the different situations. These definitions are as follows: *"learning context is used to describe the current situation of a person related to a learning activity; in addition to attributes relying on the physical world model"* [12]; *" information and content in use to support a specific activity (being individual or collaborative) in a particular physical environment"* [13].

In pervasive learning systems, a context management model must possess the following properties: dynamic and "context-as-construct". According to Dourish, context can be viewed as an interactional problem: i) Contextuality is a relational property that holds between objects or activities; ii) The context is not defined in advance. On the contrary, the scope of contextual features is defined dynamically; iii) Context is an occasioned property, relevant to particular settings, particular instances of activities; iv) Context arises from the activities. Context cannot be separated from activities. It is actively produced, maintained and enacted in the course of activities [14]. In "Big issues in Mobile Learning", context is viewed as "context-as-construct", i.e. "context should be reconceived as a construct that is continually created by the interaction of learners, teachers, physical settings, and social environments" [8] and "learning not only occurs in the context, it also creates context through continual interaction" [1].

A. Context management model

Our context model has to consider the context-as-construct property, the combination of formal (at school, for professionals) and informal (outdoor, at home, at the workplace etc.) learning, the integration of mobile devices in broader educational scenarios, and seamless learning across different contexts. Our context management model is composed of: i) a context model, defining a set of relevant context features (represented as triplets) and their

structure; ii) a set of views, a view consists of a subset of context features which are relevant to a given content category (methods, resources, or web services) and a given domain (at present, learning or working) for adaptation. An adaptation process does not manage the same features for different content categories. In the adaptation process, the current situation, filtered by a view, and the corresponding content description are compared; iii) a set of situations, organized in historical dependencies. A situation is a partial instantiation of the context model consisting of the obtained features describing the current learning or working situation and its physical environment. A user activity can be influenced by his previous work and learning activities. As soon as a new situation occurs, the next activity or the activity continuation can be chosen according to the historical dependencies, for instance, to ensure seamless learning.

Definition 1(Context management model): Let *CMM* be a context management model, $CMM = <CM, CV, S, P>$, *CM* be the context model that is composed of dimensions $CM = \{d_1, d_2, ..., d_n\}$, where *n* is the number of dimensions and d_i is a context dimension; *CV* be a set of different context views $CV = \{cv_1, cv_2, ..., cv_m\}$, where *m* is the number of views, cv_i is a context view. Each cv_i is composed of a set of properties from different dimensions in terms of their usage relating to a domain type, *S* be a set of situations (or historical situations) $S = < \{s_1, s_2, ..., s_p\}, \leq_s >$, where s_p is the current situation. A total order $\leq_s$ on *S*, where $s_i \leq_s s_j$ implies that the situation s_i precedes temporally the situation s_j, *P* be a set of predicates that is used to manage situations including: the change, detection, creation and storage.

Definition 2 (Context dimension): Let d_i be a context dimension, $d_i \in CM$, $d_i = < d_i^c, d_i^m, d_i^p >$, d_i^c be a set of characteristics, each of them is defined by a name and a domain, d_i^m be a subset of context metadata associated to the dimension d_i that is used to manage the dimension d_i, d_i^p be a subset of predicates, used for managing the dimension d_i. These predicates use metadata in d_i^m and dimension characteristics in d_i^c.

The context dimensions are divided into two categories: abstract dimensions and atomic dimensions. An abstract dimension can be recursively broken down into sub-dimensions which are either abstract or atomic. An atomic dimension only consists of a set of characteristics. Our context dimensions are 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 device, the location (place, co-ordinate), the time, the pedagogical tools, the network, the physical environment and the resource (e.g. learning object, services, media resource, system resource, etc.). The management of a context dimension (abstract or atomic) is described by context metadata. It is a set of attributes used by predicates *P*.

Definition 3 (Context view): A context view $cv_i = < P_i^{cv}, AT, CC >$, where $P_i^{cv}, P_i^{cv} \subseteq CM$, is a subset of properties from some dimensions or dimension features to describe a viewpoint related to a particular activity type $AT \in \{learning, professional\}$, and a content category $CC \in \{methods, resources, webservices\}$.

Context views are used to define different viewpoints for adaptation. Consequently, different adaptation categories and seamless learning strategies are specified accordingly.

B. Pervasive learning situations in the workplace

A learning situation in the workplace is composed of physical environment, learning and commercial setting of the user's current work situation. Several context dimensions are combined to describe different pervasive learning situations. In a formal way, a definition of the situation is given as follows:

Definition 4 (Situation): Suppose s_i is a pervasive learning situation in workplace, $s_i = < (e_{i1}, e_{i2}, ..., e_{ik}), t_i^s, t_i^e > \in S$ defines a complete context state associating all interactions between the user and the learning system in the workplace in a given time interval. It is specified temporally with a starting time t_i^s and an ending time t_j^e. e_{ij} is a dimension characteristic in s_i. It is acquired as follows: automatically detected by the learning system or gathered from interactions between the user and the system or dynamically inferred or generated by the learning system (based on inferences or adaptation rules defined in the adaptation model). A situation and its historical dependencies are used to select and/or to continue an activity according to the scenario model.

IV. SCENARIO MODEL

The main role of a scenario model is to integrate mobile devices in broader learning and working scenarios, formal and informal learning and to enable us to manage seamless learning across contexts. As a scenario describes all users' activities (with or without mobile devices, formal or informal learning activities), an author can manage a global activity consistency to deal with the previously mentioned issues. Several research studies in artificial intelligence focus on the hierarchical task model using the task/method paradigm [15-17]. In learning environment, hierarchical task models were also used for designing, for instance, authoring tools [18], learning systems [19-21]. The mechanism of hierarchical and recursive decomposition of a problem into sub-problems is one of the basic characteristics of the hierarchical task model [15-17].

A. The task/method concept

Within the framework of the Task/Method paradigm, tasks represent activities and sub-activities managed by a knowledge-based system. A method describes how a particular task can be achieved. There are two types of tasks: abstract task and atomic task. An atomic task is not composed of sub-tasks. It can be achieved by a simple procedure defined inside a method. An abstract task represents a high level activity composed of sub-activities. A method defines how an abstract task is composed of sub-tasks which can either be abstract or atomic tasks. For a given task (abstract or atomic), several methods can be used to accomplish it. Methods are described by contextual features for adaptation and selection. It is composed of a property set according to the corresponding context view.

A Method, associated to an abstract task, defines a control structure which allows the recursive decomposition of tasks into sub-tasks and the sub-task order at
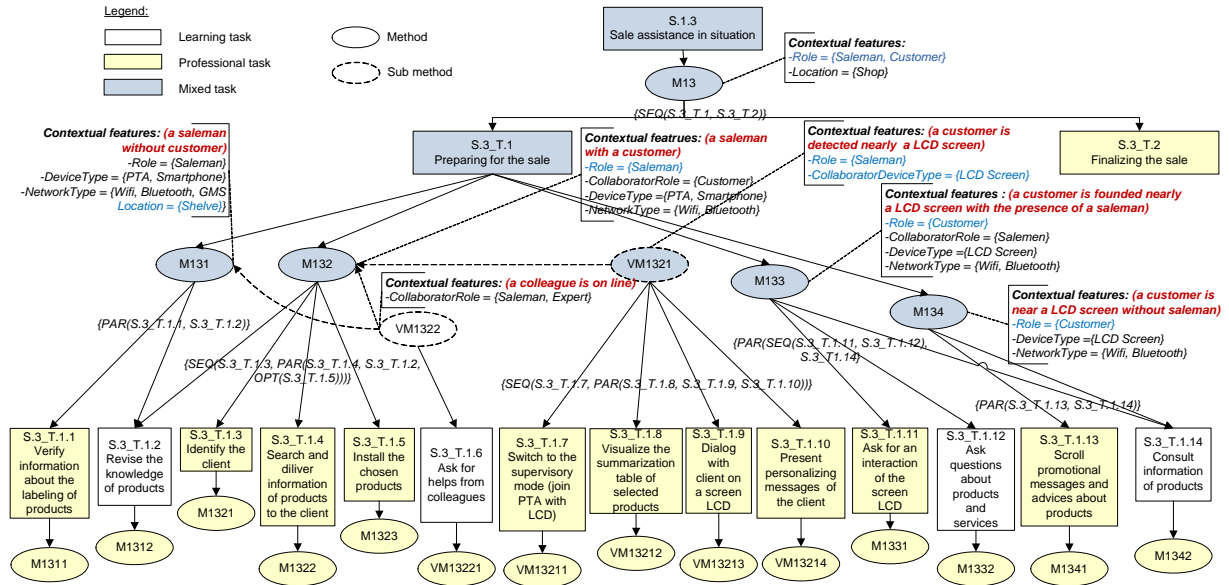
Figure 1.   A small part of the e-retail scenario

runtime - by means of operators. At present, three different operators are used: sequence, alternative and parallel. A Method, associated to an atomic task, can have: i) a resource specification for resource retrieval; ii) a service specification for web service retrieval; iii) a procedure/function specification for a simple procedure or function.

B.   Learning scenario for workplace

Pervasive learning scenarios in 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 Figure 1, the contextual features and the control structure of some methods are represented for different categories of tasks.

Figure 1 shows a part of the e-retail scenario, which represents the decomposition of the mixed task "S.1.3 – Sale assistance in situation" by a method "M13" in a sequence order of sub-tasks ("S.3_T.1" and "S.3_T.2"). The decomposition of the task "S.3_T.1" is made by several methods ("M131", "M132", "VM1321", "VM1322", "M133" and "M134"). Each one is associated to a set of contextual features, which specifies the relevant situations for which the method could be selected. Because the task "S.3_T.1" is an abstract task, hence its methods decompose it into sub-tasks by means of operators determining the sub-task execution order. The method "M131" aims to provide the seller with both activities available ("S.3_T.1.1 - verify information about the labeling of products" and "S.3_T.1.2 - revise the product knowledge") when he approaches to shelves without customers (Location = {Shelve}). The method "M132" achieves the task "S.3_T.1" for the seller with a customer (CollaboratorRole = {Customer}). It is divided into four sub-tasks that are carried out as follows: (SEQ (S.3_T.1.3, PAR (S.3_T.1.4, S.3_T.1.2, OPT (S.3_T.1.5)))) means that we start with the sub-task "S.3_T.1.3" then follow with one of three sub-tasks

("S.3_T.1.4", "S.3_T.1.2" and "S.3_T.1.5") that will be executed in parallel.

Both methods "VM1321 and "VM1322" are the specialized methods which are added to one (or several) other method. The method "VM1321" is aimed at providing the seller with ways to switch his PTA to the supervisory mode when a large LCD screen is available. This method is coordinated with the method "M133" which is designed for the customer (Role ={Customer}) to provide him interactions with the LCD screen. Thus the method "M133" is seen as part of the method "M132". The method "VM1322" aims to provide the seller with a collaborative activity that allows him to ask his colleagues for help or advice on the product or the situation. The methods "M133 and M134" are designed for a large LCD screen (DeviceType ={ LCD screen}) and a client (Role = {Customer}). It is dedicated to the LCD screen to provide interaction functionalities to the customer, so the customer can interact and communicate with the seller according to his requirements. While the "M133" is for the seller (CollaboratorRole = {Saleman}), the "M134" is reserved for the customer who is alone in front of a LCD screen.

V.   CONTEXT MAMAGEMENT PPROCESS

To solve complex and dynamic context changes and demands of situated learning, the learning system detects the situation changes. The context management process generates a new situation or updates the current situation for maintaining activity relevance and continuity. This process is divided into five main stages:

- Stage 1: Context change detection and aggregation: this stage determines context changes and checks whether these changes lead to the creation of a new situation or an update of the current one. The changes can come from collaborators (colleagues) or customers or tutors or learners - user interactions, location, network, device, time, scenario, etc.;

- Stage 2: Gathering of the initial context information to create the new situation, this is done by a partial or

complete copy of the last situation and/or by querying the context ontology;

- Stage 3: Selection of the current task: the last task status can be restored in the new situation to ensure seamless learning and working based on the historical dependencies of situations. When the last task is finished, the next one is selected according to the current scenario and the historical dependencies of situations. When the last task is not finished, it is necessary to continue it;

- Stage 4: Searching for the relevant couples (Task, Method): According to the current situation and the task, the adaptation process has to select the relevant methods. The main role of the adaptation process is to maintain consistency between the learning system, the physical environment and the current learning or working situation and also to ensure seamless learning and working;

- Stage 5: Discovery and execution of relevant web services to serve the current task: Realization of a task requires the discovery, orchestration, invocation and execution of relevant web services. Service requirement specification which specifies the required service functionalities and characteristics is defined in the selected method of the selected Task/Method pair in Stage 4. The adaptation process is hence aimed at searching for the relevant web services to realize the current task according to this specification and the current situation. The specification and management of web services is described in the next section.

VI. SERVICE SPECIFICATIONS AND REQUIREMENTS

In our learning and working scenarios, an activity may be realized by either a simple internal or a set of external services. The problem to solve is to search for a service or a number of services that may be composed in certain order to fulfill the requirements of the current task. Towards this goal, services have to be semantically annotated. In systems without context adaptation, web services are statically bound to a method during design time. In this regard, the problem is to bind, invoke and execute these web services in a "known" order. Take for instance Task "S.3_1.4" in Figure 1 above, web service A can be bound to this task (through its method "M1322") at design time for searching for product information and resources (images, sound messages, documents, etc.). At runtime, logics are built-in to web service A to search for relevant resources annotated with description metadata from repositories or from the database and deliver them to the client of the task making a request.

The solution that we propose in this paper is semi dynamic service searching and matching. The relevant services in this regard can be a single matched service or a set of matched services. The latter is aimed at searching for all the relevant services provided by service providers according to current situation and the service requirement identified in the selected method. This is done based on an adaptation process applied to services (Please refer to Context Management, Service Requirement Specification and Adaptation Process sections). Taking Task "S.3_1.4" above for example again, the realization solution consists of four phases. Each phase is considered as a single service or a set of matched services:

- 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.

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

- Phase 3: The found resources are assembled and adapted according to the current situation. They are filtered and sorted according to their degree of relevance.

- Phase 4: Display and/or deliver the relevant resources to the targeted peripheral devices. All services are managed and executed by our ISS architecture described later.

A. Service requirement specification

Characteristics and functionalities of a service required for a relevant task or activity have to be specified semantically to facilitate accurate and efficient discovery and matching of the right services. The primary goal of a service requirement is the description of 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 should be performed on behalf of the learner in the current situation. By our definition, a service satisfies a service requirement by providing a set of desired output parameters for a desired goal with a set of pre-existing input parameters and situational context features. Our proposed service requirement is summarized as follows:

Functional requirements: describe the capabilities of web services desired by a user. It is characterized by input parameters, expected output parameters, pre-conditions and expected post-conditions;

Non functional requirements: include the identity of a service (e.g. name, owner, type etc.) as well as performance related characteristics, such as Quality of Service (QoS), security etc;

Content requirements: specify a list of domain concepts or a query identifying the content requirements of services (e.g. Mark, Price, ProductModel etc.).

B. Service description

Service request and provision are modeled with a common service description metadata model to formally specify the functional and non-functional requirements of services. 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*. *Interface* dictates how the service can be invoked and what resources are to be assembled to provide the desired functionalities of the service. It is syntactic in nature. We propose to follow the WSDL binding standard for message format and protocol details for interoperability. Hence, the problem of searching and matching the right service provision to a client service requirement in a web service architecture is basically reduced to a problem of matching the service *function* and *behavior* descriptions to the service requirement specification.

In this paper, we propose a service description ontology to describe a service with a service description feature set.

TABLE I.
SERVICE GLOBAL FEATURE SET

Sub 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}
Service Requirement	{listInput(name, type, value, ontologyURI), listOutput(name, type, value, ontologyURI), expectedEffect}
Domain Content	{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}

Table 1 enlists features which are generally applicable to most web services. Those features can be extended to include features which describe more specific functionalities of web services such as pedagogy. The features on a feature list can either be *mandatory* or *optional*. They serve to index a service for *searching*, *filtering* and *ranking* purposes.

VII. SERVICE EXECUTION ARCHITECTURE AND MANAGEMENT

We have derived a generic ISS software architecture organized around a dedicated middleware to support Human-Computer Interaction in different modalities, including speech recognition and speech synthesis. It reuses our previous work about multi-channel and multimodal intermediations between a mobile personal user device and a collection of e-services [23]. However, the case of an ISS is simpler here because the number of services provided is small (specialized local functionalities), and the numbers of specific devices used in the interaction with the customer is, for a precise ISS, relatively small. We can consider that the ISS is relatively autonomous, required few connections with the information system of the shop, and can be easily described in a small ontology such as a micro-world. So dynamic discovering of its services will be easier than in most of the Ubiquitous Computing projects, e.g. Smart Homes. Our generic software architecture takes into account the modularity of the ISS (adjunction or suppression of some interactive elements, needs for adaptation to a particular retail company) and its openness to the Personal Sale Assistant (PSA). Figure 2 gives an overview of the software architecture.
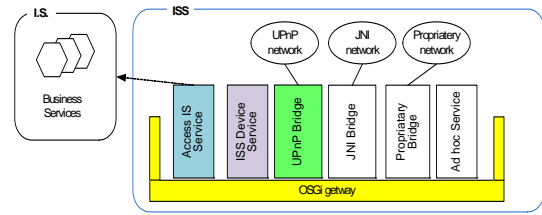

Figure 2. An overview of an IIS architecture

An ISS is built by assembling UPnP services [24] into an OSGi [25] gateway: i) UPnP is a wired IP protocol allowing the creation of spontaneous networks of devices (TV sets, HVAC, light control etc.) and control points (PDA, Smartphone, touch panels etc.). UPnP enables the live detection of devices and the use of their services by the control points; ii) OSGi framework allows deploying and redeploying Java-based plug-in applications (bundles) offering services [26]: a) thanks to different OSGi bridges, ISS devices can be seen as UPnP devices. The gateway allows plugging in any kind of devices as long as the OSGi driver is available. Each device company can provide drivers; OSGi and UPnP are open protocols; b) I.S. (Information System) Access Service provides the access to business objects; c) thanks to ISS Device Service, ISS can be seen as an UPnP device itself. This allows a "control point" to manage the ISS. It is what we call a "remote control device"; d) OSGi gateway can also provide ad hoc services linked with the particularities of the ISS.

VIII. ADAPTATION

The fundamental issue in a pervasive learning environment is how to provide learners with the right learning content at the right time in the right way. Thus, adaptation is mandatory to all types of learning activities in pervasive learning environments [6]. At present, we focus on adaptation mechanisms dedicated to two categories: scenarios and (web) services. At the scenario category, adaptation is aimed at accomplishing an activity according to the current situation, or in other words, how to select the relevant methods for a given task. Put differently, the learning system has to select dynamically the relevant way to achieve the different tasks included in a scenario. At the service category, adaptation is to refine the service retrieval process for a realization of an atomic task. Thus, the learning system needs to select the relevant web services according to the current situation. As context is dynamic, it is not possible to know in advance how the next situation will be structured. In other words, it is not possible to anticipate the set of features composing the different situations. For managing adaptation, it is necessary to trigger rules using the contextual features of methods and the service description of web services which are defined a priori and the set of situation features which can be unexpected – not known in advance. Consequently, it is not possible to define rules for each possible configuration of features in a situation.

Like Mobilearn European project [27], we associate specific metadata to situation properties and/or dimensions for managing adaptation. Situation features are divided into two categories: permanent and transitory. Features describing scenarios and users which are available in all situations are permanent features in a situation. Thus, it is possible to manage them as usual

because they are known a priori. Other features are transitory. For them, it is necessary to analyse how each property contributes to content adaptation – methods and services selection - differently according to its role in the adaptation process. Some of them are used to filter the content while others are used to rank or annotate it where filtering, ranking and annotation are the adaptation techniques. For instance, learning and working methods can be filtered according to learning places or used devices while it is annotated according to the user's knowledge, know-how or preferences). As soon as features are used to filter content (methods or services), it is easy to manage them. In other words, content will or will not be filtered out by a present transitory feature. At present, all transitory features of content are used to filter. Nevertheless, we shall have to investigate this issue in depth according to more detailed scenarios in future.

A. Adaptation process

The adaptation process is specified for a content type (methods and services) and an adaptation category. The input content of the adaptation process can be achieved in different ways depending on the content type: input methods are specified directly by the current task while input services are selected by a search process based on a query which "compares" the service requirement of the selected task/method coupled with the service description. The three stages of the adaptation process are presented as follows (see Figure 3):

- Evaluation/Classification: input content is classified according to the current situation in several equivalence classes: two classes *{"Good", "Bad"}* for each transitory feature and up to five equivalence classes *{"VeryBad", "Bad", "ToConsider", "Good", "VeryGood"}* for all permanent situation features, together. The content belongs to an equivalent class if it satisfies its membership rules;

- Filtering: all content belonging to "Bad" classes according to a transitory feature are filtered out. In other words, these content are discarded. For example, with the network dimension, the class *"Good"* is considered as relevant while the class *"Bad"* is not. This means that the system hence will eliminate all content that belongs to the class *"Bad"*;

- Adaptive navigation: permanent situation features are used to evaluate and classify the remaining content. An adaptive technique can be chosen by the system or by the user according to an author decision. All content belonging to the same equivalence class are treated in the same way. Annotation and sorting are processed according to the total order of equivalence classes. For hiding, only contents belonging to the class *"Good"* and *"VeryGood"* are maintained.
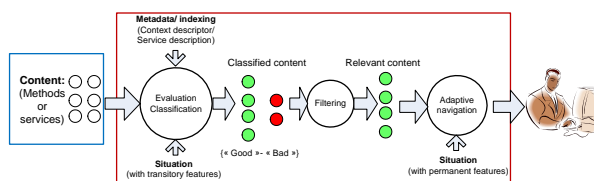

Figure 3. Adaptation principle

IX. CONCLUSION

We have proposed an adaptive and context-aware model of a scenario based on a hierarchical task model having the task/method paradigm - with methods defining how to achieve a task - for a pervasive learning system supporting working and learning situations. This model enables us to choose how to achieve activities according to the current situation. In other words, the relevant methods and services are selected dynamically according to the current situation for the realization of activities. We have also integrated web services described at semantic level for indexing services and an ISS architecture for executing them. Our context model and adaptation process deal with dynamic "context-as-construct" by means of transitory and permanent situation features managed in different ways. As an item for future work, we will study in greater depth our adaptation policies to manage dynamic context because at present, detail of our scenarios is limited.

REFERENCES

[1] N. Balacheff, "10 issues to think about the future of research on TEL," Les Cahiers Leibniz, Kaleidoscope Research Report, 2006.

[2] J. Hundebol and N. H. Helms, "Pervasive e-Learning - In Situ Leaning in Changing Contexts," 2006.

[3] M. Sharples, "Learning As Conversation: Transforming Education in the Mobile Age," in Proceedings of Conference on Seeing, Understanding, Learning in the Mobile Age Budapest, Hungary, 2005, pp. 147-152.

[4] T. Siobhan, "Pervasive Scale: A model of pervasive, ubiquitous, and ambient learning," in An International Workshop on Pervasive Learning, in conjunction with Pervasive 2007, Toronto, Ontario, Canada, 2007, pp. 10-18.

[5] C. Brodersen, B. G. Christensen, K. Gronboek, C. Dindler, and B. Sundararajah, "eBag: a ubiquitous Web infrastructure for nomadic learning," in Proceedings of the 14th international conference on World Wide Web Chiba, Japan: ACM Press, 2005.

[6] 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.

[7] V. Jones and J. H. Jo, "Ubiquitous Learning Environment: an Adaptive Teaching System using Ubiquitous Technology," in ASCILITE, Perth, Australia, 2004, pp. 468-474.

[8] 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.

[9] p-LearNet, "p-LearNet project," 2006.

[10] C. Paris, M. F. Wu, K. V. Linden, M. Post, and S. Lu, " Myriad: An Architecture for Contextualized Information Retrieval and Delivery," in Proceedings of the 3th International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems Eindhoven, The Netherlands: Springer, 2004, pp. 205-214.

[11] A. Derycke, V. Chevrin, and T. Vantroys, "P-Learning and e-retail: a case and a flexible Sotware Architecture," in Pervasive Learning 2007, an International Workshop on Pervasive Learning, Toronto, Ontario, Canada, 2007, pp. 43-50.

[12] M. Derntl and K. A. Hummel, "Modeling Context-Aware E-learning Scenarios," in Pervasive Computing and Communications Workshops, 2005. 3rd IEEE PerCom 2005 Workshops. , 2005, pp. 337-342.

[13] A. Kurti, M. Milrad, F. Alserin, and J. Gustafsson, "Designing and implementing ubiquitous learning activities supported by mobile and positioning technologies," in the Ninth IASTED International Conference computers and Advanced Technology in Education, Lima, Peru, 2006, pp. 193-199.

[14] P. Dourish, "What we talk about when we talk about context," Personal Ubiquitous Comput., vol. 8, pp. 19-30, 2004/2// 2004.

[15] B. Wielinga, W. V. d. Velde, G. Schreiber, and H. Akkermans, "The KADS Knowledge Modelling Approach," in Proceedings of the 2nd Japanese Knowledge Acquisition for Knowledge-Based Systems Workshop, R. Mizoguchi and H. Motoda, Eds. Hitachi, Advanced Research Laboratory, Hatoyama, Saitama, Japan, 1992, pp. 23-42.

[16] F. Trichet and P. Tchounikine, "DSTM: a Framework to Operationalize and Refine a Problem-Solving Method modeled in terms of Tasks and Methods," International Journal of Expert Systems With Applications, Elsevier Science, vol. 16, pp. 105-120, 1999. (doi:10.1016/S0957-4174(98)00065-7)

[17] J. Willamowski, F. Chevenet, and J. M. François, "A development shell for cooperative problem-solving environments," Mathematics and computers in simulation, vol. 36, pp. 361-379, 1994. (doi:10.1016/0378-4754(94)90070-1)

[18] M. Ikeda, K. Seta, and R. Mizoguchi, "Task Ontology Makes It Easier To Use Authoring Tools," in IJCAI, 1997, pp. 342-351.

[19] M.-L. Betbeder and P. Tchounikine, "Structuring collective activities with tasks and plans," in IEEE International Conference on Advanced Learning Technologies (ICALT'2003), 2003.

[20] C. Ullrich, "Course generation based on HTN planning," in Proceedings of 13th annual Workshop of the SIG Adaptivity and User Modeling in Interactive Systems, J. A. and B. B., Eds., 2005, pp. 75-79.

[21] C. Choquet, F. Danna, P. Tchounikine, and F. Trichet, " Modelling Knowledge-Based Components of a Learning Environment within the Task/Method Paradigm," in Intelligent tutoring Systems, ITS'98, San Antonio (USA), 1998.

[22] N. Milanovic and M. Malek, "Current Solutions for Web Service Composition," Berlin 2004.

[23] V. Chevrin, S. Sockeel, and A. Derycke, "An Intermediation Middleware for supporting Ubiquitous Interaction in Mobile Commerce," in ICPS'06, IEEE International Conference on Pervasive Services, Lyon, 2006, pp. 321-324.

[24] UPnP, "UPnP, Official web site," 2007.

[25] OSGi, "OSGi Alliance," 2007.

[26] D. Donsez, "On-Demand Component Deployment in UPnP Device Architecture," in CCNC 2007, 2007.

[27] P. Lonsdale and R. Beale, Towards a dynamic process model of context. To appear in Proceedings of Ubicomp 2004 workshop on Advanced Context Modeling, Reasoning and Manage, 2004.

AUTHORS

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

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

Benjamin Barbry, is with the LIFL lab, University of Sciences and Technologies of Lille, Villeneuve d'Ascq, France (email: b.barbry@ed.univ-lille1.fr).

Thomas Vantroys, is with the LIFL lab, University of Sciences and Technologies of Lille, Villeneuve d'Ascq, France (email: thomas.vantroys@univ-lille1.fr).

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