TOWARDS EXTENDING IMS LD WITH SERVICES AND CONTEXT AWARENESS

Application to a Navigation and Fishing Simulator

Valérie Monfort [1,2], Slimane Hammoudi[3] and Maha Khemaja[4]

[1]*Université de Sfax, MIRACL, Tunisie*

[2]*Université de Paris 1, Panthéon Sorbonne 90 rue de Tolbiac, 75634 Paris cedex 13, France*

[3]*ESEO 4, Rue Merlet de la Boulaye, B.P. 9249 009, Angers Cedex 01, France*

[4]*PRINCE Research Group, ISITC, GP1 Hammam Sousse 4011, University of Sousse, Tunisia*

Keywords: e-Learning, LD Standard, Web Services, Context Aware Adaptation, Meta Modeling, MDD.

Abstract: A few e-Learning platforms propose a solution for ubiquity and context aware adaptability. Current standards, as Learning Design (LD), require an extension to propose context awareness. Based on previous related works, we define a fully interoperable and learner (ambient) context adaptable platform, by using metamodeling based approach mixing MDD, parameterized transformations, and models composition. The scope of this paper is to extend LD metamodel as a first step. We use a concrete software engineering industrial product that was promoted by French Government.

1 INTRODUCTION

E-learning aims the delivery of a learning, training or education program by electronic and it involves the use of a computer or electronic device (e.g. a mobile phone), in some way, to provide training, educational or learning material. Concerning the architecture point of view, e-learning platforms gather two separated and distributed parts as: authoring tools (for pedagogical contents definition) and execution platforms. So, e-learning may: i) use several media and devices, ii) promote specific training according to learner skills, iii) send specific events to increase complexity of lessons and to assess learner reactions, …

Previous works allowed us to use Web services to get interoperability and flexibility to changes. But, we noticed the lack of adaptability, so, we extended Web services to introduce adaptability with aspects (Kiczales, 1997) (Tomaz, 2006). We noticed this very efficient and pragmatic solution was very technical. Recently, we have investigated a model driven approach and context awareness to provide developers mechanisms that allow them representing an application in abstract way (in a model) and, then, automatically generating the corresponding code (Monfort, 2010), (Monfort, 2009). We aimed to explore adaptability and flexibility on a service platform using context with the benefits of an MDD (Model Driven Development) (OMG, 2001) development strategy. These benefits are related to productivity, quality, adaptability and maintenance.

Moreover, e-learning standards as Learning Design (LD) tend to extend their semantic to Web services standards (Dietz, 2004). We studied e-Learning standards meta models, but, we noticed no semantics concerning context aware adaptability.

We aim to propose a fully interoperable and learner (ambient) context adaptable platform, firstly by using meta modeling based approach mixing MDD, parameterized transformations, and models composition. This paper is based on a concrete industrial project and we present here a part to illustrate our research work. This project aims to develop a concrete navigation and fishing simulator. This e-learning application allows different learners to discover navigation and fishing business and to become fishery captain, boat mechanics, sailor, …

We shall process as followed. The second section presents a part of the fishing and navigation simulator project. The third section shows the diversity of e-learning standards, the efforts of connection between e-learning and Web services standards, and focuses on LD. The fourth section

Monfort V., Hammoudi S. and Khemaja M. (2010).
TOWARDS EXTENDING IMS LD WITH SERVICES AND CONTEXT AWARENESS - Application to a Navigation and Fishing Simulator.
In *Proceedings of the 12th International Conference on Enterprise Information Systems - Information Systems Analysis and Specification*, pages 80-91

introduces context aware meta model. The fifth section presents parameterized transformations and our global approach. The sixth shows a composition example between LD and context aware meta model as an extension of LD meta model based on our industrial project. Last sections present some related works and conclusion. Let us present our project.

2 THE FISHING SIMULATOR PROJECT

We aim to design and implement a genuine fishing simulator. This software was intended to navigation and fishing schools or fishing ship owner companies. Following partial UML model (fig.1) shows the different services proposed by the navigation and fishing e-learning system. This platform provides diploma for students (learners) in fishing schools and certificate for companies. The Teacher may be human or not. The system can take decision and can send specific events to complicate lesson if the student has a good level.

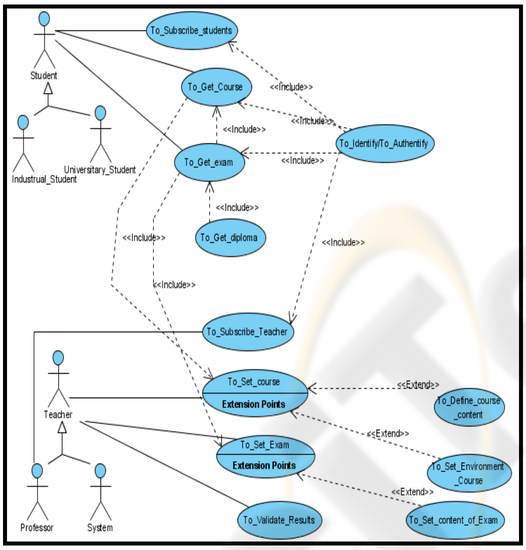

Figure 1: e-learning Services modeling.

Fig. 2 shows a process where the Teacher is preparing training and is sending it to the student who identifies himself by invoking identification and authentification services, linked to rules manager. While training learner may receive specific events during current lesson to assess learner's skills. The learner is evaluated at any time. This platform provides diploma for students (learners) in fishing schools and certificate for companies. The Teacher may be human or not. The system can take

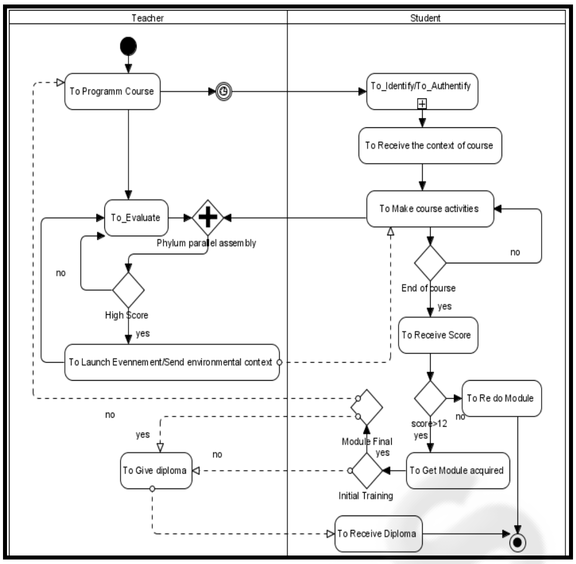

Figure 2: Training services according to BPMN.

decision and can send specific events to complicate lesson if the student has a good level.

Fig. 2 shows a BPMN process where the Teacher is preparing training and is sending it to the student who identifies himself by invoking identification and authentification services, linked to rules manager. While training learner may receive specific events during current lesson to assess learner's skills. The learner is evaluated at any time.

The following e-learning architecture (fig.3) shows services proposed by providers. Actors as learner and teacher can work anywhere and use different media. Services are available according to SaaS model and managed by providers according to Cloud Computing (Reese, 2009) principles. Contents are defined by the teacher with authoring tools according to standards (see previous section). He defines e-learning tasks sequences, and so, e-learning services orchestrations. He informs the services providers by loading training content and the concerned learner. The learner may use genuine navigation equipments and/or simulation. The genuine navigation equipments are linked to a middleware able to interpret signals coming from equipments and to send them to Execution platform. The learner may also use PDA or mobile phone. Learner receives the training scenario and while training sends information and invokes services. The e-learning middleware (ESB for Enterprise Service Bus) manages: routing messages, transporting messages and transforming exchanged data.

The providers repositories manage at least these basic following services as: business, training supervisor, course virtual management, planning management, collaborative management, subscribing management, time/tracking management, …

The fishing and navigation e-learning platform architecture is fully flexible, it is able to take into account any functional or technical changes. E-learning platforms are generally based on specific standards. Let us see how these standards contribute to converge on Web services standards that support interoperability between systems.

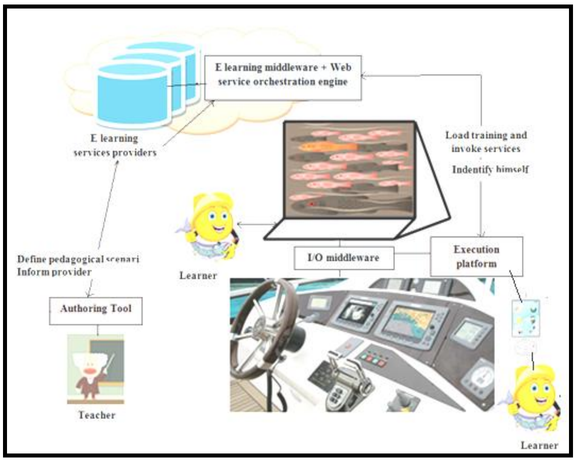

Figure 3: General architecture.

3 E-LEARNING STANDARDS

3.1 Convergence on Web Services Standards

Many definitions may be found, we can sum up them with this definition: e-learning aims the delivery of a learning, training or education program by electronic means. E-learning involves the use of a computer or electronic device (e.g. a mobile phone) in some way to provide training, educational or learning material. E-learning systems gather two separated and distributed parts as: authoring tools (for pedagogical contents definition) and execution platforms.

Authoring tools are based on several standards as: LOM (Learning Object Metadata) to define a structure for elements and meta data useful to describe pedagogical resources. Content Packaging proposes to describe a pedagogical data package referencing resources in a set of interoperable packages.

Learning Design defines a pedagogical scenario with its components (roles, activities, environment, results). SCORM (Sharable Content Object Reference Model) is a set of technical standards allowing to find, import, share, reuse, export,… normalized teaching contents.

On the other hand, IMS-GWS (General Web Services Standards) allows interoperability between any Web services. IMS-GWS may be compared to WS-IBP (WS-I Basic Profile) to promote interoperability with Web services and profile. They use the same mechanisms and semantics based on SOAP, WS-Security and Addressing and allow referencing a document into SOAP message.

IMS-TI (Tools Interoperability) and WS-Federation also promote interoperability between e-learning platforms and/or components. They aim to build a trustable architecture by using WS-Security tokens. IMS-TI uses IBAT tools based on UML meta modeling and models transformations from one platform to another. BPEL and IMS-SS (Simple Sequencing) may be compared as far as concerned training tasks orchestration. IMS-SS provides a simple description and BPEL semantic is wider. Other e-learning standards are useful to manage pedagogical contents. They are all based on XML so they are Web services compliant.

In following section we focus on Learning Design (LD) that helps to define a pedagogical scenario with its components (roles, activities, environment, and outcomes).

3.2 LD Model

LD is a description of a method enabling learners to achieve intended learning objectives and outcomes by performing predefined learning activities. More specifically, a learning design is a means allowing the Instructional designer to describe a learning scenario in terms of a set of activities that learners should perform according to the different roles that they may play within environments (e.i. Run-time environment). Environments are described in terms of Learning Objects and Services that should assist learners during the-learning process. IMS-LD (Instructional Management Systems-Learning Design) (Boticario, 2007) (Davinia, 2004) specification provides for previously described concepts a meta-model (fig.4.) that was and is still used by LD authoring tools developers. According to IMS-LD specification, LD concepts must meet height requirements. We name the third one because it deals with personalization that is relevant to our work. The LD specification states that: “The content and activities within a unit of learning can be adapted based on the preferences, portfolio, pre-knowledge, educational needs, and situational circumstances of users. In addition, the control over the adaptation process must be given, as desired, to the student, a staff member, the computer, and/or the designer”. However, IMS-LD provides neither means nor modeling solutions to take into account contextual data of mobile users for instance. We should stress that the contextual data, unlike those already defined for personalization, are dynamic and may depend on the user’s external environment.

We notice LD proposes an entity for context in its meta model but, according to us, this approach is too semantically poor. Moreover, this entity does not take into account ambient and/or context adaptation. It is the reason why we want to propose to extend this model with model showed in fig.2 with composition mechanisms according to (Klein, 2006) (Lundesgaard, 2007). But, first of all, let us define what we mean by Model Driven Development and “context aware”.

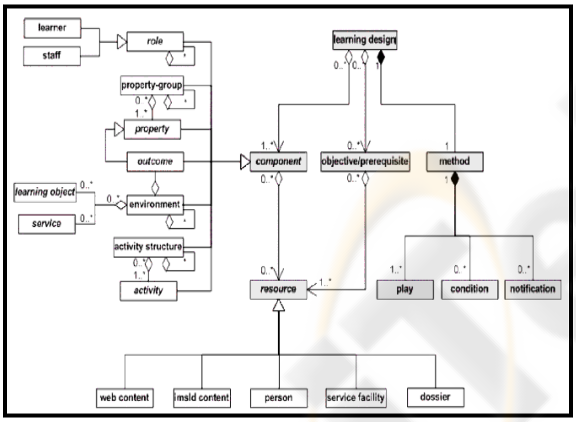

Figure 4: Semantic aggregation in LD specification (http://tecfa.unige.ch/guides/tie/pdf/files/pedago-ld.pdf).

4 CONTEXT AWARENESS MODELING

4.1 Model Driven Development

At the beginning of this century, software engineering needs to handle software systems that are becoming larger and more complex than before. Object-oriented and component technology seem insufficient to provide satisfactory solutions to support the development and maintenance of these systems. To adapt to this new context, software engineering has applied an old paradigm, i.e. models, but with a new approach, i.e. Model Driven Development (MDD). In this new global trend, Model Driven Architecture (MDA) is a particular variant. MDA is based on standards from the Object Management Group (OMG) (Tomaz, 2006); it proposes an architecture with four layers (Dey, 2001): meta metamodel, metamodel, model and information (i.e. an implementation of its model). MOF (Meta Object Facility) is a standard from OMG for metamodels specification. The development is based on the separation of concerns (e.g. business and technical concerns), which are afterwards transformed between them. So, business concerns are represented using Platform-Independent Model (PIM), and technical concerns are represented using Platform-Specific Model (PSM). Finally, it is well recognized nowadays that model transformation is one of the most important operations in MDA. In the context of the basic four levels Metamodeling architecture of MDA, various scenarios of model-to-model transformation have been identified. Fig.5 presents the most common scenario of these transformations, which is compatible with the MOF2.0/QVT standard (Tomaz, 2006). Each element presented in Fig. 5 plays an important role in MDA. Transformation rules specify how to generate a target model (i.e. PSM) from a source model (i.e. PIM). To transform a given model into another model, the transformation rules map the source into the target metamodel. The transformation rules are based on a transformation language, such as the standard QVT. The transformation engine takes the source model, executes the transformation rules, and produces the target model as output.

Adaptable Service platforms have been proposed for the development of mobile context-aware applications. The development of such platforms involves a number of challenges from which we consider two main issues in the context of our approach of model driven development: i)the definition of a metamodel to describe the contextual domain in which a given application or service is defined, ii)a mechanism to integrate the context into the business application using a model driven approach. These two main issues are described in the sequel.

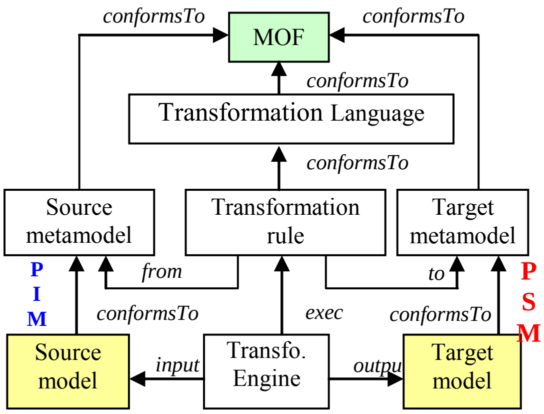

Figure 5: Architecture of Transformation Process.

4.2 MDD and Context for Service Adaptability

Context awareness is a quite new discipline in e-learning domain. For instance, in (Strang, 2004)(Mary, 2005), the authors noticed the context acts like a set of constraints that influence the behavior of a system (a user or a computer) embedded in a given task. They discussed the nature and structure of context but they notice the lack of representation of context in e-learning domain. The emergence of new technologies, in particular wireless communications and the increasing use of portable devices (smart phones, Personal Digital Assistants (PDA), laptops…), has stimulated the emergence of a new computing paradigm called: pervasive computing. In fact we have moved from the desktop computing paradigm to the mobile and ubiquitous computing paradigm. Pervasive computing refers to the seamless integration of devices into the user's everyday life. "Appliances should disappear into the background to make the user and his tasks the central focus rather than computing devices and technical issues."(Weiser, 1991). Computing applications now operate in a variety of new settings; for example, embedded in cars or wearable devices. They use information about their context to respond and adapt to changes in the computing environment. They are, in short, increasingly context aware. The context awareness of such applications is the subject of a recent field of studies in pervasive computing called: context-aware systems. This terminology was discussed in (Shilit, 1994) and presented as "software that adapts according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time". Since then, there have been numerous attempts to define context-aware computing. (Strang, 2004), (Tomaz, 2006) define context-awareness as the ability of a program or device to sense or capture various states of its environment and itself. Referring to these latter definitions a context-aware application must have the ability to capture the necessary contextual entities from its environment, use them to adapt its behavior (run time environment) and finally present available services to the user. In this sense and to describe context-awareness independently from application, function, or interface, (Pascoe, 1998) proposes four features of context-aware application : (1) Contextual sensing which refers to the detection of environmental states and their presentation to the user; (2) Contextual adaptation refers to the adaptation of application behavior to the current context; (3) Contextual resource discovery is the use of context data to discover other resources within the same context; (4) Contextual augmentation in which the environment is augmented with digital data associated to a particular context.

In (Dey, 2001), the authors introduce another definition in which they insist on the use of context and the relevance of context information. The authors consider that: *"a system is context-aware if it uses context to provide relevant information and/or services to the user, where relevance depends on the user's task"*. They explain how to use context and propose a classification of the features of context-aware applications that combine the ideas of (Shilit, 1994). They consequently define three categories of features that context-aware applications may support as (1) presentation of information and services to the user; (2) automatic execution of a service; and (3) tagging of context to information for later retrieval.

All the works converge to a general architecture composed of five ordered layers presented in fig.6. The complete description of every layer is given in (Strang, 2004).

Application/Services
Storage/Management
Preprocessing
Raw Data Retrieval
Sensors

Figure 6: Architecture of Context-Aware System.

"Context Information" in context-aware systems and what could be the definition of context has been the topics of many recent works. Various definitions are given and are summarized in (Mary, 2005).

Other definitions are extremely broad; the most popular one is given by (Dey, 2001): "Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves". The authors give a general definition that can be used in a wide range of context-aware applications. To refine their definition, they identify four categories of context that they feel are more practically important than others. These are location, identity (user), activity (state) and time (Dey, 2001). In (Winograd, 2001) the author approves this definition and claims that it covers all proposed works in context. However he considers it as a general definition that does not limit a context. Thus he proposes his own definition in which he limits a context in "a set of information, which is structured and shared. It evolves and is used for interpretation". We stress that the notion of hierarchy (structure) of context introduced by (Winograd, 2001) is important. The definition proposed in (Chen, 2000) also presents the context as hierarchically organized. In this work the authors differentiate between environmental information that determines the behavior of mobile applications and that which is relevant to the application. They thus define the context as "the set of environmental states and settings that either determines an application's behavior or in which an application event occurs and is interesting to the user".

4.3 Maintaining the Integrity of the Specifications

We consider that the defining context here is a set of information structured in three dimensions:
- Actor: A person which is a central entity in our system.
- Environment: in which the person evolves and
- Computational devices which are used by a person to invoke services and capture the different states of the environment. All the information related to the three dimensions can also be shared by other mobile applications. According to (Monfort, 2010), (Monfort, 2009), fig.7 shows our context metamodel. Our metamodel identifies and adds the most relevant and generic contextual entities that will be held in account in modeling any mobile and context aware application. This context meta model consists of six generic contextual entities (represented in dashed colour), and four deduced entities specific to a category of mobile applications. The class "ContextView" groups all contextual entities involved in a given application. It is identified by name attribute and has two types of relation: the aggregation "involves" and the association "belongsTo". The first relation expresses that a given "ContextView" is composed of many "ContextEntity" that are involved in a context-aware application. The second relation "belongsTo" expresses the use of historical context information. A given context entity may have participated in different context views. This information can be helpful in the design of future context views. The second generic entity of the metamodel is the "ContextEntity". As we see on the figure bellow, it is specialized in three generic entities: *Actor*, *ComputationalEntity* and *Environnement*. Actor may be a person or another object that has a state and profile. It evolves in an environment and uses computational devices to invoke services. With the ComputationalEntity, the computational device is used by the actor to access the services and to capture contextual information from the environment. Usually, a mobile device is used in context aware mobile applications, and can obtain information concerning the type of device it is (PDA, laptop, cellular phone…), the application, the network, etc. The environment is constituted of all the information surrounding the actor and its computational device that can be relevant for the application. It includes different categories of information as:

i) Spatial context information can be location, city, building,

ii) Temporal context information comprises time, date, season,

iii) Climate can be temperature, type of weather… The last entity is a *profile*. We are convinced this entity is important in any user centred context aware application. In fact, profile is strongly attached to the actor and contains the information that describes it. An actor can have a dynamic and/or a static profile, and as ContextEntity class, he owns a status. The static profile gathers information relevant for any mobile context-aware application. It can be the "date of birth", "name" or "sex". On the opposite, dynamic profile includes customized information depending on the specific type of application and/or the actor. It can be goals, preferences, intentions, desires, constraints, etc.

5 PARAMETERIZED TRANSFORMATIONS FOR CONTEXT BINDING: GLOBAL APPROACH

5.1 Parameterized Transformations

The separation of concerns (business and context) is emphasized at a model level of our approach where PIM and context models are defined independently, and then merged by suitable transformation techniques. Two types of transformations are involved in our proposal. The first type of transformation called "Parameterized transformation" allows merging context information with business logic at model level.

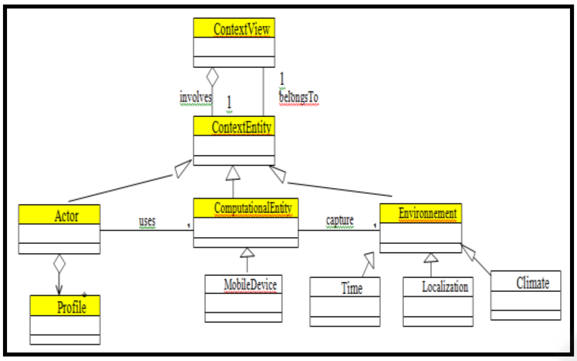

Figure 7: A Context Meta Model.

We have investigated (Vale, 2008) this type of transformation which is not explored and there is not a standard transformation language implementing it. We will discuss shortly this type of transformation. A CPIM model (Contextual Platform Independent Model) is then obtained and fits together business requirements with contextual data. The second type of transformation is the traditional transformation technique using a language such as QVT, which operates in two steps.

Based on parameter paradigm, parameterized transformation is a transformation technique which is not sufficiently explored nowadays and there is not a standard transformation language implementing it. OMG (OMG, 2005) has defined the concept of parameter as follows:

"A parameter specifies how arguments are passed into or out of an invocation of a behavioural feature like an operation. The type and multiplicity of a parameter restrict what values can be passed, how many, and whether the values are ordered".

In (Frankel, 2003) David Frankel mentions the importance of parameterization in model operations using the association of tagged values with PIM and PSM models. Tagging model elements allows the model language to easily filter out some specific elements. Transformation by parameter could be used to improve new functionalities (values, properties, operations) or to change the application behaviour (activities).

In our approach, we are convinced that parameterized transformation focusing on PIM to PIM transformations is the fitted solution. The designer must specify the parameters to be inserted at the transformation phase. In our proposition these parameters are context or context-aware and after the transformation the application will join the context information specified into the parameters as illustrated in Fig. 8. A PIM model can be developed without contextual details. User name, profiles, device type, location can be added as parameters in transformations. The same PIM can be re-transformed and refined many times adding, deleting or updating context information. The designer has to specify into the application model the elements that will receive the context information. A mark, identified by the symbol #, is given for these elements to be recognized by the transformation engine. The marked elements represent context-aware elements, in others words, the model elements that can be contextualized.

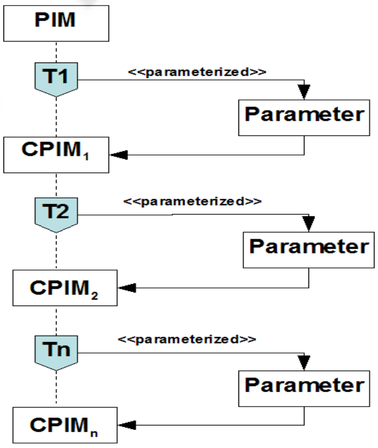

Figure 8: Parameterized Transformation Concepts.

The transformation language must support parameterization techniques. In our case the parameters can be a Context Property and/or a Context Data Type. We use templates to specify which elements in application model are potentially context-aware as depicted in Fig. 9. The transformation engine has to navigate into the PIM model verifying the parameters and the elements marked and then make the transformation which consists in an update of contextual properties in a PIM.

Template parameter (Vale, 2008) is an element used to specify how classifiers, packages and operations can be parameterized. UML 2.0 presents that any model element can be *templateable*. For independent context-aware models we need to identify context elements that could be *parameterable*. A *parameterable* element is an element that can be exposed as a formal template parameter for a template, or specified as an actual parameter in a binding of template (Vale, 2008).

Context parameter can be expressed as constraint and compared with the elements signature in template parameter. This operation is named "*matching* operation". UML presents a Template Signature element that defines the signature for binding the template. Lets us see now related works concerning this approach.

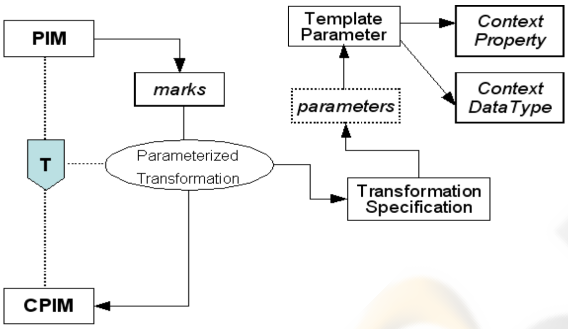

Figure 9: Context binding.

5.2 Our Global Approach

This section aims to present the two techniques of context and aspect could be combined to achieve service adaptability using a model driven approach. Through Model Driven Development, context models are built as independent pieces of application and at different abstraction levels then attached by suitable transformation techniques called parameterized transformation. Context model specify contextual entities that are involved in a given context aware application. From a context model, an aspect based services (ABS) model is derived. This aspect model specifies the behaviors linked to the context model.

- Fig. 11 illustrates the main models and transformations techniques involved in our MDD approach. Five main objectives are illustrated: A separation between context information (CM) and business
- logic (PIM) in individual models,
- The derivation of an aspect based service model (AM) from a context model. A context model specifies the contextual entities with their properties (static view), while the service based aspect model specify behaviors (dynamic view).
- The integration of the context model into the business logic using parameterized transformation techniques. At this stage, the CPIM model is enriched by contextual data but the behavior part for adaptability at execution level is missing.
- The Weaving process adds adaptability mechanism producing a CPSM model.
- Finally, A CPSM model is mapped into a service platform for future execution of context-aware services.

Notice Aspect Based services weaving is a specific model dynamic composition as related to (Klein, 2006). In our approach aspect may be purely business (AM) or technical (CPSM). Faced to these previous research works and to our observations concerning e-learning standards we propose now to extend LD standard with model composition based on our concrete case study.

6 EXTENSION OF LD TO CONTEXT AWARENESS

6.1 Model Composition

Let us see now fig. 4 and fig. 7 and focus on "Component " and "Context View" classes respectively. We propose following part model (fig.10.) that shows Component Class may be composed of Context View class and, so, compose the two approaches (Monfort, 2009): (ambient) context awareness and LD meta models.

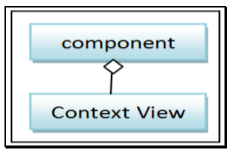

Figure 10: LD and Context Meta Models composition.

During course, learner according to his skills receives a navigation and/or fishing scenario as " go to 100 miles from Saint Jean de Luz and fish tuna". So, the learner has to do obligatory tasks as: to check weather, to define the road, to check the fitted nets, to check mechanic,… The generic BPMN (Business Process Modeling) process shows (fig.2) the different tasks to do by the learner and the teacher. The teacher programs a course that will be received by the learner anywhere he is, via any media, … after identifying himself. The learner is assessed in real time and the teacher may send him events. At the end of the module, the diploma is delivered or not.

The learner is in front of his laptop and receives the training. All the navigation tools (radar, sounder, GPS,…) are simulated. According to his skills, the teacher (human or system) can send to the learner desktop specific events as mist, rain,… and the learner has to react properly. Moreover, the system provides an estimation of learner skills in real time.

The following partial models show a resulting composed model: i) training meta model (that could be later formatted to e-learning standards), ii) a contextual model that was already composed to component class from LD. Another composition may also be done with fishery business metamodel. For each training module, a link may be done with specific business data. For instance, a training module about tuna fishery involves the choice of the fitted net. A mark is put on the required classes.
We have now to describe these models with Kermeta tool and to define transformation rules from these business models to technical models (corresponding to the implementation platform).

6.2 Model Instantiation

Fig. 13 shows the different layers mentioned previously. We present a part of fig. 12 as a Meta model including for instance the following classes: actor, profile, environment (Weather, location, mobility settings). M1 classes instantiate M2 meta classes and M0 instance level includes all the instances of M1 classes. Let us see related works.

7 RELATED WORKS

Among related works, we can distinguish on one hand those dedicated to e-learning as CSCW and CSCL and, on the other hand, i) those that use MDD for general context aware applications, ii) and those that apply MDD and context for e-learning application. We focused on context adaptability and to our opinion, works as CSCW and CSCL do not provide concrete solutions. In (Sheng, 2005) the authors propose a UML based context metamodel for the development of context-aware mobile applications implemented on a Web services platform.

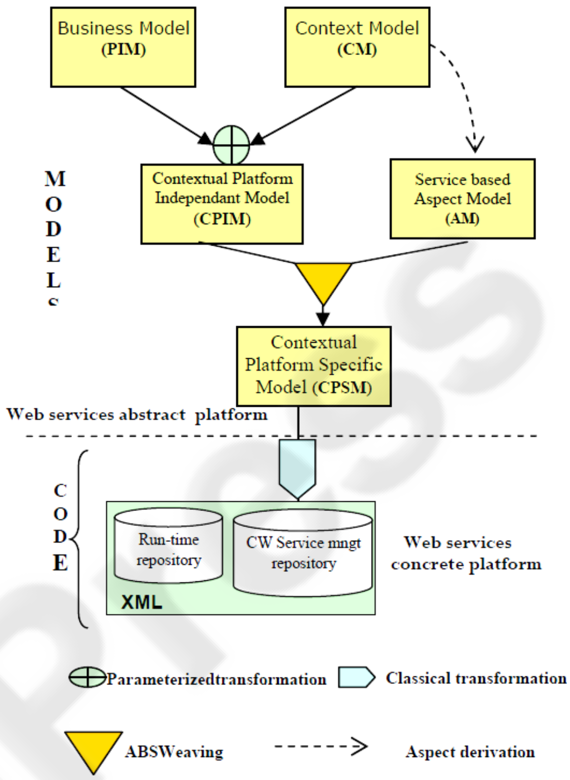

Figure 11: Models, context and aspect for adaptability.

The proposed metamodel does not refine contextual information and focuses on the association between basic contextual structures with service invocation interfaces for both contextual providers and context-aware applications. In (OU, 2006), they have applied MDA in context-aware application development. They focus on the development of context-awareness based on ontologies. However, neither a context metamodel is proposed nor are transformation techniques used.

In (DeFarias, 2007) authors investigate a number of context models described in literature and propose a context metamodel based on the main concepts and strengths found in these models. The metamodel is formally described using MOF and has been used as a basis for the development of context-aware applications and an associated service platform. All these research works aim to explore adaptability and

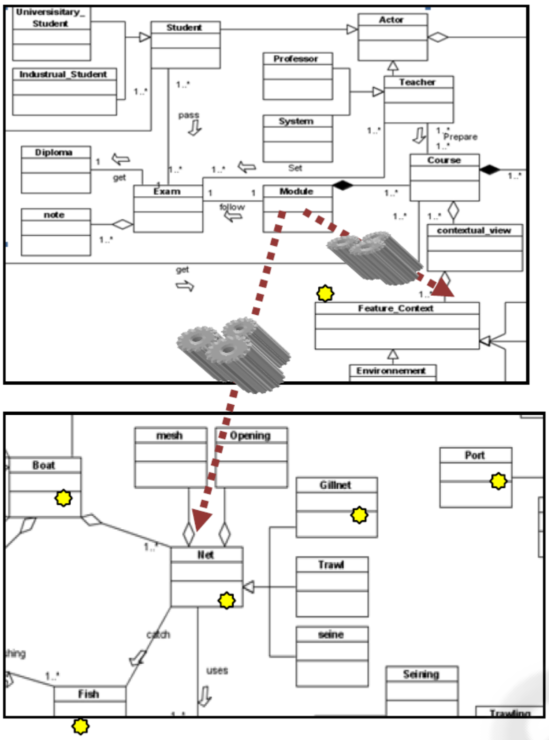

Figure 12: Example of Composition models.

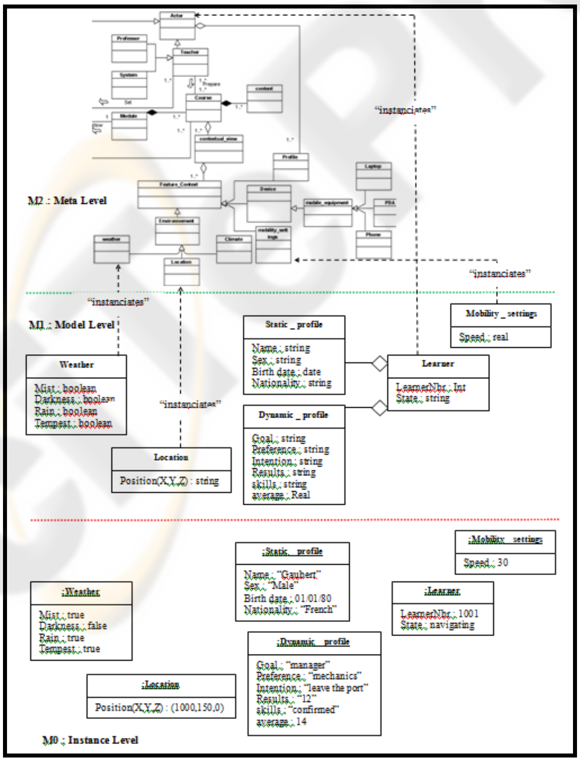

Figure 13: Instantiation of models.

flexibility on a service platform using context and models. But, neither of these works proposes an explicit approach to integrate context into business logic. For e-learning applications, some approaches aims to use metamodeling:

i) to define e-learning interoperable and platforms independent system
ii) and to extend standards as in (Boticario, 2007) (Dietz, 2004) (Davinia, 2004).

Some researchers introduce adaptability with Multi Agent System but we do not choose Artificial Intelligence based approach. Previous works as (Monfort, 2010), (Monfort, 2009), propose solutions to model context. We did not find any concrete and relevant related works concerning such an approach in e-Learning domain, but we are convinced our approach is pertinent because we got good results with fishing simulator and in other Web based application domains.

8 CONCLUSIONS

This paper proposes a metamodel approach to introduce (ambient) context awareness in LD model. It is based on our previous works about adaptability and models composition based MDD. We propose examples coming from a concrete industrial project. We aim:

i) to define an independent platform model based on services,
ii) to implement models transformations to link these models to implementation platform,
iii) to promote automatic code generation…

We have to present transformation rules via a technical platform based on services and supporting context awareness and to enrich our example with these rules.

Future works will contribute to propose other extra formalism to describe dynamicity (as sequence diagrams and Petri nets) and aspect based approach supporting context awareness.

REFERENCES

Boticario, J. Olga. C Santos: An open IMS-based user modeling approach for developing adaptive-learning managements systems. Retrieved from www-jime.open.ac.uk/2007/02/.

Chen, G., and Kotz, D. (2000). A survey of context-aware mobile computing research. Tech. rep., Dept. of Computer Science, Dartmouth College.

Davinia, H., Juan I. Asensio Perez, Yannis A. Dimitriadis, "IMS Learning Design Support for the Formalization of Collaborative-learning Patterns" Advanced Learning Technologies, IEEE International Conference on Advanced Learning Technologies (ICALT'04), 2004. , pp. 350-354.

De Farias, C. R. G., Pires, L. F., and van Sinderen, M. (2007). A MOF Metamodel for the Development of Context-Aware Mobile Applications. In Proceeding of the 22nd ACM Symposium on Applied Computing (SAC'07) pages: 947 - 952.

Dey, A. K. (2001). Understanding and Using Context. Personal and Ubiquitous Computing 5, 1, 4-7.

Dietze S., Alessio Gugliotta, and John Marko Rosić Vlado Glavinić Branko Žitko: Intelligent authoring shell based on Web services : http://www.pmfst.hr/~bzitko/radovi/files/INES2004.pdf

Frankel S. David. (2003). Model Driven Architecture: Applying MDA to Enterprise Computing, Wiley Publishing, Inc.

Klein, J, Hélouet, and J. M. Jézéquel. -- Semantic-based weaving of scenarios. -- In Proceedings of the 5th International Conference on Aspect-Oriented Software Development (AOSD'06), Bonn, Germany, March 2006.

Kiczales, G., Lamping, J., Maeda, C., and Lopes, C. (1997). Aspect-oriented programming. In Proceedings European Conference on Object-Oriented Programming, volume 1241, pages 220–242. Springer- Verlag, Berlin, Heidelberg, and New York.

Lundesgaard, S., Arnor Solberg, Jon Oldevik, Robert France, Jan Oyvind Aagedal, Frank Eliassen, Construction and Execution of Adaptable Applications Using an Aspect- Oriented and Model Driven Approach, IFIP DAIS 2007, LNCS 4531, 76-89, 2007

Matthias, B., Dustdar, S., and Rosenberg, F. (2007). A survey on context-aware systems. International Journal of ad Hoc and ubiquitous Computing 2 007.

Mary, B., and Patrick, B. (2005). Understanding context before to use it. In 5th International and Interdisciplinary Conference on Modeling and Using Context vol. 3554 of Lectures Notes in Artificial Intelligence, Springer-Verlag, pp. 29-40.

Monfort, V., Hammoudi, S. When Parameterized MDD Supports Aspect Based SOA , IJEBR 2010, International Journal of E-Business Research (To appear).

Monfort, V., Hammoudi, S. ICSOC, Towards Adaptable SOA: Model Driven Development, Context and Aspect The 7th International Conference on Service Oriented Computing, November 23-27 2009, Stockholm, Sweden.

OMG (Object Management Group). (2001). Model Driven Architecture (MDA), OMG document number ormsc/2001-07-01.

Ou S., Georgalas N., Azmoodeh M., Yang K. and Sun X., (2006). A Model Driven Integration Architecture for Ontology-Based Context Modelling and Context-Aware Application Development, A. Rensink and J. Warmer (Eds.) (2006). ECMDA-FA, LNCS 4066, pp.188–197. Springer-Verlag Berlin Heidelberg.
Pascoe, J. (1998). Adding Generic Contextual Capabilities to Wearable Computers. In 2nd International Symposium on Wearable Computers (Los Alamitos, California), Press, Ed., pp. 92-99.
Reese, G. Cloud Application Architectures: Building Applications and Infrastructure in the Cloud, O'Reilly Publisher (2009)
Sheng, Q. Z., and Benatallah, B. (2005). ContextUML: a UML-based Modeling Language for Model-Driven Development of Context-Aware Web Services. In Proceedings of the International Conference on Mobile Business (ICMB'05).
Schilit, B. N., and Theimer, M. (1994). Disseminating active map information to mobile hosts. IEEE Network 8 22-32.
Schneider, L. "LD" Retreived from http://tecfa.unige.ch/guides/tie/pdf/files/pedago-ld.pdf
Strang, T., and Linnhoff-Popien, C. (2004). A Context Modeling Survey. In First International Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp.
Tomaz, R. F., Hmida, M. B., and Monfort, V. (2006). Concrete solutions for web services adaptability using policies and aspects. The International Journal of Cooperative Information Systems (IJCIS), 15(3):415–438.
Vale, S. Hammoudi, S. (2008) Context-aware Model Driven Development by Parameterized Transformation. Proceedings of MDISIS 2008 Workshop, joint to CAISE'2008. pages 121-133
Weiser, M. (1991). The computer for the 21st century. Scientific American. PP 94-104.
Winograd, T. (2001). Architectures for context. Human-Computer Interaction (HCI) Journal 16.