



HAL
open science

Context Awareness and Model Driven Engineering: A Multi-Level Approach for the Development of Interactive Applications in Public Transportation

Arnaud Brossard, Mourad Abed, Christophe Kolski

► To cite this version:

Arnaud Brossard, Mourad Abed, Christophe Kolski. Context Awareness and Model Driven Engineering: A Multi-Level Approach for the Development of Interactive Applications in Public Transportation. 27th European Annual Conference on Human Decision-Making and Manual Control, EAM'08, Jun 2008, Delft, Netherlands. hal-03450648

HAL Id: hal-03450648

<https://uphf.hal.science/hal-03450648v1>

Submitted on 14 Feb 2024

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.

Context Awareness and Model Driven Engineering: A Multi-Level Approach for the Development of Interactive Applications in Public Transportation

Arnaud Brossard¹, Mourad Abed¹, Christophe Kolski¹

(1) LAMIH – UMR CNRS

Université de Valenciennes et du Hainaut-Cambrasis
Le Mont Houy, F-59313 Valenciennes Cedex 9, France

Phone: +33 (0)3 27 51 13 50

E-mail: {arnaud.brossard, mourad.abed, christophe.kolski}@univ-valenciennes.fr

Abstract

Model Driven Engineering (MDE) is becoming today the new paradigm in applications development. Concurrently to that, with the generalization of mobile computing, the notion of context awareness needs to be taken into account in more and more applications. The problem is that integrating context awareness in a MDE approach is not strait forward. In this paper, we will show our solution to integrate them by using a multilevel approach for the notion of context awareness and an adaptation of the classic Model Driven Architecture (MDA) through the use of multilevel frameworks.

Keywords: Context Awareness, Model Driven Engineering, Public Transportation.

1 Introduction

Today, and in particular in the field of public transportation, interactive applications should be multiform by being able to adapt themselves to different platforms but also to their context of use (Schilit *et al.*, 1994). This involves an increasing technical complexity which seek to manage the Model Driven Engineering (MDE) approaches (Favre *et al.*, 2006) (Schmidt, 2006). And if today, there are numerous researches in context awareness and MDE fields, very few establish explicit links between them. In this paper, we will show that, in a MDE approach, the context should not be taken into account only at the conceptual level but at all levels of the architecture.

2 Model Driven Engineering (MDE) and Context Awareness

2.1 Model Driven Engineering

The purpose of Model Driven Engineering (Schmidt, 2006) is to allow the automatic generation of applications by defining conceptual models and more specifically by assembling conceptual models. For doing that, MDE is based on a three levels MDA¹

¹ MDA: Model Driven Architecture

architecture (OMG, 2001) which allow to convert the conceptual models to the final application through a succession of models transformations (cf. figure 1).

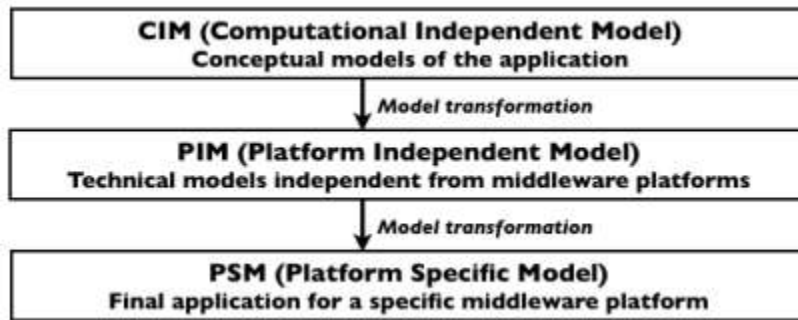


Figure 1. The Model Driven Architecture

In this paper, and by referring us to (Brossard *et al.*, 2007), we will use, by extension, the term of business process to characterize the conceptual models of the CIM level of the MDA architecture.

2.2 Context Awareness

The concept of context, in the field of interactive applications, was defined by (Dey and Abowd, 2000) as following: "*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.*" Using similar definitions, some other works have tried to characterize the context (Schilit *et al.*, 1994) (Gross and Specht, 2001). For this paper, we have retained those of (Thevenin and Coutaz, 1999) which propose a vision of the context using three dimensions: user, environment and platform.

3 Context adaptation and MDE: a multi-level approach

3.1 The characterization of the three dimensions of context

When developing interactive applications, each dimension of the context can be characterized by several elements. Thus, the user could be characterized by:

- Explicit properties (name, address, job, ...).
- Interests and preferences which are used to specify his/her tastes and goals (for instance: "I rather would use the bus than the subway", ...).
- Information about his/her mind state, emotional states (traveler in a hurry, sleeping traveler, ...).
- ...

The environment could be characterized by:

- Time (for instance: morning, evening, public holiday, ...)
- Geographical space (for instance: the type of space like a station or a store, ...).
- Others stimulus linked to the physical environment (luminosity, noise, ...).
- ...

The technical platform could be characterized by:

- Specific technical elements (all the interaction possibilities, screen size, ...).

- A platform level type, which allow to aggregate different platforms inside units having a certain number of characteristics in common from the interactions point of view.
- ...

This multiform characterization of the three dimensions, which is far from being exhaustive and this paper, involves a very significant number of possibilities of different contextual adaptations; quite apart from the fact that the various elements characterizing each dimension could interact with each other. It is all this complexity and all the possibilities that need to be managed inside a MDA type architecture, which is the foundation of any MDE approach.

3.2 Context and MDA architecture

In a classical MDE approach, the whole application must be defined at the conceptual level and each contextual adaptation should be defined there. However, this is not always coherent. Thus, if one takes the automatic adaptation of the platform to the luminosity, this one is dependent on the technical possibilities of the platform like its capacities to collect the external luminosity and to regulate the luminosity by programming. That is why it is not relevant to define this kind of adaptation in the conceptual level, all the more it will be necessary to take into account this adaptation in all the applications. In fact, it will be more efficient to let the platform do this adaptation automatically and thus to integrate it to the PSM level. In table 1, we present our vision of the management of context elements in a MDA architecture.

Table 1. Management of context elements in a MDA architecture

MDA Level	Examples of Context Elements
CIM	<ul style="list-style-type: none"> - Conceptual characterization of the context elements (for instance: definition of geographical spaces like stations or cities, definition of the platforms levels, ...) - Use of the characterization in the business processes (Example: a buying business process of transport document differentiated according to whether the user is in a railway station or in a subway station)
PIM	<ul style="list-style-type: none"> - Choice of different generic technical solutions according to the context (Example: technical solution of purchase validation different in a station than in a train)
PSM	<ul style="list-style-type: none"> - Use of local information to automatically adapt the application to the context of use without calling a specific business process (Examples: automatic choice of a screen presentation type like Windows XP style or Mac OS style, automatic adaptation of the platform to the luminosity, ...). - Definition and characterization of the sensors and the actuators associated with the contextual adaptation: What can the sensor measure? How? Which are the different valid values? And, if one can not make a measure how is the context element taken into account (default value, users choice, specific process associated to the platform, ...)?

4 Case study: Contextual adaptation of a traveler's information system in public transportation

4.1 General introduction to our approach:

To be able to use the multilevel approach of context awareness as shown in 3.2, we propose to use an adaptation of the classic MDA architecture as presented in the figure 2. It is important to point out that our architecture is, for the moment, only able to manage classical interactive applications of WIMP (Window, Icon, Menu, Pointing device) type.

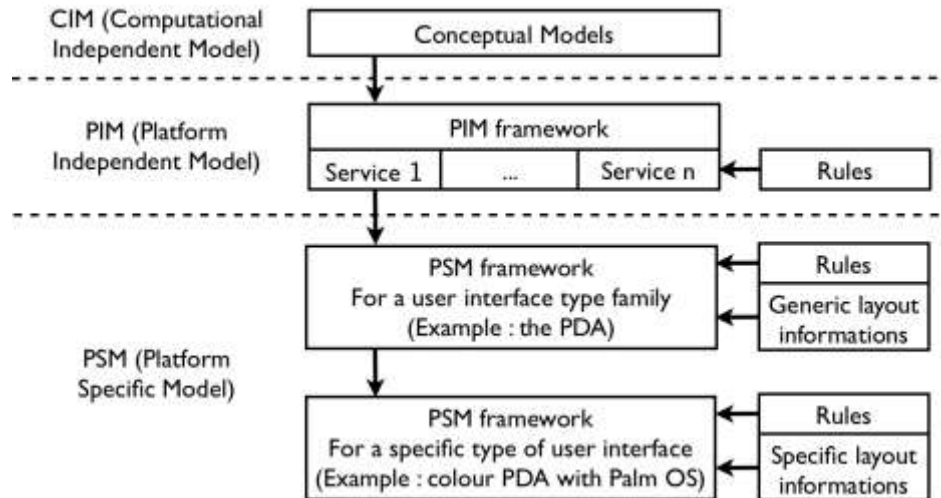


Figure 2. Our proposal for the MDA approach adaptation

In our proposal, which is described in details in (Brossard *et al.*, 2007), the conceptual models are defined using different specific models, which allow to modelize the application and its environment. In these models, the main model is the Business Process Model, which uses the other models to define, from the business point of view, the processes associated to the application. These conceptual models are designed relying on a whole set of functional services defined at the CIM level which will be presented in details in 4.2. These services are developed within a parameterized framework using rules to allow a quick adaptation of the behavior of each service. The transformation from the level PIM to the level PSM uses a two levels framework. The first one allow to take into account all the specificities common to all the platforms from a specific family like the Personal Digital Assistant (PDA) family. The second one, specific to the final platform, such as for example a PDA color under Palm OS, allow to adapt the application to the specificities of each final platform. The PSM frameworks are used to translate the PIM services to PSM services and to transform the conceptual interactive elements into concrete interactive elements. Thus, while passing by PIM framework, an element of the conceptual type text field, for example, could be translated into a dropdown list or a field text according to the rules associated with each framework at the PSM level. It is also at the PSM level that we introduce information relating to the physical positioning of the various graphical elements and to their presentation (color, form, an so on). As for the PIM level, the rules defined at the PSM level are used to allow a quick adaptation of the PSM framework.

4.2 First example: Context awareness at the CIM level

In this example, we will show how the context of use of a business process could be taken into account in our approach. For defining the business processes, we use a specific notation directly derived from the Business Process Modeling Notation (BPMI, 2004). The business process used in this example is quite simple: it is the process associated to the purchase of a train ticket (cf. figure 3). This main process contains three sub-process: the first one allows the user to search for and select a trip, the second one is associated to the payment validation and the third one is linked to the payment confirmation.

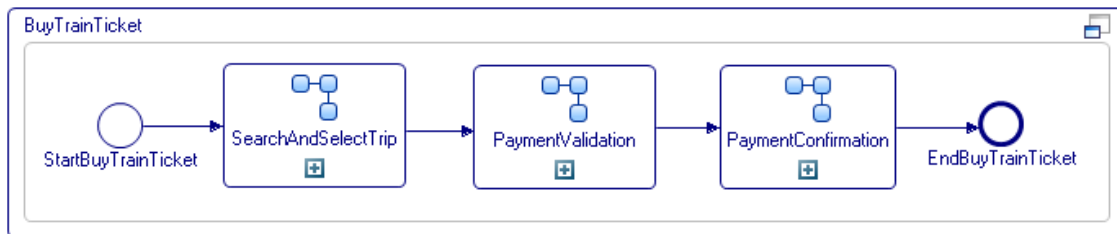


Figure 3. The Business Process associated to train ticket buying

In our example, the main process is the same for all users whatever the place where the user is (at home, at the station, in the train, ...). As the user still need a concrete ticket to take the train, he always needs to print is ticket in a Self Service Terminal at the railway station but the payment confirmation sub-process is not the same depending on the context of use of the process. Thus, if the user buys his train ticket directly at the Self Service Terminal at the railway station, he/she receives the payment confirmation on a small ticket otherwise he receives an email (cf. figure 5) and should go to a Self Service Terminal to pick up is ticket. For being able to do that, the payment confirmation sub-process should be able to take into account the context of use and more specifically it should be able to know if the user uses it on a Self Service Terminal or on another place. For doing that we should define into the sub-process a selection step which allows us to evaluate the context (cf. figure 4). This is done by the evaluation of a specific rule "UseSelfServiceTerminal" which could return only two values: "Yes" if the process is used on a Self Service Terminal and "No" otherwise. If the result is "Yes", the Self Service Terminal prints the ticket and a payment receipt. If the result is "No", an email is sent to the user.

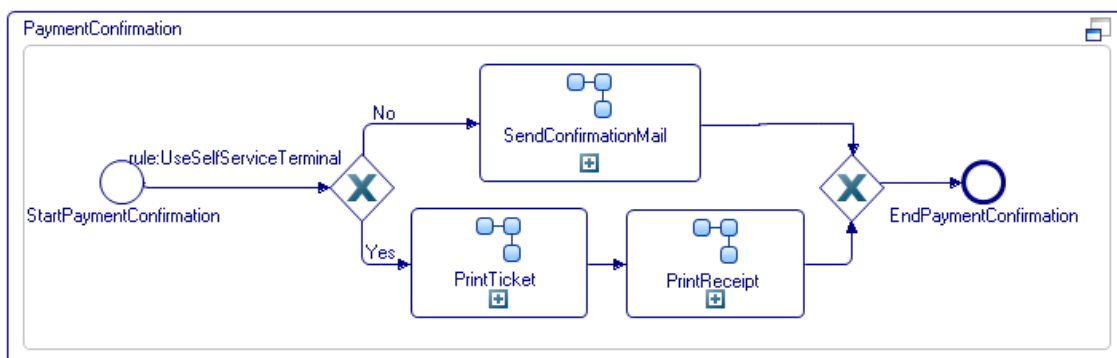


Figure 4. The contextual adaptation of the payment confirmation process

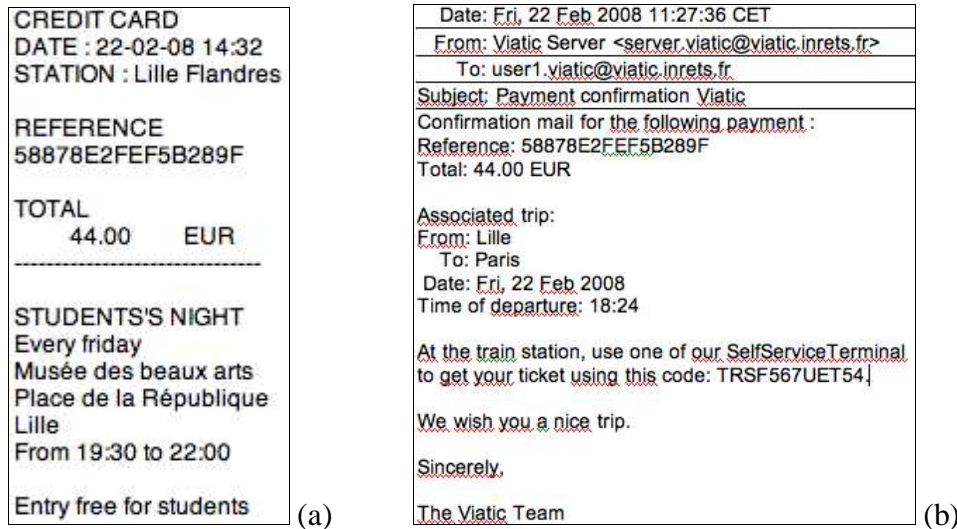


Figure 5. Payment confirmation ticket(a) and email(b) associated to the business process

4.3 Second example: Context awareness at the PIM level

At the PIM level, the notion of context awareness is taken into account through the PIM Framework which contains all the technical and functional services that could be used explicitly or implicitly at the CIM level. An implicit service is a service used by the application which is not taken into account in the business processes defined at the CIM level. For example, the session management is an implicit service: all applications use it to manage the user session and it does not have to be defined explicitly at the CIM level to be used. An explicit service is a service which could be used in a CIM model but does not have to be defined at the CIM level. For example, the personalization of information returned by user requests has not to be defined at the CIM level. The only thing which it is important to define at the CIM level is the kind of personalization (by country, by user preferences, ...) and not the personalization process itself. As at the PIM level all the services are defined through the PIM framework, the notion of context awareness could be taken into account at two places. At first, inside the framework and more exactly inside the service itself where we could use context information to define different behaviors for a same service. Or, through the definition of rules where we could define the use of different sub-services depending of the context of use. In our example, we will show this second option through the definition of two different music request services regarding the user localization: inside a vehicle or outside a vehicle. If the user is outside a vehicle, each request will go directly to the application server (cf. figure 6) and the user will use a service called "DirectMusicRequest". If the user is inside the vehicle, the request will go the embedded server and if anything is found, the request will be transmitted to the application server (cf. figure 7) and the application will use a service called "EmbeddedMusicRequest".

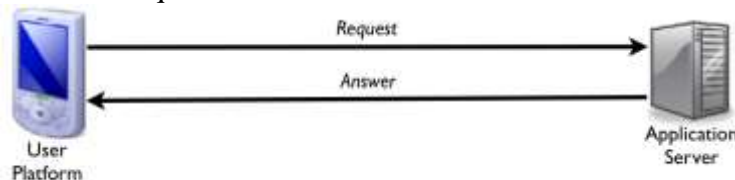


Figure 6. The standard technical architecture (service DirectMusicRequest)

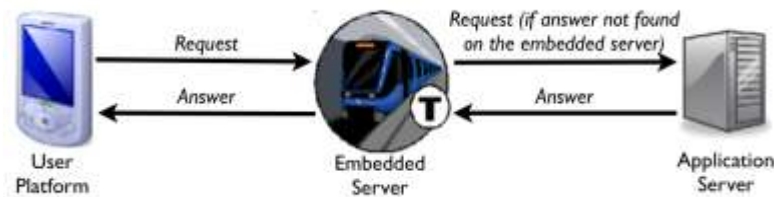


Figure 7. The technical architecture in a vehicle (service EmbeddedMusicRequest)

4.4 Third example: Context awareness at the PSM level

Today some mobile devices contain sensors, which allow them to catch some information about their environment (sound, luminosity, geographical localization through GPS). Some contextual adaptations could be done automatically by the platform like, for example, the automatic adaptation to the external luminosity on the notebook "MacBook Pro" from Apple Inc. Otherwise, the information given by the sensors could be used in the PSM framework to do some automatic adaptation. For example, if the user uses an application which guide him/her to his/her next connection inside a railway station, we could define, if we have a speed rate sensor on the user platform, inside the PSM framework that the font style used for the current location will not be the same if the user is moving slowly or if he/she is moving fast or even running (cf. figure 8).

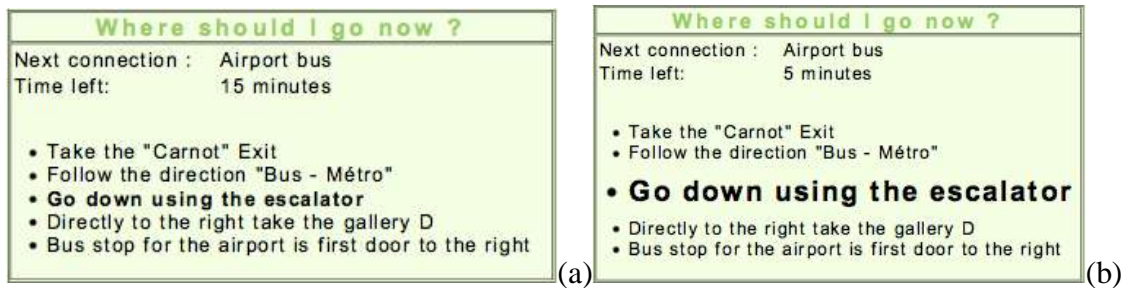


Figure 8. Adaptation to the user moving speed: slowly moving (a) or fast moving (b)

In this example, the choice made of managing the context at the PSM level is relevant if the information about the user speed could be only available for one or two platforms and if this information will not be used as a criteria at the CIM level. In fact, for each contextual information, an analysis have to be done to define the level where the information about the context will be managed. As the results of these analysis are linked to the business domain and more explicitly to the company for whom the application is programmed, there is no good and bad choice as far as the contextual information is not managed concurrently at two different levels. It is probably one of the key points of our approach but one of its drawback as it will not be easy to manage the evolution in time of each contextual information: management at the PSM level and then, after a while, at the CIM level. One of the solutions to avoid this drawback could be to manage all the contextual information at the CIM level but it will introduce more complexity in the models and less flexibility without allowing to manage some technical adaptations like those presented in 4.3. In fact, the choice of the management level is a balance between all this constraints and limitations and, for us, it is itself a very interesting research area.

5 Conclusion

The future of interactive applications, in particular in the field of public transportation, will need, most probably, a closer integration between MDE approaches and contextual adaptations. As we have seen in this paper, this integration should not be made only for conceptual models but for all the levels of the MDA architecture. In fact, it imposes an adaptation of the architecture through an enrichment of the PIM and PSM levels. But, it implies also a new grid for the analysis of contextual elements in order to split up their adaptation in the three MDA levels. It is certainly the most important point to succeed, in the future, the perfect integration between MDE approaches and contextual adaptation.

6 Acknowledgment

The present research work has been partially supported by the "Ministère de l'Education Nationale, de la Recherche et de la Technologie", the "Région Nord Pas-de-Calais" and the FEDER (projects MIAOU, EUCUE, SART), the ANR ADEME (Viatic.Mobilité). The authors gratefully acknowledge the support of these institutions.

References

- Brossard A., Abed M. and Kolski C. (2007): Modélisation conceptuelle des IHM : Une approche globale s'appuyant sur les processus métier. *Ingénierie des Systèmes d'Information - Revue des sciences et technologies de l'information*, 2007/5, 69-108.
- BPMI (2004): Business Process Modeling Notation version 1.0, <http://www.bpmn.org>
- Dey A., Abowd G., (2000) Towards a Better Understanding of Context and Context-Awareness. *In the workshop on the What, Where, When and How of Context-Awareness, 2000 Conference on Human Factors in Computing Systems (CHI 2000)*, The Hague, The Netherlands.
- Favre J.-M., Estublier J. and Blay-Fornarino M. (2006): *L'ingénierie dirigée par les modèles*, Hermès-Lavoisier, Paris.
- Gross T. and Specht M. (2001): Awareness in Context-Aware Information Systems. In *Proceeding of Mensch & Computer 01*, Bonn, Germany, 173-181.
- OMG (2003): MDA Guide v1.0.1.
- Schilit B., Adams N. and Want R. (1994): Context Aware Computing Applications. In *Proceedings of the first International Workshop on Mobile Computing Systems and Application*, Dublin, Ireland, 85-90.
- Schmidt D. C. (2006): Model-Driven Engineering, *Computer*, vol. 39, n° 2, 25-31.
- Thevenin D. and Coutaz J. (1999): Plasticity of User Interfaces: Framework and Research Agenda. In *Proceedings of IFIP 13th International Conference on Human-Computer Interaction, Interact'99*, Edinburgh, United Kingdom, 110-117.
- Viatic Mobilité, <http://viatic.inrets.fr>