XML in Formal Specification, Verification and Generation of Mobile HCI

Ines Riahi, Meriem Riahi, and Faouzi Moussa

Laboratoire Cristal, Ecole Nationale des Sciences de l'Informatique de Tunis, rue des Entrepreneurs Charguia II, 2035 Tunis-Carthage, Tunisia
ines.riahi@yahoo.fr, Meriem.Riahi@insat.rnu.tn, faouzimoussa@gmail.com

Abstract. Our work is carried out in the framework of a global approach for Human-Computer Interaction design and automatic generation. This approach is based on a formal modeling of the Human-Computer Interaction. We propose a Model Based Design Approach (MBDA). We are concerned with identifying the user tasks and requirements and further with the automatic graphical interface validation and generation. Therefore we use Petri Nets. Indeed, the Petri Nets are very efficient in formal modeling of HCI. Our research focuses on mobile HCI. It aims to analyze the ubiquitous environment using ontology described in OWL2 standard. We face difficulties in modeling ontology in XML using Petri Nets. Thus, it becomes necessary to adopt approaches for manipulation of Petri nets via XML as PNML or XML Nets.

Keywords: Human-Computer Interaction (HCI), Model Based Design Approach (MBDA), Ubiquitous environment, Petri Nets, Ontology, OWL2, XML, XML Nets, PNML.

1 Introduction

Our research field is Human-Computer Interaction (HCI) and it concerns especially mobile HCI. We intend to elaborate a global approach for formal specification and generation of mobile interface. In fact, ubiquitous environment is characterized by an important rate of information received from different sources. To represent this knowledge, ontology- described in the standard OWL2 (Ontology Web Language V2) based on XML (Extensible Markup Language) - is used. For modeling the Human-Computer Interaction, we use the Petri Nets. We face difficulties in modeling the ontology using Petri Nets and how to express the interaction between an ontology represented by OWL2 and Petri nets formalized by XML. Thus, it becomes necessary to adopt approaches for manipulating Petri Nets via XML. PNML and XML Nets are proposed to deal with this problem.

Our paper focuses on the use of XML in formal specification (with Petri Nets), verification and generation of mobile graphical interfaces. The first part of this paper presents a global approach for formal specification and generation of mobile graphical interfaces. Then, a bibliographic review of the methods XML Nets and PNML is presented and discussed. Finally, we propose our approach based on the profit of using both XML and Petri Nets.

J.A. Jacko (Ed.): Human-Computer Interaction, Part III, HCII 2011, LNCS 6763, pp. 92–100, 2011.

2 Approach for Formal Specification, Verification and Generation of Mobile HCI

The effervescence of new technologies and means of mobile communication in recent years has inspired developers to bring mobile devices into their applications giving rise to ubiquitous computing. This new environment integrates mobile devices with different hardware and software capabilities.

The adaptation of human-computer interfaces to an ubiquitous environment consists in producing a number of interfaces that can be dynamically adapted to different types of mobile devices while respecting these ergonomic properties. Most of the works done on adapting mobile HCI are based on models describing the interaction between human and machine. [1]

The mobile HCI for PDA, mobile phones or tablets are characterized mainly by small screens, usually limited treatment capacity, backup information and the availability of random network [2]. All these constraints must be taken into account when modeling the HCI Mobile.

Until now, most of the works has focused mainly on technological aspects of mobile terminals or on the problems of assessing the usability of mobile devices without considering a crucial point which is modeling mobile HCI. These devices must continue to be functional in any environment in which they are used. This requires the definition and the integration of the concept of mobility in a model based approach.

In this context, we propose an approach that is made up of five steps as shown in figure 1:

1. The first step consists in analyzing the whole Human-Machine System in terms of the system (in different contexts and environments) [3], the interaction and the user's tasks. The modeling of the System's behavior becomes possible. It expresses the interaction of the User with the Graphical Interface. This analysis is carried out using Petri Nets modeled in XML. This is discussed further, in section 4.
2. The second step is the achievement of the deduction of the user requirements. In fact, the ubiquitous environment analysis and its modeling provide the set of user requirements in accordance with each functional context.
3. The third step ensures the identification of the interface objects according to the user requirements. Once the interface objects are identified, this step consists in specifying the interface in terms of displays, graphical objects and dialogue.
4. In the next step, we take advantage of the formal technique used for interface specification to verify critical properties of the generated specifications. Indeed, formal techniques are specially recommended because they allow the designers to proceed to the validation (even mathematical) of the UI (User Interface) before going on to their actual creation and implementation.
5. The last step in this approach is devoted to the automatic interface generation.

We focus, in this paper, on the first step: Human-Machine System Analysis in an ubiquitous environment. First, we study the environment to detect the different characteristics and functional contexts. Then, we identify, for each context, the user tasks. The task analysis is, afterward, carried out.

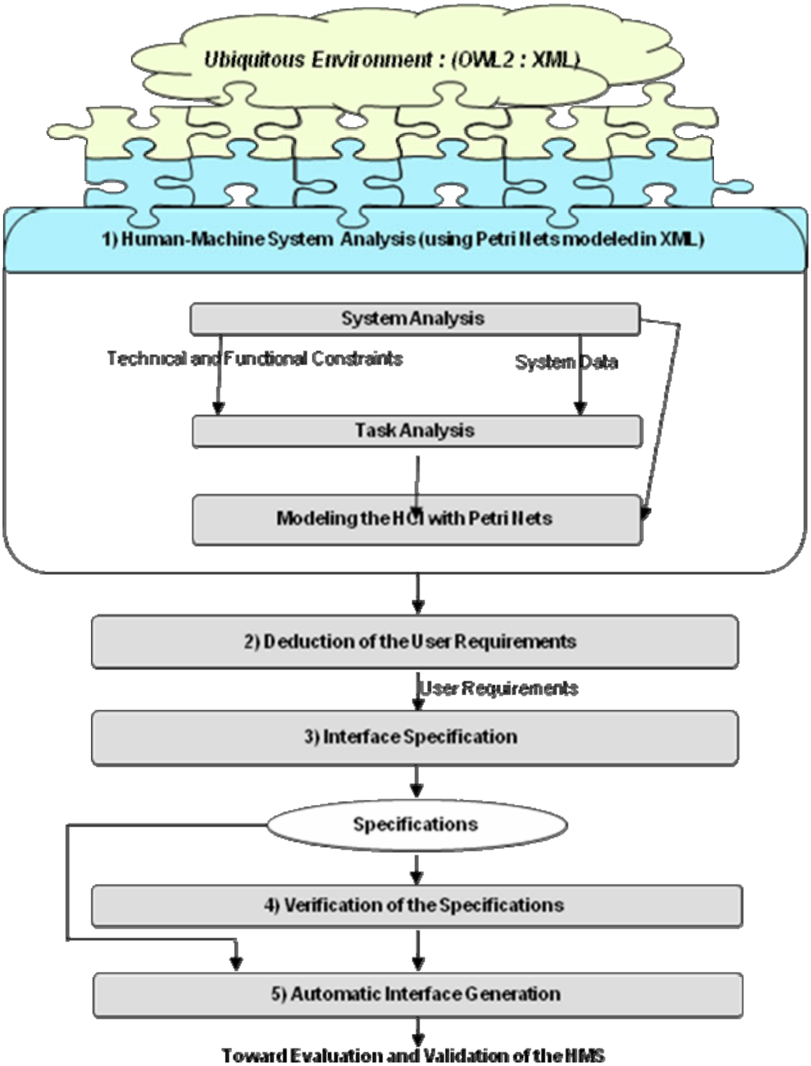


Fig. 1. Proposed Approach for HCI Automatic Generation [4]

This analysis allows the definition of the operator's actions according to different contexts. The parameters of these actions will constitute, therefore, the user requirements. Thus, this analysis will provide a document containing the different variables necessary for the mobile HCI modeling. This document is described in OWL2.

Once this is done, we proceed to the modeling of the Human-Computer Interaction in terms of operator's actions. This modeling takes into account the different contexts and evolution of the environment. This modeling must be performed with a formal technique. This allows validating certain properties of the interface as stability and absence of blocking and conflict. Petri nets are proposed for that. The difficulty, here, is how to combine Petri nets with XML documents obtained by the description in OWL2.

The section below presents a bibliographic study about these formalisms.

3 XML and Petri Nets

3.1 XML NETS

XML nets are a new variant of high-level Petri nets. They are a formal graphical modeling language which models the flow of XML documents. The choice of Petri nets was obvious. Petri nets have the advantages of the processes' graphical representation with a formal semantic modeling of the behavior. XML nets are based on GXSL, a graphical XML schema definition language, and the corresponding XML document manipulation language XManiLa.

Graphical XML schema definition language (GXSL): is a graphical language for XML schema definition for the design of XML document types. It consists of a set of markup declarations of different types: element, attribute list, entity or notation declaration. GXSL is based on DTD (Document Type Definition) that can be derived from XML schema. Instead of creating a completely new graphical modeling language for XML document types, we rely on well known data modeling concepts such as the Unified Modeling Language (UML) [5].

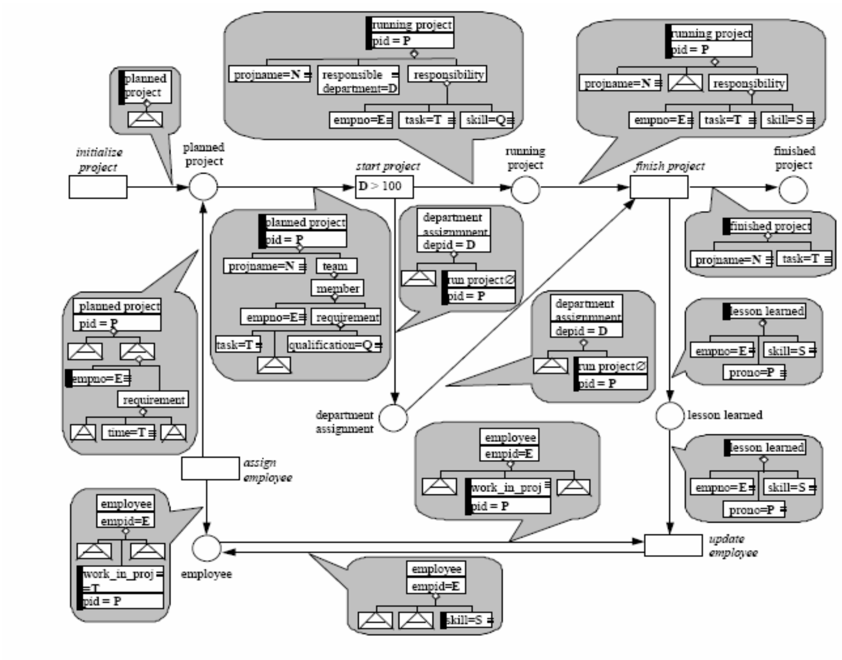


Fig. 2. Example of XML Nets [5]

The presented version in [5] of GXSL is DTD-based, i.e. a DTD can be unambiguously derived from the XML schema diagram XSD. The full GXSL provides concepts for graphically modeling data type-based XML schemas with enhanced features like an additional kind of element aggregation, element substitution, element and association references, and finally the concept of keys and foreign keys.

XML document manipulation language (XManiLa): XManiLa is the GXSL based XML document manipulation language, which can use GXSL with some extensions also for querying and manipulating XML documents. In order to enable also a content based query, we allow for assigning constants or variables to an element or an attribute. XManiLa is not only suited for document retrieval, but also for insertion and deletion (and thereby also for updating). These operations either concern a whole document or elements on lower hierarchy levels. [5]

The combination of GXSL with XMANILA gives XML Nets as shown in figure 2. The static components of XML nets (i.e. the places of the Petri net) are typed by XML schema diagrams, each of them representing a DTD. Places can be interpreted as containers for XML documents which are valid for the corresponding DTD. The flow of XML documents is defined by the occurrences of transitions.

3.2 Petri Net Markup Languages (PNML)

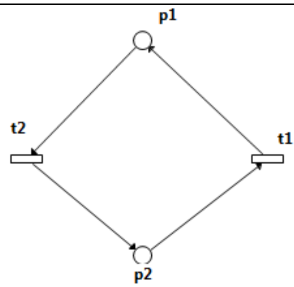
The Petri Net Markup Language (PNML) is an XML-based interchange format for Petri nets. The design of PNML was governed by the following principles:

- Readability: The format must be readable and editable with any text editor,
- Universality: The format should be able to represent any kind of Petri nets with its possible extensions,
- Mutuality: The format should allow us to extract as much information as possible from a Petri net. Therefore, the format must extract the common principles and the common notations of Petri nets [6].

The use of XML guarantees the readability of the format. Universality can be guaranteed by labeling net objects and the net itself. The legal labels, their possible values, and the possible combination of values are defined by a Petri Net Type Definition (PNTD). Mutuality can be guaranteed by conventions, which are a set of standardized labels [6].

The main idea of PNML is that any kind of Petri net can be considered to be a labeled graph. In particular, all information that is specific to a particular kind of Petri net can be captured in labels.

A file that meets the requirements of PNML is called a Petri net file; it may contain several Petri nets. Each Petri net consists of objects, which, basically, represent the graph structure of the Petri net. Each object within a Petri net file has a unique identifier, which can be used to refer to this object. In basic PNML, an object is a place, a transition or an arc. For convenience, a place or a transition is called a node. In order to assign further meaning to an object, each object may have labels. Typically, a label represents the name of a node, the initial marking of a place, the guard of a transition, or the inscription of an arc. Two kinds of labels are distinguished: annotations and attributes. An annotation comprises information that is typically displayed as text near the corresponding object. In contrast, an attribute specifies a graphical property of an object such as color, style, form, or line thickness. Each object and each annotation is equipped with graphical information. For some tools, it might be necessary to store tool specific information, which is not supposed to be used by other tools. In order to store this information, each object and each label may be equipped with such tool specific information. Its format depends on the tool and is not specified by PNML [7].




```
<?xml version='1.0' encoding='UTF-8'?> <model ….
<nodes> <node nodetype='place' …
<attributes><attribute name='name' xposition='296'
yposition='218'>p2</attribute>          <attribute
name='marking' xposition='0'
yposition='0'>0</attribute></attributes>    </node> <node
nodetype='place' … <attributes><attribute name='name'
xposition='315' yposition='19'>p1</attribute><attribute
name='marking' xposition='0'
yposition='0'>0</attribute></attributes> </node> <node
nodetype='transition' … <attributes> <attribute
name='name' xposition='201'
yposition='106'>t2</attribute></attributes> </node> <node
nodetype='transition' … <attributes> <attribute
name='name' xposition='406'
yposition='114'>t1</attribute> </attributes> </node>
</nodes> <arcs> <arc arctype='arc' id='7' startid='6'
endid='4'> </arc> <arc arctype='arc' id='8' startid='4'
endid='5'> </arc> <arc arctype='arc' id='9' startid='5'
endid='3'> </arc> … </arcs> </model>
```


Fig. 3. Example of PNML

In order to guarantee compatibility among different Petri net types, the Conventions Document comprises the definitions of all standard labels; technically, the Conventions Document consists of a sequence of more or less independent label definitions. Each label definition in the Conventions Document must be assigned a unique name. For each label, the Conventions Document gives a reference to its meaning and states in which PNML elements it may occur. The conventions Document guarantee that the same labels have the same meanings in all types of Petri nets which allow the exchange between them [8]. The above example (figure 3) shows the description of a Petri nets with two places and two transitions.

3.3 Discussion

As a variant of high-level Petri nets as shown in figure 2, XML nets have formal semantics, graphical nature, and the strength in exchanging XML-based structured data. They are very suitable for Web Service Composition WSC because messages can be modeled and manipulated as place tokens for message passing, and the labels in arcs can be used to model constraints for WS discovery and selection. Using XML nets for WSC can thus improve WSC models and increase their dynamics [9]. In addition, XML Nets support inter-organizational business processes [10] and the supply chain management [11].

XML Nets have additional advantages in the description of process objects and inter-organizational exchange of standardized structured data (e.g. XML documents) [10]. However, XML Nets cannot verify the validity of Petri Nets. We may, also, get complex XML schemas for a complex system.

The Petri Net Markup Language (PNML) is an XML-based interchange format for Petri nets. In order to support different versions of Petri nets and, in particular, future versions of Petri nets, PNML allows the definition of Petri net types. Due to this flexibility, PNML is a starting point for a standard interchange format for Petri nets.

Another important issue for interchange formats is the size of real world systems. Typically, real world systems are too large to be models. PNML Modular, extension of PNML, introduces the concept of modules. Thus, a system can be built recursively from module instances (sub-Petri Nets) [6].

However, this interchange format does not guarantee the accuracy and validity of Petri nets obtained on sub-Petri Nets used recursively [12]. In addition, neither XML Nets nor PNML Nets have been used in the field of Human-Computer Interaction specification.

4 Proposed Approach

We are interested in our research work, in mobile HCI. Ontology is proposed to represent and formulate the knowledge of an ubiquitous environments. It is a set of structured concepts and relationships modeling the knowledge and providing a sense to each information [13]. OWL2 is proposed for this fact. Indeed, this web ontology language is based on a dialect of XML.

So we use XML as a mediator language allowing the analysis and the modeling of the ontology. Our contribution is to elaborate a model of a HCI based on Petri Nets using XML documents. XML guarantees the exchange of the ubiquitous environment and the mobile HCI specification as shown in figure 4.

In fact, Petri nets will specify the behavior and the treatment that will undergo the data extracted from the ontology. This data will "feed" the Petri nets gradually along the evolution of the environment.

The first step of this approach consists in translating Petri nets in XML documents. For each transition, we associate the useful concepts of the ontology. These concepts written in an XML document are used to execute the adequate tasks.

Thus, the result will be an XML document allowing the deduction of the user requirements and the generation of the user interface.

This approach can be appropriately applied to the example the car speed controlling.

This application is intended to control the speed of a driver according to her environment. This management is produced through a mobile HCI introduced in a mobile device.

Whatever its nature: PDA, GPS or a mobile phone, its role is to capture the nature of the environment and adjust the speed of the driver according to these data. These are defined using ontology written in OWL2 as we have specified in the previous section.

Then the Human-Machine Interaction while the operator is performing these tasks will be adapted using one of the two approaches presented in the third section.

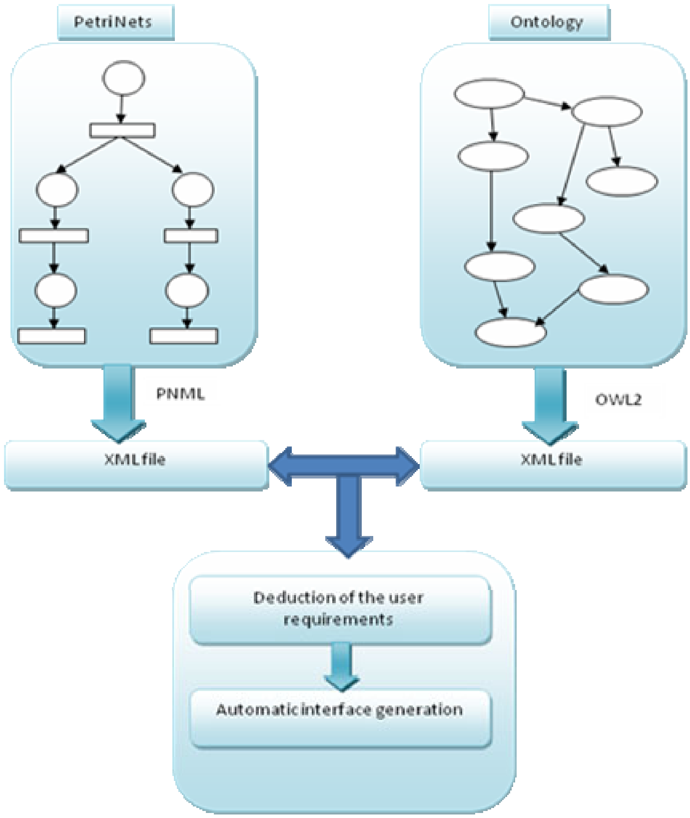


Fig. 4. Proposed approach

OWL data will feed into our Petri nets with XML based on the treatment chosen by the driver. Thus, from a technical point of view, we have two XML files. One representing our ontology which contains the description of our environment and the other contains the modeling of the Human-Computer System. Further, we will present an algorithm that ensures the interaction between these two XML files.

The output of this algorithm is an XML file that will allow the automatic generation of mobile HCI. This file will take into account the characteristics of mobile HCI contained in the first section.

The analysis and the modeling of ubiquitous environments are difficult and complex.

PNML Modular, extension of PNML, presented in section 3, offers the possibility to reduce the complexity by dividing a complex system in a set of simpler modules. These modules can be, afterward, used in a recursive manner.

In our previous works, we have focused on "how to make composition" while modeling the HCI with Petri Nets. According to these composition rules, we guarantee the validity of the Petri Net's critical properties. With verified properties, it becomes possible to validate the user interface. We combined these concepts with XML notation in order to propose a new extension of the standard PNML. These works will be presented in the near future.

5 Conclusion

In this paper we use the Petri nets with XML to model the system, the user's task and the interaction. A user's task is composed of a well-organized set of elementary actions. A global model is then constructed integrating the different evolutions of the functioning system's states and the associated user's tasks. The purpose of this work is to be able to deal with ontology while using a Petri Nets based modeling approach. This allows us to better link the two steps (i); the description of the ubiquitous environment using ontology (OWL2), (ii) and the formal modeling of the mobile Human-Computer Interaction. Further, we will take advantage of the formal technique used to verify critical properties of the generated specifications of the user interface.

References

1. Chaari, T.: Adaptation d'applications pervasives dans des environnements multi-contextes. Thèse de doctorat, L'institut national des sciences appliquées de Lyon (2007)
2. Calvet-Inglad, G.: Etude empirique de l'interaction multimodale mobile en situation naturelle. Thèse de doctorat, Université de Toulouse (2008)
3. Riahi, M.: Contribution à l'élaboration d'une méthodologie de spécification, de vérification et de génération semi-automatique d'interfaces homme-machine: Application à l'outil Ergo-Conceptor+. Thèse de doctorat, Université de Valenciennes et du Hainaut-Cambrésis (2004)
4. Moussa, F.: Vers une méthodologie globale de conception et de génération semi- automatique des IHM pour les systèmes industriels. Habilitation Universitaire en Informatique, Faculté des Sciences de Tunis (2005)
5. Lenz, K., Oberweis, A.: Inter-Organizational Business Process Management with XML Nets. In: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.) Petri Net Technology for Communication-Based Systems. LNCS, vol. 2472, pp. 243–263. Springer, Heidelberg (2003)
6. Weber, M., Kindler, E.: The Petri Net Markup Language. In: Ehrig, H., Reisig, W., Rozenberg, G., Weber, H. (eds.) Petri Net Technology for Communication-Based Systems. LNCS, vol. 2472, pp. 124–144. Springer, Heidelberg (2003)
7. Hillah, L.M., Kindler, E., Kordon, F., Petrucci, L., Trèves, N.: A primer on the Petri Net Markup Language and ISO/IEC 15909-2. Petri Net Newsletter (2009)
8. Billington, J., Christensen, S., van Hee, K.M., Kindler, E., Kummer, O., Petrucci, L., Post, R., Stehno, C., Weber, M.: The Petri Net Markup Language: Concepts, Technology, and Tools. In: van der Aalst, W.M.P., Best, E. (eds.) ICATPN 2003. LNCS, vol. 2679, pp. 483–505. Springer, Heidelberg (2003)
9. Che, H., Li, Y., Oberweis, A., Stucky, W.: Web Service Composition Based on XML Nets. In: Proceedings of the 42nd Hawaii International Conference on System Sciences (2009)
10. Che, H., Mevius, M., Ju, Y., Stucky, W., Trunko, R.: A Method for Inter-organizational Business Process Management. In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China (August 18-21, 2007)
11. Che, H., Stucky, W., Ju, Y.: Using XML Nets and Grid Services to support SCOR. In: Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming (July 12-15, 2008)
12. Vidal, J., Lama, M., Bugarin, A.: A High-level Petri Net Ontology Compatible with PNML. Petri Net Newsletter (2006)
13. http://www.technoscience.net/?onglet=glossaire&definition=324