A Model-Driven Approach for Building Ubiquitous Applications

Philippe Le Parc, Amara Touil, Jean Vareille

A Model-driven Approach for Building Ubiquitous Applications

Philippe Le Parc, Amara Touil, Jean Vareille
Université Européenne de Bretagne, France
Université de Brest
EA3883 LISyC - Laboratoire d'Informatique des Systèmes Complexes
20 av. Victor Le Gorgeu, BP 809
29285 Brest Cedex - France
{philippe.le-parc, amara.touil, jean.vareille}@univ-brest.fr

***Abstract*—This paper presents an on-going work, which aims to build a generic framework for the conception and the development of ubiquitous applications. This work is based on a Model Driven Engineering approach and pays attention to possible failures and to the use of Virtual Reality to improve the Quality of the Experience for the user.**

Keywords : Model Driven Engineering; Remote Robotics; Home Automation Systems, Industrial Web.

I. INTRODUCTION

Ubiquity is often defined as the ability to be in several places at the same time. In the field of Information Technologies, this definition can be refined in, at least, two ways that may look opposites. The first approach is to consider as in [1] that users are surrounded with "intelligent" systems, that may deliver them the needed information: in this case, computing facilities are used to locate users, to understand their environment, to anticipate their needs and to make it possible for the necessary information to be available anywhere at anytime. The second approach, is to offer people to be able to get information from a system located somewhere and to be able to act safely on it: in this case computing facilities are used to make it possible to be "virtually" present in several places at the same time (e.g. active telepresence).

Both approaches are relying on networks, that nowadays, tend to become transparent to users. They both require communicating devices/objects and distributed software and intelligence. They are both requested.

We definitely place our work in the second approach.

Ubiquitous computing has to face several challenges. Among all of them, and according to us, the following ones seem to be essential:

- Being able to communicate in order to achieve reachability under any circumstances.
- Insuring safety in case of failure of one of the system components.
- Being able to provide the right information at the right moment in an understandable and easy manner.
- Developing tools, such as frameworks, that may help designers to create new applications.

In this paper, we will first describe the three types of systems we are studying, and we will then define the topics we will focus on. We will then describe the top down approach, based on Model Driven Engineering, that we would like to use and will provide you with some results, before concluding.

II. UBIQUITOUS SYSTEMS

The term "ubiquitous systems" is used in literature to describe different classes of systems. The systems we are studying will be described and the objective of our work will be presented.

A. STUDIED SYSTEMS

Basically a ubiquitous system may be seen as a user getting information and sending commands to a distant system, through a communication network. This view covers several realities, and each element (i.e. user, network, controlled devices), may vary from one application to another one. For example, the "user" parameter may take several values according to: interface type used (stand-alone/web), users' context (stationary or mobile), users' skills (regular/experts), users' device (mobile terminal/PC workstation), number of users served (single/multiple). This is also valid for the other two parameters: network and controlled device(s). For example, the network can be wireline/wireless/mixed, small/large, heterogenous (PAN[1], LAN[2], WAN[3]) with multiple protocols (WiFi[4], WiMax[5], TCP/IP[6], HTTP[7], WAP[8], GPRS[9]), etc. Controlled device(s) can be: receiving input (from sensors), giving an output (to actuators), have computing capabilities (provide services), managing others, complicated systems (robotic arms).

For this on-going work we have identified three major classes :

[1] Personnal Area Network [2]
[2] Local Area Network [3]
[3] Wide Area Network [4]
[4] Wireless Fidelity [5]
[5] Worldwide Interoperability for Microwave Access [6]
[6] Transport Control Protocol/Internet Protocol [7]
[7] HyperText Transfer Protocol [8]
[8] Wireless Application Protocol [9]
[9] General Packet Radio Service [10]

- Remote Robotics: this first class may be seen as the historical one, with several projects [11] such as PumaPaint [12], KheponTheWeb [13], or the Autralian Telerobots [14]. Some just intended to prove that remote control over Internet was feasible, others are still in-use and included in course of studies like the Autralian Telerobots [15]. Other experiments can also be found in the field of space [16], or medecine [17].
  All these projects, even if they may be seen as really different in their objectives, costs and technological choices rely on a simple scheme: a user + a network + a machine to control inside a specific environment. It can also be noted that in these cases, the machine under control has some computing facilities (or the server close to it), which makes it a high level of control on its behavior possible.
- Home Automation Systems (HAS, also known as domotic systems): this class of systems will evolve very rapidly in the next mfew onths, in order to offer end-users a way to get and send information to their homes remotely [18].
  Such systems are based on a set of simple devices (smoke, temperature, presence... sensors) that will be interconnected using specific protocols to the family's "Home Internet Box (HIB)", that also provides phone calls, television and web access. Users will interact through classical web browsers on PCs or phones. In this case, devices embed few processing capabilities, network connections rely on an external provider and users are not experts. Nevertheless, the control of one single element in one room may have potentially unexpected effects on the whole house and even the neighborhood. .
- Industrial Web: that is to say plants equipped with a large variety of sensors and actuators interconnected using wired and wireless technologies, in order to improve the production. Compared to HAS, these systems will be controlled by trained employees and devices will embed or not computing facilities. The network will be composed of hundreds of nodes and its reliability will be one of the major concerns for the engineers in charge of such systems.

These three cases show the potential complexity of the deployment of ubiquitous systems due to their variety. The next section will insist on the difficulties we would like to tackle.

B. WORKING DIRECTIONS

As stated in the introduction, several challenges have to be faced in order to build ubiquitous applications.

a) NETWORK: In our work, the network will be considered as a black box. Indeed, for example in HAS, a user will use his phone connected via GSM to his network operator that will provide him an access to the WAN of his Internet Service Provider (ISP). The latter will establish the connection to the HIB that could be connected via WIFI or UPnP or any other protocols to the different home devices, that may have establish an ad hoc network between them. One can see the complexity of the communication, the number of different actors involves, the possible risks of failures (hardware problems, software bugs, power cut, user mistakes etc.) and also, the fact that a large amount of the involved components are not accessible.

We then consider that it is difficult even impossible to be able to manage and to insure the quality of the connection, from the user to the controlled system with the actual technology, and that we have to find other solutions in order to face this problem.

Our approach [19] is to have a permanent idea of the quality of the network, using a network sensor based on a simple Ping/Pong mechanism : one side of the system (either the user or the devices or both) will emit a request to the other side that will send back an answer. Measuring the delay to get the answer will give the emitter an idea of the quality of the connection. Of course, this value provides an information from the past, but, it has been shown that knowing the past is helpful to predict the future, knowing that the Round Time Trip (RTT) is rather constant between two locations[10].

This information has to be delivered to the user and also to the controlled system, and both of them have to adapt their behaviors if necessary. A method based on working modes depending on the estimated quality of the network, GEMMA-Q, has been proposed in previous work [20].

b) HUMAN MACHINE INTERFACE: Remotely controlling a system requires information from the controlled system. Besides, having a mental model of the system and of its environment ito make the right decision and send the right commands is also needed. Depending on the complexity of the controlled system, users will have to be trained and have to get easily understandable information.

Basic interfaces, using sliders, gauges, buttons, etc. are according to us not sufficient in the case of complex environments, like for industrial web or remote robotics and even HAS. High level human machine interfaces based on virtual reality may be used to face this problem while immersing the user in a virtual environment as close as possible to the real environment. It is easier to see an action running than to look at a yellow diode on a dashboard.

Virtual Reality (VR) may also be used to train users and to simulate actions and reactions in a specific situation with a predefined scenario. It can also be used to plan missions.

But, the main interest of VR is to use it in parallel with real time control, which leads to have several views of the same world in terms of position and time. For instance, if you are controlling a robot, you generally get a real-time video to monitor the system, but the camera is in a predefined position. Thanks to VR, you can use a virtual camera to have another view to look at hidden parts. You can also "play" with the time, as you can display the current image of the real robot and also an image of the future, considering that a virtual robot can execute a command immediately whereas the real robot

[10] If the average RTT is x, then the probability that its next value is below two time x is very high.

needs to wait for the transfer on the network to start executing the command: the virtual robot is ahead comparing to the real one and operators may anticipate unexpected behaviors and risks. You could also play your commands virtually before sending them to the real system.

VR is also interesting for Industrial Web and Home Automation Systems, as you can give a more comprehensive view of the system they want to access and control. It can also be used to simulate the results of your actions to verify that what you have planned is correct (lights will be switched off during the day, for example).

c) DEVELOPMENT FRAMEWORK: Developing ubiquitous systems is quite a hard task if you want to insure reliability. It needs to incorporate various technologies, to manage uncertainty and to provide a good Quality of Experience (QoE) to users. Developers will need environments to be able to take all these constraints into account and we propose using a top down approach based on Model Driven Engineering to face this problem. Another approach based on Agents may be found in [21].

III. MODEL DRIVEN ENGINEERING

In this section, we will first describe the approach we choose to follow and then some of the first results we have obtained.

A. GLOBAL APPROACH

The Model Driven Engineering (MDE) approach [22] is based on model transformations. Ideally, starting from a high level description (abstract model) automatic generation tools are able to produce various types of low level models (code) for verification, testing, simulation or execution.

The MDE approach makes models productive more than contemplative, so they can be manipulated during the development process. This facility lets us design a framework that allows the passage from independent platform meta-models (PIM : Platform Independent Model), to instantiate Platform Specific Models (PSM) of ubiquitous systems, as proposed by OMG in the Model Driven Architecture initiative.

UML profiles such as SysML (Systems Modeling Language [23]), MARTE (Analysis of Real-Time and Embedded systems [24]) or TURTLE (Timed UML and RT-LOTOS Environment [25]) are working in this way and may be used for the specification of ubiquitous systems. AADL (Architecture Analysis and Design Language [26]) is also an interesting specification language, while being able to describe software and hardware aspects, devices, buses and also including modes specification.

These different approaches, through their meta-models, give guidelines for the correct development of applications in their respective domains. Engineers will create instances that respect predefined patterns or scenari. Associated frameworks will be able to transform models in various sorts of codes. Our aim is to define such kinds of meta-models, that let the user specify all the needed concepts and build an associated development environment, specifically for ubiquitous systems. Starting from a well-known environment and customizing it for our purposes could be a promising way.

B. FIRST PROPOSAL AND RESULTS

The prototype scheme of our framework is composed of 5 main components: Device, Medium, User, Administrator, and Safety, Security and Data Management (SSDM).

- Device package: it will contain device properties, behavioral schemes and constraints. This information should be obtained as much as possible from constructors.
- Medium package: to let ubiquitous systems work on general and heterogeneous networks, we have to specify the required features for the medium. This package will include the necessary specification for Internet LAN and WAN networks, bluetooth connection, WIFI, Zigbee..., but also a way to specify unknown networks (e.g. black box).
- User package: this package will give all users commands and control for devices. It will also let us specify all the needed procedures to communicate and exchange data.
- Administrator package: each device and users' profile will be specified separately. The Administrator package will include the ubiquitous system configurations, managing exchange data between all actors.
- SSDM package: it will include all strategies and scenari that guarantee reliability and maintainability between all ubiquitous system actors. Also, SSDM will include data management profiles that control, protect, deliver and enhance the value of data and information assets.

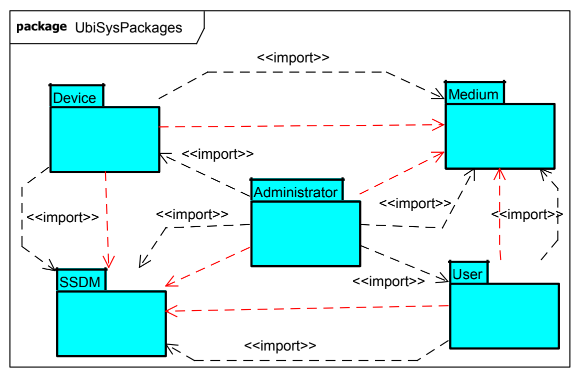

Fig. 1. Package interconnections

The scheme (Figure 1) presents a high level view, and each package has to be refined, like the Device one (Figure 2). Following this proposed model, a device will be represented by several characteristics: a *Type*, a *Context*, a *Behavior*, a *UserInterface* and a *DCManager* (Device Connection Manager). It is important to note that the *Behavior* inherited from *BehaviorMode*, to take working modes into account and is connected to the *Environment*. It is also noticeable, that the *UserInterface* inherited from *VirtualRealityInterface* is itself connected to *MechanicalPropery*.

Compared to other modeling approaches and ontology definitions of the "Device" concept [27][28][29][30], we can say that our model introduces the mechanical structure definition of the device and its interaction with the environment. In addition, we introduce the fall-back scenari and the main links between behavior, environment, context and the mechanical structure of the device. Indeed in [27], hardware and software are specified without mentioning the concept of geometric or mechanical structure. In [28]and [29], the knowledge management and the data management aspects are clearly described but they do not take into account the prevention of fall-back scenarios. In [30], a complete description that includes all devices aspects, particularly the device and its environment, is shown, but no presentation of mechanics or geometry, although the space and the physical property of environment are described. Our modeling approach is based on some parts of mentioned works by adding the needed concepts.

This model of a device will be the refined and detailed to take into account all the needed information. A similar work will be done for the other packages.

IV. CONCLUSION AND FUTURE WORK

Our objective is to create a framework to be able to develop safe ubiquitous applications for Remote Robotics, Home Automation Systems and Industrial Web. We use a top-down approach, while choosing to follow the Model Driven Engineering concepts. We want to focus on working modes, to manage as much as possible uncertainty, and to add virtual reality to give a better feedback to the user thus making it possible to play with point of view and time delay.

Right now, we are not ready to propose real tools and solutions, but a first model has been built, the Gemma-Q methodology is already in use for remote robotics experiments and a virtual model of a real robot has been developed for test purposes. We are now considering the best working environment, probably SysML with some AADL flavours, to be able to propose automatic transformation from model to code.

ACKNOWLEDGMENT

This work is supported by the city of Brest, ”Brest Métropole Océane”, and performed in partnership with Terra-Nova Energy. We thank them for their help.

REFERENCES

[1] M. Weiser, “The computer for the 21st century,” *Scientific American Special Issue on Communications, Computers, and Networks*, 1991.

[2] Wikipedia, the free encyclopedia, “Personal area network,” Jun. 2009. [Online]. Available: http://en.wikipedia.org/wiki/Personal_area_network/

[3] ——, “Local area network,” Jun. 2009. [Online]. Available: http://en.wikipedia.org/wiki/Local_area_network

[4] ——, “Wide area network,” Jun. 2009. [Online]. Available: http://en.wikipedia.org/wiki/Wide_Area_Network

[5] IEEE 802.11 working group, “Ieee 802.11 wireless local area networks,” Jun. 2009. [Online]. Available: http://www.ieee802.org/11/

[6] IEEE 802.16 working group, “Ieee 802.16 working group on broadband wireless local aces standards,” Jun. 2009. [Online]. Available: http://grouper.ieee.org/groups/802/16/

[7] D. Comer, *Iternetworking With TCP/IP Volume 1: Principles Protocols, and Architecture, 5th edition*. Prentice Hall, 2006.

[8] T. Berners-Lee, R. Fielding, and H. Frystyk, “RFC 1945: Hypertext Transfer Protocol — HTTP/1.0,” May 1996, status: INFORMATIONAL. [Online]. Available: ftp://ftp.internic.net/rfc/rfc1945.txt, ftp://ftp.math.utah.edu/pub/rfc/rfc1945.txt

[9] Open Mobile Allicance, “Material from affiliates - wireless application protocol,” Jun. 2009. [Online]. Available: http://www.openmobilealliance.org/Technical/wapindex.aspx

[10] Wikipedia, the free encyclopedia, “General packet radio service,” Jun. 2009. [Online]. Available: http://en.wikipedia.org/wiki/General_Packet_Radio_Service

[11] K. Goldberg and R. Siegwart, *Beyond Webcams : an introduction to online robots*. The MIT Press, 2001.

[12] M. Stein, “Painting on the world wide web : the pumapaint project.” in *In Proceeding of the IEEE IROS’98 Workshop on Robots on the Web*, Oct. 1998, pp. Victoria, Canada.

[13] P. Saucy and F. Mondada, “Khepontheweb: One year of access to a mobile robot on the internet.” *In IEEE International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, 1998.

[14] K. Taylor and J. Trevelyan, “A telerobot on the world wide web.” *1995 National Conference of the Australian Robot Association*, Jul. 1995.

[15] J. Trevelyan, J. Stewart, and N. Scott, “The telelabs project,” Apr. 2009. [Online]. Available: http://telerobot. mech.uwa.edu.au/

[16] P. G. Backes, K. S. Tso, and K. Tharp, “Mars pathfinder mission internet-based operations using wits,” in *Proceedings of the 1998 IEEE International Conference on robotics and Automation*, Leuven, Belgium, May 1998.

[17] WebSurg, “Operation lindbergh : first transatlantic robot-assisted operation,” Sep. 2001. [Online]. Available: http://www.websurg.com/lindbergh/index.cfm

[18] Nokia, “Smart home solution,” Apr. 2009. [Online]. Available: http://smarthomepartnering.com/cms/

[19] J. Vareille, P. Le Parc, and L. Marcé, *”Web remote control of mechanical systems: delay problems and experimental measurements of Round Trip Time*. John Chiasson and Jean Jacques Loiseau Eds, Mar. 2007, chapter of ”Applications of time delay systems”, LNCIS 352.

[20] P. Le Parc, J. Vareille, and L. Marcé, “Long distance remote control over internet: a reliability challenge,” *Intelligent Machines and Factories. Journal of Machine and Engineering ISSN 1642-6568*, Mar. 2005.

[21] M. Wang, J. Cao, S. J.I., V. Raychoudhury, and J. Li, “Ubiquitous intelligent object: Modelling and applications,” in *Third International Conference on Semantics, Knowledge and Grid*, 2007, pp. 236 – 241.

[22] D. C. Schmidt, “Model-driven engineering,” *IEEE Computer Society*, pp. 25 – 31, february 2006.

[23] SysML Partners, “Sysml - open source specification project,” Apr. 2009. [Online]. Available: http://www.sysml.org

[24] M. Faugere, T. Bourbeau, D. Simone, and R. Gerard, “Marte: Also an uml profile for modeling aadl applications,” *12th IEEE International Conference on Engineering Complex Computer Systems*, pp. 359 – 364, July 2007.

[25] P. Apvrille, L. Courtiat, J. Lohr, and C. de Saqui Sannes, “Turtle : a real-time uml profile supported by a formal validation toolkit,” *IEEE Transactions on Software Engineering*, vol. 30, no. 7, pp. 473 – 487, July 2004.

[26] SAE : Society of Automotive Engineers, “Aadl predictable model-based engineering,” Apr. 2009. [Online]. Available: http://aadl.info/

[27] P. Kostelnik, M. Sarnovsky, J. Hreno, M. Ahlsen, P. Rosengren, P. Kool, and M. Axling, “Semantic devices for ambien environment middleware,” *EURO TrustAMI, Internet of Things and Services Workshop*, 2008.

[28] R. Muhammad, L. Youg Koo, and L. Sungyoung, “Knowledge management framework for ubiquitous systems,” *IEEE International Conference on Management of Innovation and Technology*, vol. 1, pp. 412 – 416, June 2006.

[29] S. Khungar and J. Riekki, “A context based data management system for pervasive computing collaborative applications,” in *ACM Conference on Computer Supported Cooperative Work, Chikago, IL*, 2004.

[30] Amigo working group, “Amigo middleware core: Prototype implementation & documentation,” INRIA, FT, ICCS-NTUA, IKER, IMS, Microsoft, TELIN, TID, VTT, Tech. Rep. D3.2, 2OO6.

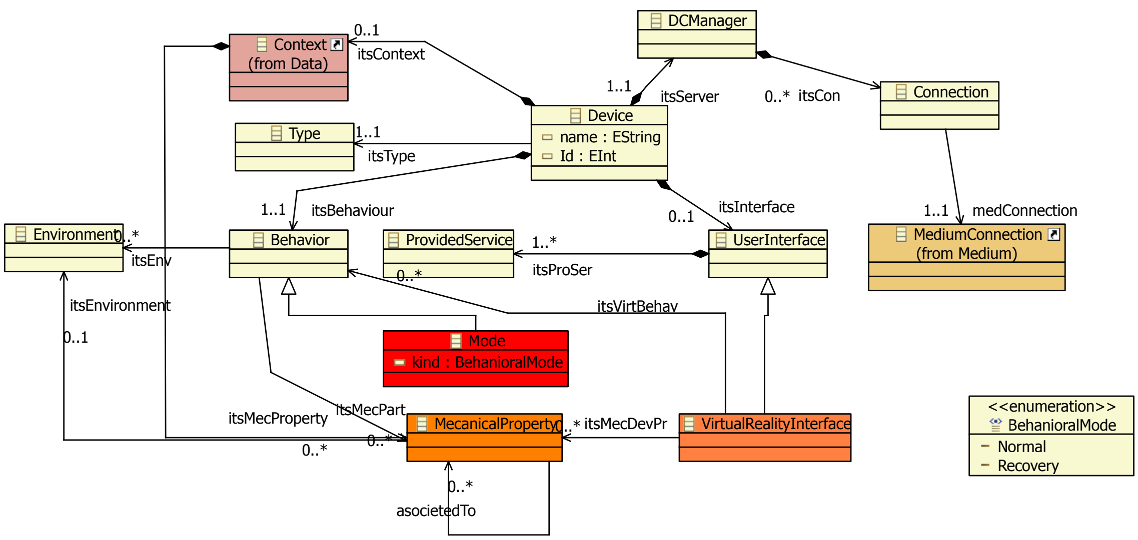

Fig. 2. Device Model