Model-Driven Software Development for Pervasive Information Systems Implementation

José Eduardo Fernandes
Instituto Politéc. de Bragança
jef@ipb.pt

Ricardo J. Machado
Universidade do Minho
rmac@dsi.uminho.pt

João Álvaro Carvalho
Universidade do Minho
jac@dsi.uminho.pt

Abstract

Model-driven development (MDD) conceptions and techniques essentially centre the focus of development on models. They are subject of current research as they allow enhanced productivity, technological platform independence and longevity of software artifacts. Another area of current research is the ubiquitous/pervasive computing area. This field of computing research focuses on the widespread adoption of embedded or mobile heterogeneous computing devices, which, when properly orchestrated, globally compose pervasive information systems (PIS).

This work intends to clarify how should be MDD concepts and techniques structurally consolidated into an approach to software development for PIS. It involves two projects as case studies. From these case studies, it will be proposed methodological insights to design approaches for software development of PIS. While clarifying several issues pertaining to MDD for PIS, it shall promote other research works based on issues needing further study.

1. Introduction

Along the years, organizational, technological, and social evolutions brought a shift from monolithic organization's information systems with well-defined and limited source inputs, into complex, distributed, and technologically heterogeneous information systems. Nowadays, a digital world emerges with prevalence over the real world: everything has or produces information in an increasingly real-time fashion. This world acquires computational and communication capabilities and is ever more ruled with digital information and processes, producing more and faster information about everything and everyone. Future points to a world full of embedded or mobile computing devices with an emerging robotics industry which is "*developing in much the same way that the computer business did 30 years ago*" [1]. This reality and inherent potential has been subject of study and research under the ubiquitous computing field (the term "pervasive computing" is commonly also used with the same meaning).

The innovative computing devices and its widespread availability brought the organizations' attention for its potential on collecting, processing, and disseminating information. Organizations see this as an opportunity to improve their business's processes and therefore to better compete and respond to market pressures and challenges. As consequence, an increasing demand occurs for software development in order to realize intended applications for these pervasive information systems that take advantage of those technologies.

Software engineering is a discipline whose objective is the "*cost-effective development of software systems*" [2, 3]. There have been several research efforts in order to improve performance and convenience on development of these systems, and also to achieve a better satisfaction on accomplishment of requirements and stakeholders' expectations. Improvements on programming and modeling languages, on algorithms, on techniques and paradigms, on processes and tools, on patterns and on the "*level of reuse in system construction*" [4] (sub-routines, objects, components and frameworks) are among approaches undertaken on these research efforts. Model-Driven Development (MDD) constitutes an approach to software design and development that strongly focuses and relies on models, through which "*we build software-platform independent models*" [4]. MDD entails a rising of abstraction from higher-level programming languages to modeling languages.

Model-driven development (MDD) of software systems, taking an approach to development strongly based on models and transformations among models, allows for reduction of semantic gaps among the developed artifacts. It also enables higher independence and resilience of domain models from particular characteristics and changes on system's technological platforms. On the other hand, PIS are generically characterized as potentially capable of having elevated number of heterogeneous devices involved that are subject of a high pace of technological innovations.

To meet the PIS requirements and goals, devices' functionality and interactions are conveniently orchestrated in order to provide appropriated information and actions. Those requirements and goals are ever and ever also subject of changes, as organizations have the necessity to rapidly respond to market competition, and thus establishing new or improving existing business requirements and process. Such necessity, in conjunction with permanent technological innovations and developments, requires that organization's supporting information systems and inherent subsystems, be conceived to deal with change and evolution at minimal business disturbance and reduced costs.

This PhD work intends to clarify how should be MDD concepts and techniques structurally consolidated in an approach to software development for PIS that allows for its consistent development and evolution. It is not intended to either propose new methodologies for software development or notations. It mainly aims, face to current techniques, standards, notations, and model-driven development conceptions and approaches, to contribute with methodological insights to a structured and disciplined use of those on MDD-based software development approach that can effectively and adequately satisfy PIS characteristics and needs. While contributing and clarifying several issues pertaining to MDD for PIS, it shall also promote other research works based on issues needing further study.

2. State of the art

This section starts by presenting ubiquitous computing and pervasive information systems. Then, it introduces MDD fundamental concepts.

2.1 Ubiquitous computing

Ubiquitous computing is a research field of computing technology that started at the 90s of the last century with Mark Weiser's seminal work entitled "*The Computer for the 21st Century*" [5]. In this work, he shared his vision of a new way of thinking about computers. Ubiquitous Computing represents a new direction on the thinking about the integration and use of computers in people's lives. It aims to achieve a new computing paradigm, one in which there is a high degree of pervasiveness and widespread availability of computers or other IT devices in the physical environment. As consequence, the physical world is enriched with the advantages of processing power, storage and communications capabilities of computers.

This new computing paradigm does not simply restrict to enhancing the physical world with embedded computing devices, sensors, actuators or other elements to provide communications among these. It also concerns the way computing is made available for interaction with users in support of their activities. Ubiquitous computing proposes a philosophy that values the nuances of the real world and embodies the assumption that computers should fade into the physical environment in an "*virtual or effective*" invisible way to people [6]. As stated by Weiser, *"Ubiquitous computing takes place primarily in the background. (…) leaves you feeling as though you did by yourself"* [6], ubiquitous computing is gracefully and seamlessly integrated in the environment, allowing for people to not actively notice that it is there. In this way, people can fully focus on completion of the tasks needed to the prosecution of their goals, benefiting from a non-intrusive and non-distracting computing.

Several pervasive computing characteristics, issues and challenges have been identified [7-10]. Context-awareness of applications and easy interoperability of devices and applications are identified as requirements for system support of pervasive applications [11].

Weiser stated that *"The real power of the concept comes not from any of these devices; it emerges from the interaction of all them"* [5]; his statement of "*Applications are of course the whole point of ubiquitous computing*" [12] (and also cited by [13]) reinforces that, among all the innovative and outstanding pervasive technologies, the applications get the final focus. It is through these applications that the ultimate vision's objective is achieved - the invisibility and engendering of calmness of computing on the support and enhancement of everyday activity.

2.2 Pervasive information systems

Pervasive systems and technologies have been increasingly employed either in business domains, trying to improve the way business are done or even to enable new and innovative ways of carrying business, or in more personal or social domains, trying to improve the people's life quality. Several aspects have been focused, such as social concerns [15] and the economic implications of its deployment [16]. The advent of pervasive computing systems enabled information technology to gain a further relevance in its role in human social lives [17], narrowing the relationship between humans and technology and fostering focus on human to human communication. The potential for applications using smart objects is vast, being the limits "*less of a technological nature than economic or even legal*" [18].

Business competition among organizations demands that an organization timely meets market demands with the best suitable and competitive supply at minimal cost. Among others requirements, efficiency of business processes and effectiveness of processes' arrangement constitute central issues to the organizations ability to be competitive and successful. Beyond land and natural resources, human labor, and financial capital, information and knowledge are, fundamental resources of an organization [19]. Information can be used not only as a resource to the production of goods or services, but also as an asset inherent the organization structure, or even as a product to be commercialized [20]. In this context, information, information technologies, information systems, and information management play a crucial role.

Widespread availability of affordable and innovative information technologies promoted the attention individuals and organizations to the efficiency and effectiveness of information management – the way they acquire, process, store, retrieve, communicate, use and share information. The adoption of these technologies can in fact contribute to more profitable and more advantageous business processes performance and ultimately, to maximize potential competitive advantages. To take full benefits from the opportunities offered by information technologies, these need to be "*appropriately integrated within organizational frameworks*" [19]. Services are supported and deployed over the interconnected computing devices and other information augmented objects, enabling higher-level applications that come onto the assistance of the user In this, "*we become aware of the presence flow and processing of information, not only by the individual computing devices, but also, and with a more deep significance, by the overall system that emerges from the interactions of all the computing devices, linking them together in a coherent fashion*" [21]. We can then recognize the presence of some sort of information system, which in this context of pervasive computing, we can denominate as a *pervasive information system (PIS).* Indeed, all these systems dealing with information constitute some form of information system; they gather, collect, process, store and produce information aimed at contributing to an organization or personal needs in order to achieve a set of well-established objectives.

2.3 Model-driven development

Interest and focus on models arise today with further emphasis due to recent developments that resulted into the establishment of important, widely known, and recognized standards, particularly those originated from the Object Management Group (OMG) such as Unified Modeling Language (UML) specification and the Model-Driven Architecture (MDA) initiative. While UML specifies notation and semantics for representing models, MDA tries to raise de abstraction levels at which main development occurs, essentially shifting the focus from coding to modeling. In essence, fundamentally relying on models, MDA proposes for the system development life cycle the development of a Platform Independent Model (PIM) of the system that, free from specific platform technological issues, details the structure and behavior of the system. Given a chosen technological platform, this PIM is transformed into a Platform Specific Model (PSM) that incorporates all the necessary technological details inherent to the chosen technological platform on which the system is to be implemented. From this PSM, system code foundations are generated for the target technological platform. This separation of concerns between PIM and PSM allows that with no further modification to the PIM itself, other technological platforms can be easily targeted, since the PIM still represents the desired system structure and functionality with no contamination of technological details. These standards (and as well as others) represent, through common agreement and acceptance, knowledge of best practices and set up the basis for further innovations or developments. They also enable reuse of knowledge and artifacts, tools' specialization and interoperation, providing thus a "*significant impetus for further progress*" [22].

For some, model-driven development is considered "*the first true generational shift in programming technology since the introduction of compilers*" [22], and it can, in fact, profoundly change the way applications are developed [23]. Automating many of the complex and routine programming tasks, MDD allows for developers to be able to focus on the functionality that the system needs to deliver and on its general architecture, instead of worrying about every technical details inherent to the use of a programming language [23].

The model-driven architecture (MDA), the OMG initiative fostering model-driven development, presents a valuable set of core concepts and a description of how the MDA is to be used [2], allowing a better insight on MDD thematic.

3. Research objectives and approach

As previously stated, this work intends to clarify how should be MDD concepts and techniques structurally consolidated in an approach to software development for PIS that allows for its consistent development and evolution. It is expected that the results indicate how to face the software development attending to issues as: (i) development of software supporting the pervasive information system; (ii) evolution of software for pervasive information system due to changes on new technology or on changes on business requirements; (iii) management and evolution of software due to reconfigurations of the devices that composes de pervasive information systems.

The research approach of this work will be based on two case studies involving two projects developed in the field of ubiquitous and mobile computing that directed their software development on a model-driven software development basis. These projects were selected as subject of this work's analysis as they were based on corresponding scientific and subject areas, and developed with cooperation of the Department of Information Systems at University of Minho at which this PhD is held. The first project presented is the uPAIN project (Ubiquitous Solutions for Pain Monitoring and Control in Pos-Surgery Patients) that was conceived with the purpose to create a networked informational computing system to enhance the anesthesiology services of healthcare centers on the control and monitor at pain level on post-surgery. The second project is the USE-ME.GOV (USability-drivEn open platform for MobilE GOVernment) project, which focuses the development of an open platform for mobility government services. The case studies will help to try to identify common facts, issues and challenges that they faced when using a MDD approach on the project. From these results, and jointly with the research and knowledge acquired on MDD and PIS, it will be proposed methodological insights for design approaches to software development for PIS that contribute to enhanced software development results for those. While clarifying several issues pertaining to MDD for PIS and giving appropriated answers to the research objectives, it shall promote other research works based on issues needing further study.

4. Current work and preliminary results

Revision of literature has been performed. The first case study, corresponding to the uPAIN project, has so far resulted on the description of the main artifacts and techniques used in the project, particularly at requirements level. From the study, several issues and thoughts arose that may influence the strategy taken on the MDD approach for PIS. Among these issues, some relate to functionality management and others relate with proper development of the PIS. Some key ideas relevant for MDD of PIS also emerge, such as the dimensioning of the development, considering types of devices, functional profiles and abstraction, and the need of several related, but independent, development structures of development. The second case study has begun and is actually under research. It is expected in the remainder work to: (i) further explore conceptions and models transformation techniques and concepts that can lead to more automated activities; (ii) better frame the work on a perspective of software architecture. Next steps include: (i) concluding research on second case study; (ii) extract common facts, issues and challenges; (iii) to incorporate the knowledge acquired on results that meets the research objectives.

5. Work plan and implications

The PhD work is structured in the following main activities, for a total period estimated on 36 months:

(1) Literature review (4 months). (i) Exploration of main thematic of ubiquitous/pervasive computing field. (ii) Exploration of model-driven development concepts, techniques, and approaches. The knowledge acquired about those fields will allow for better reasoning and insight on the case studies.

(2) Case studies (14 months). (i) Case study of uPAIN project. (ii) Case study of USE-ME.GOV project. Those projects will allow extracting common facts, issues and challenges that these projects faced. At this stage, it will be made again an incursion into literature to further clarify issues emerged during the case studies.

(3) Reflection and contribution (12 months). (i) Reflection on the previous case studies under a critical analysis of methodological advantages/disadvantages of model-driven based software development for PIS implementation. (ii) Elaboration of methodological insights on model-driven based software development for pervasive information systems implementation, promoting opportunities, benefits, and difficulties of the adoption of model-oriented methodologies to the support of software development for PIS, having in mind the assurance of semantic continuity of software artifacts and integration of technological heterogeneity.

(4) Conclusion (3 months). This period of time is dedicated to elaboration of conclusions about work realized, definition of guidelines for future research, and proper composition of the introductory chapter of

the thesis.

(5) Thesis writing (3 months). The writing of the document of the thesis is done along the realization of the work. Final revision will be performed in order to achieve consistency and finalization.

It is intended to produce, at each end of main activity, the corresponding chapter(s) of the thesis containing the reflection of the work realized. In accordance to the work performed, the work shall be submitted for acceptance either on main international conferences on the areas of research, or relevant international publications.

6. Conclusions

Software supporting pervasive information systems is target of frequent modifications due either to technological evolutions or requirements changes. This requires that software development and artifacts must be, in its essence, well designed, developed and deployed. By this way, it will be possible to satisfy requirements and explore the potential offered by the pervasive computing, maximizing the revenue of this kind of systems. Model-driven development approaches have the potential to accommodate such needs, allowing for higher independence of technological platforms, reduction of semantic gaps, greater development productivity and longevity of software artifacts.

The definition and correct use of MDD-based conceptions, techniques, and procedures to software development for PIS assume major importance on current days. This work aims to provide scientific contribution for appropriate model-driven development approach to pervasive information systems.

7. References

[1] B. Gates, "A Robot in every home," in *Scientific American*. vol. 296, 2007, pp. 58-65.

[2] OMG, "OMG's MDA Guide Version 1.0.1," 2003.

[3] I. Sommerville, *Software Engineering*, 6th ed.: Addison-Wesley, 2001.

[4] G. Miller, S. Ambler, S. Cook, S. Mellor, K. Frank, and J. Kern, "Model driven architecture: the realities, a year later," in *Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications* Vancouver, BC, CANADA: ACM Press, 2004, pp. 138-140.

[5] M. Weiser, "The Computer for the 21st Century," *Scientific American*, vol. 265, pp. 94-104, September 1991 1991.

[6] M. Weiser, "Hot topics-ubiquitous computing," *Computer*, vol. 26, pp. 71-72, 1993.

[7] G. D. Abowd and E. D. Mynatt, "Charting Past, Present, and Future Research in Ubiquitous Computing," *ACM Transactions on Computer-Human Interaction*, vol. 7, pp. 29-58, March 2000 2000.

[8] M. Satyanarayanan, "Pervasive computing: vision and challenges," *Personal Communications, IEEE [see also IEEE Wireless Communications]*, vol. 8, pp. 10-17, 2001.

[9] K. Lyytinen and Y. Yoo, "Introduction [Issues and challenges in ubiquitous computing]," *Communications of ACM*, vol. 45, pp. 62-65, December 2002 2002.

[10] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *Computer*, vol. 36, pp. 25-31, 2003.

[11] R. Grimm, "One.world: Experiences with a Pervasive Computing Architecture," in *Pervasive Computing, IEEE*. vol. 03, 2004, pp. 22-30.

[12] M. Weiser, "Some computer science issues in ubiquitous computing," *Communications of ACM*, vol. 36, pp. 75-84, July 1993 1993.

[13] G. D. Abowd, E. D. Mynatt, and T. Rodden, "The human experience [of ubiquitous computing]," in *Pervasive Computing, IEEE*. vol. 1, 2002, pp. 48-57.

[14] U. Hansmann, L. Merck, M. S. Nicklous, and T. Stober, *Pervasve Computing*, 2nd ed.: Springer, 2003.

[15] A. Stone, "The dark side of pervasive computing," in *Pervasive Computing, IEEE*. vol. 2, 2003, pp. 4-8.

[16] M. Langheinrich, V. Coroama, J. Bohn, and M. Rohs, "As we may live – Real-world implications of ubiquitous computing," Distributed Systems Group, Institute of Information Systems, Swiss Federal Institute of Technology, ETH Zurich, Switzerland April 2002 2002.

[17] D. C. Dryer, C. Eisbach, and W. S. Ark, "At what cost pervasive? A social computing view of mobile computing systems," *IBM Systems Journal*, vol. 38, pp. 652-676, 1999.

[18] F. Mattern, "The vision and techical foundations of ubiquitous computing," *UPGRADE*, vol. 2, pp. 3-5, Octobber 2001 2001.

[19] A. P. Sage and W. B. Rouse, "Information Systems Frontiers in Knowledge Management," *Information Systems Frontiers*, vol. 1, pp. 205-219, 1999/10 1999.

[20] S. R. Gordon and J. R. Gordon, *Information Systems - A Management Approach*, 3rd ed.: Wiley, 2004.

[21] J. E. Fernandes, R. J. Machado, and J. Á. Carvalho, "Model-driven methodologies for pervasive information systems development," in *MOMPES'04 -1st International Workshop on Model-Based Methodologies for Pervasive and Embedded Software*, Hamilton, Ontario, Canada, 2004, pp. 15-23.

[22] B. Selic, "The pragmatics of model-driven development," *Software, IEEE*, vol. 20, pp. 19-25, 2003.

[23] C. Atkinson and T. Kuhne, "Model-driven development: a metamodeling foundation," *Software, IEEE*, vol. 20, pp. 36-41, 2003.