Current Practices on Model-based Context-aware Adaptation

Vivian Genaro Motti[1], Dave Raggett[2], Jean Vanderdonckt[1]
[1]LILab – Louvain Interaction Laboratory - Université catholique de Louvain
Place des Doyens 1 – Louvain-la-Neuve 1348
[2]W3C/ERCIM 2004, Route des Lucioles – Sophia Antipolis - France
vivian.motti@uclouvain.be

ABSTRACT
The scientific community has already investigated in depth the benefits of combining model-based approaches for implementing context-aware adaptation. As benefits, it can be highlighted: lower development costs, faster time to market, higher usability levels, optimal usage of the resources available and a better user interaction. Although these benefits are claimed, for practitioners it may be not always evident that they actually compensate for the costs of incorporating such practices into daily work practices. Based on the hypothesis that such practices are not widely adopted, we defined and applied a survey of practitioners to identify if and how they actually perceive and adopt such approaches. This paper describes the survey, its application and discusses the results obtained.

Author Keywords
Survey; Context-awareness; Model-based Approaches.

ACM Classification Keywords
D.2.2 [**Software Engineering**]: User interfaces. H.1.2 [**Information Systems**]: Human factors. H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems. H5.2 [**Information interfaces and presentation**]: User Interfaces – *User-centered design*.

General Terms
Human Factors; Design.

INTRODUCTION

Current contexts of use vary according to the devices, platforms, user profiles and environments. The differences posed by these contexts require applications whose user interfaces (UI's) are able to recognize the contextual information [Dey, 2000] and to adapt accordingly. Because it is neither feasible nor scalable to implement UI's for each variation of the context, methodologies that support and facilitate the development phases have been proposed [Motti, 2013], e.g. model-based approaches (MDA).

Model-Based User Interface Design facilitates interchange of designs through a layered approach that separates out different levels of abstraction in user interface design. When a model-based approach is adopted for the development of user interfaces (MBUID) [Meixner, 2011], first an abstract UI must be defined. This definition, called either meta-UI or extra-UI [Coutaz, 2006, Sottet, 2009], includes necessary tasks and elements for the end user to achieve his or her goal. This definition is technology-independent, i.e. it is valid regardless of the context of use (platform, device, modality, user, etc.). MDA claims that once the abstract specification of the UI is defined, several instantiations can be more easily derived, based on specific characteristics and constraints of the target contexts.

Although conceptually such solutions aim at better usability levels and can be straightforwardly employed, in practice stakeholders tend to believe that its costs outweigh the promised benefits, i.e. a steep learning curve is required to understand its concepts, to use it, to apply it in a large scale, additional phases must be added to the development process, more resources are needed, and so on.

To verify whether the I.T. companies and their stakeholders actually adopt model-based approaches, context-awareness, and adaptation of UI's in their daily work practices, a survey was defined and applied. It aims at gathering information about: the users profile (years of experience, size of the enterprise, main role), the context information (perceived relevance, adoption, methods employed), adaptation (usage and information sources) and models (importance, approaches, benefits and drawbacks).

Thirty-three participants replied to the survey. The results obtained aid to: better understand and characterize some of the working practices that are currently adopted, to foresee some potential tendencies in this domain and also to gather main motivations for adopting model-based approaches for context-aware adaptation.

This paper presents the survey, its definition, its application steps, the results obtained, their analysis and discussion. This paper is organized as follows: Section 2 discusses related works and fundamental definitions; Section 3 presents the hypotheses, survey and methodology; Section 4 presents results obtained; Section 5 discusses them presenting future tendencies, final remarks and future steps.

RELATED WORKS

According to Bézivin (2004) models are a simplified abstract representation of a system. Model-based approaches rely on these abstract representations to provide the foundations for code generation and reverse engineering. They aim at alleviating the cost of code production while improving its qualities [Calvary, 2002]. In a Model-Driven Development (MDD) the use of models for software development is emphasized, as well as the need of transformations in all phases of the development, from system specification to implementation and testing. These transformations could enable the automated implementation of a system in successive steps [Koch, 2006]. Existing models have already achieved a high level of sophistication, however a number of implicit assumptions are often made. Although these assumptions tend to simplify implementation issues, they also tend to limit the solutions provided [Calvary, 2002].

Context-awareness consists in the capability of identifying and considering contextual information [Dey, 2000] in the development of applications, so that their UI's can be personalized, delivered across multiple platforms and adapted according to their location. Customized UI's consider the context of the users, e.g., their preferences, device characteristics, or bandwidth restrictions. Context information may influence all three applications dimensions: content, navigation, and presentation [Koch, 2006]. It is clear that a context-aware application aims at higher usability levels and a better user experience. However, to appropriately identify relevant context information and correctly incorporate it within a system, additional efforts and resources are required during the development phases. Context aware design then involves a study of the range of contexts expected in everyday use, and identifying the kinds of adaptation suitable in each case. Clearly, adaptation involves efforts that may not otherwise be necessary in a traditional design approach. Furthermore, there are trade-offs and drawbacks in quality that must be carefully handled, for instance concerning privacy, performance and scalability.

Adaptation consists in employing the context-awareness in a way that the information gathered is applied for changing, modifying or transforming the application. For Sottet (2009), due to the heterogeneity of contexts of use, adaptation is much more complex than selecting the most appropriate modality when the context of use changes. The goal is to provide users an application that is more suitable according to the context of use. However, adaptation also poses some drawbacks. For instance: users may feel confused with the changes of the UI's, performance issues may arise, privacy and scalability are also often impacted.

Context-aware adaptation aims at providing users with an enhanced user experience through improved usability in a given context. Model-based approaches are able to support the design and development of context-aware adaptation.

The traditional techniques for software development tend to involve separate teams targeting a specific platform. As a consequence, there are many challenges for coordinating, sharing and managing information, as well as for maintaining consistency among the resulting outcomes. The model-based approaches focus in shared models, which enable separation of different levels of design concerns. Theoretically, with such an approach not only the consistency among designs is improved, but also the people's specific expertise is better used, regardless of the activity considered, i.e. data modeling, programming, usability or graphics design.

Although the three concepts mentioned above target at benefits for both end users and developers, they may imply in additional costs for development. For Sottet (2009), experience shows that industry still remains code-centric and that models still fit in the contemplative category in HCI: obviously they help in reasoning, they might look nice, nevertheless in practice developers love coding above all [Sottet, 2009]. Therefore, they may hesitate in adopting different approaches during the development processes. In 2011 Meixner et al. (2011) published an article summarizing the status of model-based user interface development (MBUID). It gives an overview about definitions, approaches and projections for this domain. This overview is comprehensive, however it does not cover the actual application of MBUID in current work practices.

Actually some surveys have already been dedicated to investigate MBUID, however they are focused, e.g. on literature review [Da Silva, 2001], testing approaches [Neto et al., 2007], design tools [Perez-Medina, 2007] or transformation tools [Schaefer, 2007]. The survey presented in this paper focuses on clarifying how stakeholders actually understand, adopt and consider the concepts of interest. The definition and application of this survey is presented and discussed in the next sections.

METHOD

While for the scientific community it is clearer that model-based approaches and context-aware adaptation provide benefits for both stakeholders and end users, for the industry and its practitioners, it may be not so obvious whether the benefits actually compensate for the (additional) costs involved. To investigate it more in depth, a survey has been defined and applied.

This surveys aims at investigating two main hypotheses:

> H1) Stakeholders are aware of the importance and the benefits of considering: context-awareness, model-based approaches and adaptation.
>
> H2) Stakeholders do not fully incorporate into their daily work practices: context-gathering, model-based approaches and adaptation.

The survey has been structured as follows:

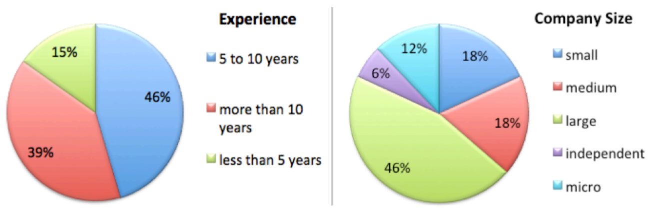

Figure 1. Profile of the Participants concerning: their years of experience (left) and size of their company (right) (n=33)

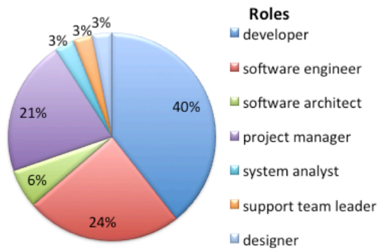

Figure 2. Profile of the Participants (n=33) concerning their main roles and activities

Target. The respondents of this survey consist of practitioners working for Information Technology companies, with different expertise, background and roles (e.g. project manager, software engineers, architects, developers, designers, system analysts). They belong to different countries, (e.g. Belgium, Brazil, France, Germany, U.K., Spain) and companies (e.g. Yahoo, Sony, BNP Paribas – Fortis, etc).

Structure. The survey is structured in 4 main parts: the first part gathers details about the practitioner profile (years of experience in the domain, main role, size of the company); the second part concerns context-awareness (dimensions and information considered, their importance, methods employed, and level of adoption); the third part covers adaptation techniques (*how* they are identified, applied and presented); and the fourth part gathers information about adoption of model-based approaches and their perceived importance (advantages and disadvantages).

Application. The survey has been defined and published online using google docs. A message has been sent via email to invite participants to collaborate in the study. All the results are anonymous. The average time to complete the survey ranges from 5 to 10 minutes. 50 persons have been contacted and 33 answers have been obtained. The results are presented and discussed in the next sections.

RESULTS

This section presents the results obtained with the application of the survey, respective figures and graphics.

Profile. The average profile of the participants consists of I.T. practitioners, working for companies or as independent consultants. The roles vary among software architects, engineers, developers, project managers, system analysts and team leaders. Small, medium and large companies of different countries have been considered. The graphics illustrated by Figure 1 and 2 show these results.

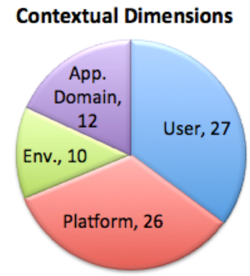

Figure 3. Absolute values for the dimensions of the context information informed as considered (n=33)

Concerning the *years of experience*, as Figure 1 left illustrates, 46% of the participants informed that they have been working from 5 to 10 years in the I.T. domain, 39% have been working for more than 10 years and only 15% for less than 5 years. Concerning the *company size*, as Figure 1 right illustrates, 46% of the participants work for large companies, 18% for small companies, 18% for medium-sized companies, 12% for micro-entities and 6% work independently (for instance as consultants or free-lancers). Concerning the main roles, as Figure 2 illustrates, 40% of the participants stated that they are developers, 24% software engineers, 21% project managers, 6% software architects, 3% system analysts, 3% support team leader and 3% designers.

Context. In absolute numbers, out of the 33 participants, 27 stated to consider as contextual dimension the users, 26 the platform, 12 the application domain and 10 the environment. Concerning the perceived relevance of context and its actual usage, as Figure 4 illustrates, the *user* is classified as the most relevant dimension for most of the participants, followed by the platform and the application domain, while the *environment* is considered as the least relevant dimension. These results concern the perception of the participants regarding the relevance of context dimensions. When compared with the actual usage we note that again the user and platform are considered as the most relevant dimensions, while in practice application domain and environment are the least considered dimensions. However, although users are perceived as the most relevant dimension of context, in practice not always is considered. While the platform is more considered in practice than perceived as relevant. The environment is perceived as relevant and considered in practice, and the application domain is more considered as relevant than actually used in practice.

Figure 5 illustrates by means of a stacked bar graphic how an absolute amount of participants consider the dimensions in both: level of relevance (left) and actual usage (right). By analysing these graphics, it is possible to note that users are perceived as the most relevant dimension (by half of the participants), followed by the platform, application domain and environment. The same tendency was observed concerning the practical usage of the dimensions (although with less significant differences). The environment was the

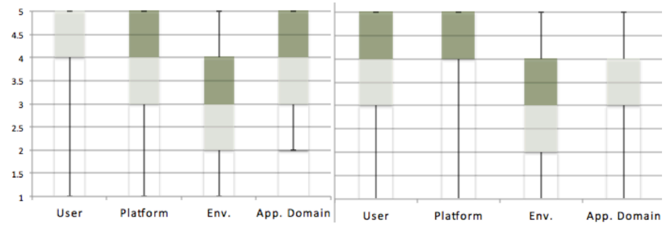

Figure 4. Box plot graphics comparing how participants perceive the relevance of context (left) and actually use it (right) (n=33)

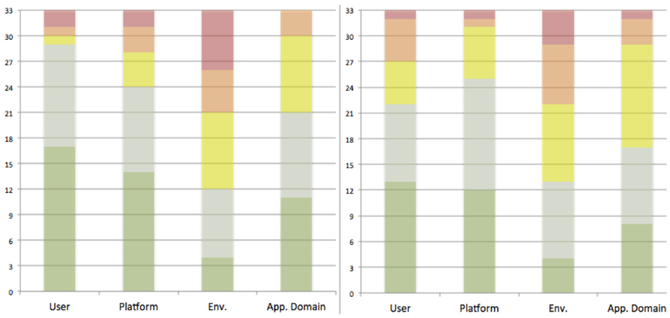

Figure 5. Comparison between the amount of participants concerning the perceived relevance of context (left) and actually use it (right), Likert scale of 5 points (n=33)

dimension considered as least relevant by more participants, and this dimension is also lesthe least used in practice. Actually, in practice, all participants informed to consider at some extent: platform, application domain and user.

Concerning the context dimensions that the participants informed to actually use while developing interactive systems, as Figure 6 illustrates, out of 33 participants, 27 stated to consider information about the user, 26 consider the platform, 14 the application domain and 10 the environment. These amounts are not exclusive. However we note that only 4 out of 33 participants informed to use all 4 dimensions simultaneously (user, platform, environment and application domain), also 9 use 3 dimensions. And the majority, 19 out of 30, stated to use just 2 (15) or even just 1 dimension at a time (4).

Figure 7 illustrates a Venn diagram presenting the dimensions of user context that are taken into account

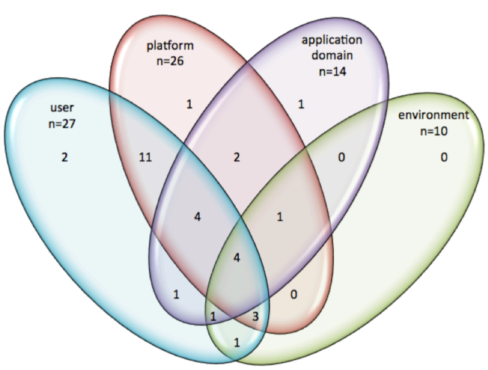

Figure 6. Venn diagram illustrating the context information considered by participants. Only 4 out of 33 participants informed to consider 4 dimensions jointly.

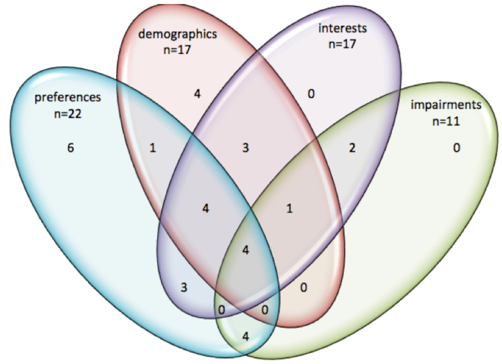

Figure 7. Venn diagram for the user context dimensions (n=33)

during the development phases. Most of the participants declared to consider user preferences (22 out of 33), followed by demographics (17 out of 33) and interests (17 out of 33). Impairments however are only considered by 11 out of 33 participants. Usually a combination of 2 dimensions is considered (13 out of 33 participants), e.g. impairments and preferences (4), or interests and demographics, 3 out of 29. Only 4 participants informed to simultaneously consider all 4 dimensions.

Concerning the methods adopted to gather information about the user, 18 out of 33 participants stated that they rely on observation, 14 on guidelines, 13 on interviews, 6 on surveys and 7 informed to not adopt any methodology per se but they just guess information. Two participants informed to collect and monitor real world usage data. Ten participants informed to adopt just one method, while 13 informed to adopt 2, 6 informed to combine 3 methods and only 1 informed to combine all 4 methods (guidelines, interviews, observations and surveys).

For the *platform*, as Figure 8 illustrates, the majority of the participants (28 out of 33) informed to consider the device and also 26 out of 33 informed to consider the technology, 26 consider the connections, and just 4 take the accessories into account. Just 5 out of 33 participants informed to not consider the device per se, but they (2 participants) consider the connections or (3 participants) the connections and the technologies available.

To gather information about the platform, 19 out of 33

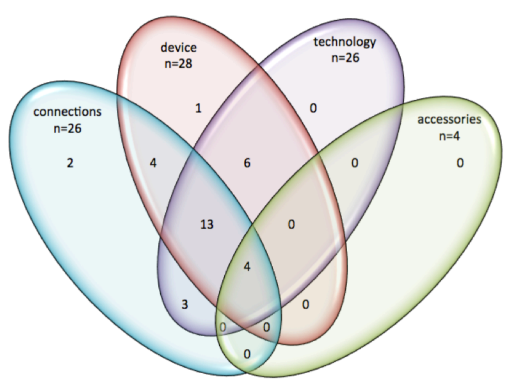

Figure 8. Venn diagram illustrating the platform context (n=33)

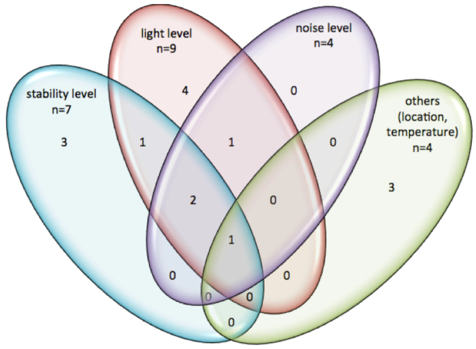

Figure 9. Venn diagram illustrating the environment context (n=33)

participants informed to use a default specification; among which 7 informed to also perform automatic tests, of which 2 also observe the context of use and 1 also track the user interaction. Two participants perform only automatic tests, 4 only observe users and just 1 participant stated to just interview users. Just 1 participant stated to track the user interaction (but combined with 3 other methods). Regarding the amount of methods, while the majority (18) employs just 1 technique at a time, the remaining participants (15) combine more than one technique: being 3 participants that combine 3 methods and 12 combine 2 of them.

Regarding the *environment*, most of the participants (18 out of 33) stated to not consider any information. As Figure 9 illustrates, among the remaining participants (15), 9 stated to consider the light level, 7 the stability level, 4 the noise level, and 4 considered other information, as the user location (via GPS), temperature, and the 3G coverage. Concerning the methods adopted, observation sessions, user interviews, and surveys are applied. Just 1 participant informed to use sensors.

Adaptation. To search for adaptation information, the participants informed to use: pattern libraries (13 of 33), public guidelines (10 out of 33), embedded features (8 out of 33), online repositories (8 out of 33). However, approximately half of the participants (17 out 33 participants) informed that no adaptation is provided. Only 1 participant informed to combine 4 sources of information, while 7 mentioned to combine 3, 9 to combine 2, and 16 participants informed to use only 1 source (or no source).

Concerning adaptation strategies, 7 out of 33 participants informed to use graceful degradation, 10 informed to use progressive enhancement, and 4 informed to combine both strategies. The majority of the stakeholders though (19 out of 33) do not use any of these, and just 1 participant informed to use animation to smoothly present to the end users the transition between an original UI and its adapted version.

Models. Concerning the usage of the models, approximately half of the participants (17 out of 33) informed they are not used at all, 7 participants informed to use MDE, 12 use UML diagrams among which 3 informed to use them combined with MDE.

Concerning their perception about models, stakeholders can certainly perceive some benefits with their adoption, such as: reuse, documentation, communication, a common language. However, in practice, they seem to not be largely adopted. The participants remarked as main disadvantages, that models: are hard to customize, delay the process to achieve results, are difficult to maintain, have incomplete definitions, require more development efforts and more skills to be used, have complex definitions and that there is currently a lack of standards for models, causing some issues, as inconsistency and incompatible outcomes.

The participants of the survey remarked some positive aspects of adopting models during the development process.

The main benefits that have been mentioned can be summarized in four aspects, the models:

- Provide a common language and standards
- Facilitate reuse
- Generate systems that are more complete and have more qualities
- Aid communication, discussion and analysis

As negative aspects of adopting models, some remarks have been highlighted, for example, the models:

- Are hard to customize, to adapt, and to maintain
- Lack support (or have incomplete support)
- Are hard or slow to synchronize changes
- Require more expertise, efforts and time

One crucial aspect has been classified as both positive and negative for different participants: the optimization of the development phases. While some participants believe that fewer efforts are needed, others stated that more developments efforts are required, mainly in terms of expertise and time. Another aspect of disagreement, concerns achieving a working prototype, while some participants consider it easier to do with models, others think it is actually harder. The same applies for the complexity of the projects, while one participant stated that models are not suitable for simple projects, other participants stated that models are not suitable for highly complex projects.

DISCUSSIONS

By applying the survey a variety of stakeholders have been reached. They have different expertise levels, years of experience, and work for different companies. Although almost 50% of the participants declared to work for large companies, they have different roles and expertizes. Different countries have also been covered, contributing to the variety of the sample of survey participants.

Regarding the context dimensions considered, it is clear that mainly the user and platform play an important role, while application domain and environment are not always considered as so relevant. Actually it is possible that stakeholders were confused with such definitions, as some participants commented after replying the survey. Sometimes the concept of *environment* was misunderstood, for instance being interpreted as the editor per se, and not the situation where the interaction takes place, or the circumstances where the interaction takes place. The term *application domain* also raised some discussion, being misunderstood with cultural aspects of the user. Even by providing a short description about these concepts and a couple of examples, not all participants could successfully comprehend such definitions.

As context dimensions mostly considered, the user and the platform are certainly the most important ones, maybe because by ignoring or omitting them, could prevent the user interaction. However, to complement such results, it would be necessary to investigate to which extent the contextual information is actually covered. Although there is some difference between the perception of relevance for context and its actual usage, it is not highly significant.

Concerning the H1, which states that stakeholders are aware of the importance of the concepts, it holds for context aspects, at least in terms of user, platform and application domain. Environmental aspects are not considered as important, or maybe it may be not clear for stakeholders *what* environment states for and *how* it can be effectively useful. Concerning H2, most of the participants stated to use context information, at least to some extent, while developing projects.

Adaptation seems to be ignored by most of the participants, since 17 out of 33 stated to provide no adaptation and to consider a standard scenario instead. This may be a result of previous work practices in software development, in which a conventional context of use was commonly adopted (i.e. an able-bodied user, a Desktop PC, and a stable environment). Besides this, it is possible that stakeholders are not aware of *what from* and *how to* consider context information. We could deduce that the participants of the survey are (to some extent) aware of the importance of adaptation, since they stated to consider context-awareness while developing their applications, which validates to some extent H1. However, concerning H2, we clearly note that adaptation is not largely employed, which could result in static applications that may be not suitable for the dynamic and varied contexts of use, in which nowadays the interaction actually takes place.

By analyzing the results regarding the perspective of the participants about models, it is clear that while they can perceive several benefits, they are still skeptical about their adoption. Mainly the lack of support to use models or existing solutions that are currently incomplete, force stakeholders to look for and adopt other alternatives. It seems that without more complete frameworks and definitions, the usage of models may be reduced for academic community or for specific situations or projects.

Concerning H1, it is possible to note that most participants are able to recognize the importance of model, however, concerning H2, we noticed that models are not widely used. Being useful to support certain activities, but not fully adopted. We selected 4 commentaries provided by the participants that support this hypothesis:

> *I believe that models are very relevant and useful but the lack of "easy to use" applications, "easy to draw models" puts a certain resistance for developers to use these tools*

> *… if the model-based approach is directly responsible for the generated code and any changes in the code automatically reflects in the model then it would be extremely relevant to have this kind of approach in my development phase.*

> *…maintaining the docs and the code in a disjoint manner makes me waste some of the time [...] given that the coding sometimes needs to be changed to work.*

> *…I'm not sure models could be used in our domain: UIs are very complex and uses custom widgets*

By analyzing the commentaries above we tend to believe that only by having more mature support, frameworks, standards and tools, stakeholders could see more advantages in using models, and then actually incorporating them into their current work practices.

The lack of consensus regarding the advantages and disadvantages of models may be justified by the fact that these assumptions are dependent on the project itself, so in certain cases it is obvious that more resources are needed, while in other cases the development is certainly optimized. For the complexity it seems like there is a range in which models are suitable, however further investigations are needed to precise specific figures and criteria to identify and measure not only the complexity levels of projects, but also the costs of applying models, aiming as such to effectively identify when it is suitable to actually adopt model-based approaches.

FINAL REMARKS

This work presents initial results about the current work practices of stakeholders regarding: context-awareness, adaptation and model-based approaches. While most of the stakeholders seem to recognize the relevance and benefits of such practice, still they are not considered or just partially adopted in the software development phases. Given that the complexity and efforts needed to incorporate them have been pointed as main drawbacks, we believe that by facilitating and supporting them through tools, it may increase their adoption.

For instance concerning the contextual dimensions, participants in general tend to pay less attention to the environment in which an application is used, than to other factors such as the user, the platform and application domain. This is reflected in the resulting designs they are able to produce. Improved tools and training would allow designers to design applications that adapt better to contextual changes such as the geolocation, ambient light and sound levels of the environment, and as such to improve the user interaction.

As future works we plan to perform deeper studies to complement and refine the results currently obtained and as such reach more conclusive interpretations.

ACKNOWLEDGMENTS

This work received funding from the European Commission's Seventh Framework Program under grant agreement number 258030 (FP7-ICT-2009-5).

REFERENCES

1. Bézivin, J. (2004). On the need for megamodels. In Proceedings of the Best Practices for Model-Driven Software Development, Workshop, held with OOPSLA.

2. G. Calvary et al. The cameleon reference framework, cameleon project. 2002. http://giove.isti.cnr.it/projects/cameleon/ pdf/CAMELEON20D1.1RefFramework.pdf.

3. Calvary, G., Coutaz, J., Thevenin, D., Limbourg, Q., Souchon, N., Bouillon, L., Vanderdonckt, J. (2003). A Unifying Reference Framework for Multi-target User Interfaces. Interacting with Computers, Elsevier Science B.V., June, 2003,15(3),289O308.

4. Coutaz, J. Meta-User Interfaces for Ambient Spaces. In Proc. of the 5th Int. Ws. on Task Models and Diagrams for Users Interface Design: TAMODIA 2006, pp 1-15, Coninx, K., Luyten, K. and Schneider, K. A. (eds.), Springer LNCS 4385. Hasselt, Belgium, October 23-24, 2006.

5. Da Silva, P. P. "User interface declarative models and development environments: A survey." In Interactive Systems Design, Specification, and Verification, pp. 207-226. Springer Berlin Heidelberg, 2001.

6. Dey, A.K, Abowd, G.D Towards a Better Understanding of Context and Context-Awareness, CHI 2000 Workshop on The What, Who, Where, When, and How of Context-Awareness, The Hague, Netherlands, April 1-6 2000.

7. J. M. C. Fonseca et al. W3C Model-Based UI Incubator Group Final Report. 2010. http://www.w3.org/2005/Incubator/ model-based-ui/XGR-mbui-20100504/

8. G. Meixner, F. Paternò, and J. Vanderdonckt. Past, present, and future of model-based user interface development. i-com 10(3): 2-11, 2011. http: //giove.isti.cnr.it/attachments/publications/ icom%202011%200026%20-%20model-based.pdf.

9. V. G. Motti, and Vanderdonckt, J. A Computational Framework for Context-aware Adaptation of User Interfaces. Seventh International Conference on Research Challenges in Information Science, RCIS 2013, Paris, France, May 29-31 2013. IEEE 2013.

10. A. C. D. Neto, R. Subramanyan, M. Vieira, and G. H. Travassos. 2007. A survey on model-based testing approaches: a systematic review. In Proc. of the 1st ACM international workshop on Empirical assessment of software engineering languages and technologies: held in conjunction with the 22nd IEEE/ACM Int. Conf. on Automated Software Engineering (ASE) 2007 (WEASELTech '07). ACM, New York, NY, USA, 31-36. DOI=10.1145/1353673.1353681 http://doi.acm.org/10.1145/1353673.1353681

11. F. Paternò, C. Mancini, and S. Meniconi. ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models. INTERACT '97 Proceedings of the IFIP TC13 International Conference on Human-Computer Interaction, Pages 362-369, 1997.

12. J. L. Perez-Medina, S. Dupuy-Chessa, and A. Front. 2007. A survey of model driven engineering tools for user interface design. In Proc. of the 6th int. conf. on Task models and diagrams for user interface design (TAMODIA'07), Marco Winckler, Philippe Palanque, and Hilary Johnson (Eds.). Springer-Verlag, Berlin, Heidelberg, 84-97.

13. R. Schaefer. 2007. A survey on transformation tools for model based user interface development. In Proc. of the 12th int. conference on Human-computer interaction: interaction design and usability (HCI'07), Julie A. Jacko (Ed.). Springer-Verlag, Berlin, Heidelberg, 1178-1187.

14. Sottet, J.S., Calvary, G., Favre, J.M., Coutaz, J.: Megamodeling and Metamodel-Driven Engineering for Plastic User Interfaces: MEGA-UI. In: Human-Centered Software Engineering. Springer Human-Computer Interaction Series, pp. 173–200 (2009)