# Model-driven Generation of Ergonomic User Interfaces in Visualization Services

**Guillaume Gronier, Sylvain Kubicki, Alain Vagner, Lou Schwartz, Eric Montecalvo, Thomas Altenburger**
Public Research Centre Henri Tudor
29, av. J.F. Kennedy
L-1855 Luxembourg-Kirchberg
{first name.name}@tudor.lu

**Gilles Halin**
Research Centre in Architecture and Engineering (CRAI)
2, rue Bastien Lepage
F-54001 Nancy
gilles.halin@crai.archi.fr

## ABSTRACT
The main goal of this work is to provide a cooperative IT services-based support for projects where heterogeneous actors teams from different organizations have to collaborate in order to realize a set of activities. This kind of "highly collaborative" environment exists in the construction sector, e.g. during building construction stage, and requires some flexibility and efficiency regarding work method and supporting tools. The research applies model-driven architecture to interface generation and takes into account people's profile in order to customize generated interfaces and to improve their ergonomics. A case study is also presented in the visualization services design.

### Keywords
User interfaces, adaptation, usability, visualization, architecture, visualization services, construction sector.

## INTRODUCTION
In the framework of highly-collaborative environments, such as construction projects involving numerous practitioners, there is a clear need for interface plasticity. Plasticity is the capacity of interactive systems to adapt to the context of use while preserving the usability of the system [10]. Since the considered context is dynamic, the user interface should be able to adapt itself accordingly. An important issue of plasticity is the application of the concepts of Model Driven Engineering (MDE) to the user interface generation process. With MDE applied to plasticity, it should be possible to model the different models representing the application and its context in order to generate an efficient user interface.

So as to better understand this kind of process, the first priority is to identify the main characteristics of highly collaborative environments, then see how we can adapt an interface to the profile of a user, and to his context; the link between our models and the Service Oriented Architecture (SOA); the main languages for MDA-oriented user interface generation; and finally the main usability techniques that can be used in our context in order to certify the quality level of the generated interfaces.

In Architecture, Engineering and Construction (AEC) industry, our case study sector, each construction project requires well-adapted software-based services and user interfaces to improve the efficiency of business collaborations as well as the quality of end-user experience. In such collaborative environments each user is involved in a wide network where he plays a role, characterized by his position in the activity organization, his skills and competencies, his contractual relationship to the other parties. Therefore the services and Human-Computer Interfaces (HCIs) offered to him have to be well adapted to these characteristics of use.

## ERGONOMIC USER INTERFACE ADAPTATION

### Personalizing User Interfaces
Nowadays, Human-Computer Interactions are part of our everyday life, from computers to mobile phones, and in the near future, they will be practically in every artifact. There is a proliferation of devices, acting in more and more different contexts and interacting with more and more users, thanks to the ever-going technical breakthroughs. This propagation of electronic devices tends to complicate interaction paradigms, that leads to loose the users' attention.

In order to make the user focus on his task among a set of devices, new interaction paradigms came as a response, such as the "pervasive computing" (a.k.a. "ubiquitous computing" abbreviated "ubicomp"). Introduced for the first time by Weiser [35], this interaction paradigm describes seamless interactions between the user and his surrounding electronic devices to an extent that the devices' presence is omitted by the user. While "pervasive" and "ubiquitous" literally means "manifesting throughout everything", by speaking of pervasiveness or ubiquity, Weiser is also referring to the seamless aspect of interactions.

#### *Context-Awareness*
A way to achieve pervasive interactions is to tailor systems that are sensitive to the interaction's context. The system then reacts accordingly and proposes services adapted to the user's task in a transparent manner. So called systems are known as "context-aware" systems and were firstly mentioned by Schilit and Theimer [31] while the authorship of the first application of such systems is commonly accorded to Want, Hopper, Falcao and Gib-



In Faure, D., Vanderdonckt, J., (Eds.), *Proc. of 1st Int. Workshop on User Interface Extensible Markup Language UsiXML'2010* (Berlin, 20 June 2010), Thales Research and Technology France, Paris, 2010. ISBN 978-2-9536757-0-2

bons [34], even though their application is limited to a location-awareness. Context-aware systems rely on the capture of contextual information (e.g. user's location and surroundings, time of the day, etc.) and on the system reactions depending on these information to better fit the user's needs or enhance its effectiveness. Working with contexts enables to gather richer information that will unlock new processing possibilities without (or with less) user intervention.

Despite Schilit and Theimer [31] described a context-aware application as being able to "adapt itself according to its location of use, the collection of nearby people and objects, as well as changes to those objects over time", researches were mainly focusing on location as the only relevant contextual element [1] [7]. Besides location-aware application, most researches were domain specific, limiting the usage of contextual information. It is only by the years 2000 that appeared more elaborated definitions of what context-awareness is. Schmidt, Beigl and Gellersen [32] are among the first to have pointed out the limitation of location-aware system. Dey's definition [11] is the one that united the domain's actors and is stated as follow:

"A system is context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task."

Such systems are characterized by the acquisition of contextual data (or sensing), the processing of these data and the applicative reaction to the result of the processing, considering the reaction to be relevant to the user's task [2].

According to Bolchini, Curino, Quintarelli, Schreiber and Tanca [5] or Hong, Suh and Kim [17], one of the common usages of context awareness is the adaptation of user interfaces, i.e. to present interfaces that are well suited to the user's task in his current context in order to enhance usability (e.g. a context-aware mobile phone could present only productivity features to the user if it identifies the user's location as being "at work" or reject any incoming call if the user is attending a meeting).

*User Interface Adaptation*

Interfaces adaptability is not a new subject. During the last years, a lot of studies have been carried out on the subject, but they was mostly oriented on Adaptive Hypermedia Systems (AHS). Brusilovsky works [6] focused on what can be adapted (content and navigation) and on adaptation methods and techniques. Since, these works have evolved and multiplied (especially applied to web sites). Besides the generic context, interface adaptation can be executed at different levels, namely at the user and device ones.

*Interface adaptation according to the device*

This kind of adaptation can be considered as one of the simplest but can be tricky since the interaction paradigms and modes often vary from one device platform to another.

For example, numerous interfaces have to be specifically redesigned to take into account touch interactions on the new batch of touch devices, since they are not completely compatible with mouse driven inputs (*mouseover* events are for example unavailable on touch devices). On the Web, several languages and technologies enable us to adapt user interfaces to the device or user-agent. CC/PP[10] profiles contain some detailed descriptions of the characteristics of a device such as screen size, input devices, etc. These profiles can be manipulated at the web interface behavior layer (i.e. JavaScript) through the use of DCCI[11] interfaces. Another opportunity for graphical adaptation will be based on style layer, using CSS3[12] module "media queries"[13]; which will allow adapting layout and presentation to the technical characteristics of the device.

*Interface adaptation according to the user profile*

One important source of interface adaptation perspectives consists in the study and modeling of the user profiles. It includes his identity information, his preferences and interests as well as his capacities. The user profile can also be useful to define user disability, as an example a visual interface will be adapted into an audio interface for a blind; or some set of colors will be never used for a color-blind. Adaptation work will focus on the different interaction channels now available on mobile devices (e.g. entry: keyboard or tactile screen, vocal command, body movements).

Adaptation to user profile implies modeling the user and profiling it, explicitly by asking for his personal data or preferences, or implicitly. As it relies on an analysis of the user behavior, implicit profiling is more efficient in the sense it is not biased by the inherent uncertainties of human assumptions.

**Usability**

*User-centered design and quality of generated HCI*

Usability oriented development processes offer a considerable advantage to both user's accessibility and from a strictly economic perspective. Many authors have mentioned this in their publications [27]. Although this approach is frequently used in the marketing domain, many

[10] CC/PP, Composite Capabilities/Preference Profiles: Structure and Vocabularies 1.0, W3C Recommendation, 15 Jan. 2004, available on the website: http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/

[11] DCCI, Delivery Context: Client Interfaces 1.0 – Accessing Static and Dynamic Delivery Context Properties, W3C Candidate Recommendation, 21 Dec. 2007, available on the website: http://www.w3.org/TR/DPF/

[12] CSS3, Introduction to CSS3, available on the website: http://www.w3.org/TR/css3-roadmap/

[13] Media Queries, avaimable on the website: http://www.w3.org/TR/css3-mediaqueries/

companies still rely on development processes that have not been adapted to a true user-centered approach. Some studies even focused on a metric approach to the “usability gain” and found cost / benefit ratios oscillating between 1:10 and 1:100 [13].

The ISO 13407 norm [18] (“Human-centered design processes for interactive systems”) presents an approach to interactive system development that focuses specifically on making systems usable. It underlines that the application of human factors and ergonomics to interactive systems design enhances effectiveness and efficiency, improves human working conditions, and counteracts possible adverse of use on human health, safety and performance. Then applying usability to the design of systems involves taking account of human capabilities, skills, limitations and needs.

However as a majority of development-oriented companies seem to remain reluctant to adopt real user-centered development processes and as development time frames tend to get continuously tighter, it is increasingly more important to analyze the degree to which a user-centered approach could be directly integrated into automated development cycles. This perspective seems particularly attractive because it can potentially contribute to conciliating the time constraints in development and the user requirements.

#### *Usability criteria*

Both the conceptual and methodological issues to deal with relate to the “usability criteria”. The criteria are the indicators the usability models are built on; they are used in an operational context to evaluate to what extent the development or the finished IT product is in line with usability requirements. In fact, as shown in our earlier research works [20] the literature offers an important variety of conceptual frameworks around the concept of usability. As a result, the usability criteria appear to be very numerous, diverse and the models seldom agree on the proposed criteria. Differences are frequent and pertain to:

- the number of criteria per model;
- the “abstraction level” of the criteria;
- the ability to operationalize the criteria;
- the inter-criteria orthogonality.

In addition to a very large inter-model diversity, we also showed [20] that the conceptual frameworks in the usability domain suffer from an additional issue: some authors define their models as pertaining only to development situations; other authors would like their model to be applied only to usability evaluations; while a third category do not take into account this potential difference but nevertheless don’t fit both situations. As a result, the overarching conceptual framework lacks consensus and there seem to be no models that deal with all these requirements.

To finish, note that the involvement of users in the development process provides a valuable source of knowledge about the context of use, the tasks, and how users are likely to work with the future system [18]. This contributes to the development and understanding of the models for profiling and personalization described above.

## CASE STUDY: VISUALIZATION SERVICES DESIGN FOR HIGHLY COLLABORATIVE ENVIRONMENTS

Highly collaborative business environments are characteristic of numerous business fields and projects. If collaboration exists inside firms and enterprises, we consider it particularly challenging when collaborative projects involve several firms, for short durations, in unpredictable contexts. In such organizations, business services as well as User Interfaces (i.e. visualization services) face the challenge of rapidly adapting to dynamic business environments.

### Business actors’ requirements in highly collaborative environments

The concept of adaptation of User Interfaces has also to take into account the adaptation to the requirements of business actors regarding visualization, also called usages. That is the aim of the case study presented here.

The usage-centered engineering has been introduced by Constantine and Lookwood [8] to develop software and HCIs that support all the tasks that users have to accomplish. This method suggests a systematic process using abstract models to guide user-interface designers. The usage-centered design is proposed as an alternative to user-centered design, which relies on three main techniques: user studies (to identify the users’ need), rapid prototyping (to get user feedback), and usability testing (to identify usability problems [9]). The difference between usage-centered design and user-centered design [9] is that the focus is not users but usage, i.e. the tasks intended by users and how these are accomplished.

By extension, we consider this approach really powerful to express the requirements based on the real business tasks of actors in a particular targeted domain. In collaborative environments, these business tasks are often closed to the role of actors in an organization, but also to the business services (i.e. IT or non-IT services) supporting a project’s collaboration. Role-specific usages in Architecture, Engineering and Construction (A.E.C.) projects have been defined in previous works [23] and served as inputs for visualization interfaces design.

### The A.E.C. Collaborative Context

The Architecture, Engineering and Construction sector (A.E.C.) is characteristic of such highly collaborative environments. Construction projects involve numerous practitioners, for short durations (project duration), in various contractual contexts (private projects, public-private partnerships etc.). Business practitioners (e.g. public owners, engineers and architects, contractors…) work together and carry out their own methods and models. Therefore in such environments groupware services and the associated User Interfaces have to fit the specific requirements related to 1) the collaborative situation, and 2) the specialized roles involved. Especially we consider as particularly challenging the adaptation of User Interfaces to the particular requirements of the various roles using it.

### Modeling Collaboration activities and support services

A model-based approach has been used in parallel works in order to:

- Represent collaborative situations (i.e. building projects). A metamodel of collaborative activities has been designed [22], and enables instantiating specific models of construction projects;
- Describe services, and especially the ones supporting collaboration (groupware services). A metamodel of service is currently under design and aims at representing services through different viewpoints (Business, service solution and technical views).

### Adaptable Visualization Services

Visualization services implemented in Human-Computer Interfaces of business services have to fit usage of actors who have specific practices according to their role in a collaborative activity. Usages differ from traditional user profiles in context-aware computing by increasing their acceptance with:

- Organizational and operational roles of a project's actor in a particular cooperative situation;
- Particular requirements generated by the available business services in the collective project.

Therefore User Interfaces could be designed according to the usages. They can be identified as "Adaptable Visualization Services" which could be chosen, integrated and used by the actors to perform their activity inside a highly collaborative project [23]. Moreover the approach of AVS has been introduced previously by Dymácek, Hocová and Kintr [14] and aimed at developing visualization services able to be re-used in various contexts, and with different data types.

#### *Design approach*

In such environments the "mapping" between 1) usage, 2) domain characteristics and 3) Business Service attributes on the one side, and 4) Visualization Service attributes on the other side is essential. The value of an AVS proposition highly relies on its alignment to the user requirements (i.e. the usage). Therefore, focusing on AVS design, the main challenge is to map the concepts of usages to the ones of Adaptable Visualization Services such as shown in Figure 1.

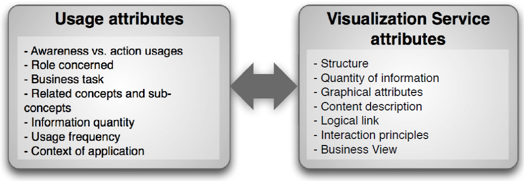


**Figure 1. Mapping between usage and AVS attributes.**

#### *Case study: AVS design*

The case studied here consists of a first experiment aiming at designing an AVS's platform on the basis of usages that we identified with practitioners. This research work helped us to consolidate a design process and led to the development of four AVS integrated in a prototype platform [37].

Usages related to the role of "architect" have been identified. The approach was based on existing practices related to document management in construction projects. We distinguished between two types of usages: the ones related to the understanding of the state of the cooperative context (i.e. awareness usages) and the others related to the specific actions the role has to perform (i.e. action usages). These usages have then been described according to a defined metamodel. A first set of 37 architect-specific usages has been identified. In a second stage we have set-up an experiment protocol in order to validate their usefulness with practitioners. Interviews have been carried out aiming at verifying qualitative aspects of usages (are they really useful) and quantitative aspects (among the 37 usages, what are the preferred ones?).

Once some visualization techniques were described, we tried to find a logical manner to associate them with usages. Attributes of usages were compared to attributes of visualization services. Two usages have been selected, and for each one, two visualization services have been selected.

Two usages have been chosen and four related AVS have been implemented in a prototype platform called E.V.A. (Experimental Visualization Application). Figure 2 shows the E.V.A. interface. A basic selection of a usage and a particular visualization enable the user to test it.

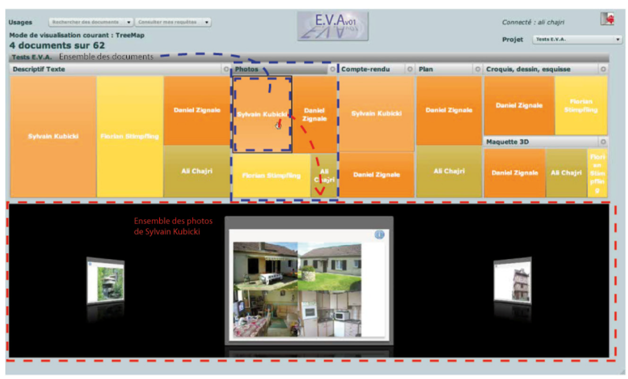

**Figure 2. E.V.A. user interface.**

The final part of this AVS design example is related to the validation of the proposition. As usages were previously validated with practitioners (see above) we had a first assessment of their utility. A second experiment stage led us to test the AVS, once implemented, with users. The process relied on testing tasks performed by users and a final questionnaire/discussion to gather qualitative feedback.

## TOWARDS A GENERATION METHOD

This case study of AVS design in highly collaborative environment (here the A.E.C. industry projects) shows the importance of aligning User Interfaces to the requirements of users in a specific context (comprising both a targeted domain and possibly existing business services). Beyond AVS design based on usages modeling, one can recognize the importance of User Interface (UI) personalization. The following parts describe a proposal for a generic method enabling the generation of personalized UI and based on models.

### A generic method: the Genius method

The method proposed here is based on an iterative process made from 4 successive major steps. This iterative process is inspired from classical methods such as "User-centered design" [18] and agile project management, since the involvement of the users is a key concept. The software framework aims at streamlining the collaboration between users and developers and automate the feedback channel.

The first step is related to the modeling of the business domain in which we wish to support collaboration through services and HCIs. By "business domain" we mean any relation to the tasks of the operator and the context in which these tasks occur. Thus, four main elements are considered:

- The project (tasks, goals, duration, planning...) [11] and the aspects related to the characterization of collective work [26] [28]. This first element is named the "*collaboration model*". It is close to the meta-model of collaborative activities presented in the case study;
- The organization, as a complex social construction, which has an impact on the IT services and HCIs. This element is named the "*organizational model*";
- The services designed for the domain. In this point we consider more precisely the e-services specifically composed to answer to the requirements of a specific collaborative situation (e.g. a construction project). The related model is called the "*service model*" (cf. case study);
- The user profile [2] [30] described in a "*user profile model*".

The second step consists in automatically generating a Human-Computer Interface based on the modeling of the various issues identified during the step 1. The method consists in modeling and implementing the transformations between models necessary to perform. The result of this stage is characterized by an adaptive interface [9].

In our approach this step is enhanced through ergonomic criteria [3] [19] from which the interface is generated. This is one of the fundamental points of this research project. Indeed, this step involves the automatic generation of user interface complying with ergonomic heuristics that can be referenced in the literature. Note also that we consider the ergonomics of human-machine interfaces such as the association of various factors like aesthetics [16] [33], acceptance [36], usability [24], perceived usefulness, and so on. This step thus gives a first set of inputs to the question whether heuristic criteria can easily be adapted in order to guarantee an improved level of usability. It will give us a better understanding of what could be the best way to operationalize these criteria.

The third step of our method allows multiple users to use and test the generated user interfaces. This is an important step in our research because it collects data from a field study. In our case, the field study will be the Architecture, Engineering and Construction sector, as previous UI design process are available (see case study). However the method could be applied to many other cases characterized by highly collaborative activities. The generated application should preferably be used in "real world" conditions in order not to disturb the UI assessment and to foster adoption.

Finally, the fourth stage of our method is consists in a feedback that improves the understanding of models developed in step 1. These feedbacks will enable collecting information on the use of the interface created previously. They will provide valuable information useful in designing future generations, adding to their plasticity. The key part of this stage will be the interpretation of these feedbacks and the mapping of the underlying concepts to the four main models of our framework. Our method is summarized in Figure 3.

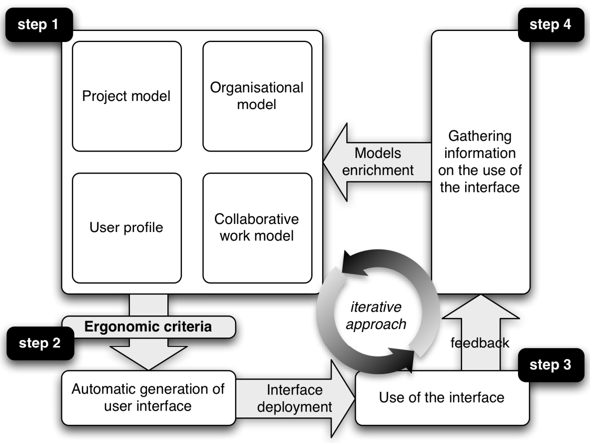


**Figure 3. The Genius method.**

**Integrating UsiXML in the Genius method**

As previously described, to improve the usability of auto-generated UIs, the Genius method considers ergonomic criteria (based on heuristics that can be referenced in the literature) within the model-driven user interface design. Moreover, it defines an iterative approach wherein the first step is focused on the modeling of business domain that impact the automatic generation of UIs. This business domain considers four main elements: the project, the organization, the collective work and the user profile. A part of the modeling of these elements can take advantage of innovative previous works such as UsiXML [37] [38].

UsiXML[14] is a XML-based mark-up language that describes a UI for multiple contexts of use such as character, graphical, auditory or multimodal interfaces. Currently UsiXML only supports interaction device independence, computing platform independence and interaction modality independence[15]. One main goal of user interface description languages as UsiXML, is to handle the complexity, the heterogeneity of final platforms and the different contexts of use. To achieve this, UsiXML provides a high level of abstraction to design UIs, by using a set of models involved in the UI conception. These models are used to provide UI designers with a way, models and tools to design interfaces in order to auto generate the final UI according to the MDE approach [15] [39]. This method aims at separating the design tasks from the development ones. A relationship between the required models in the Genius method and the available models defined in UsiXML can be established. The Domain Model, Task Model, Context Model, Resource Model, Abstract UI Model and Concrete UI Model are identified as potential interesting models to enrich the modeling of business domain defined into Genius. The links between each of these models is handled by classic MDE transformations (e.g. Mapping and Transformation model defined in UsiXML). During the modeling of software and the transformation of models, the Genius method aims at identifying how to detect the potential ergonomic issues to improve the usability of final generated UI.

The scope of UsiXML models is wide but does not consider all aspect of a user interface. In order to help UI designers to reach a better usability level, the Genius project attempts to identify where and how the modeling of ergonomic heuristics and the validation process can take place into the MDE approach. Many UsiXML related works [4] already propose different approaches which can contribute to improve the integration of ergonomics criterion in the Genius method. This is particularly innovative because many studies show the difficulty of systematizing the ergonomic design of interfaces. Moreover, the Genius project considers the UI design within an iterative approach, including user's feedbacks in the final step of iteration. The aim is to provide a basis for statistical analysis related to a number of performance indicators to be identified, and next enhance or adapt the initial modeling of software according to the feedbacks of users and the ergonomic recommendations.

The outcome of the next version of UsiXML will enhance the modeling language through the addition of versatile context-driven capabilities based on µ7 concept[16]. Another high priority in the Genius project is to identify the main characteristics of highly collaborative environments, and see how we can adapt the interfaces to the profile of a user and to his context. So as to better understand this kind of process, by initially considering a large set of contexts, we can extend the research area to determine more characteristics involved in highly collaborative environments.

## CONCLUSION

We presented here the current state of our work on the model-driven generation of ergonomic user interfaces, based on a case study about visualizations services design. We plan to generate usable and context aware user interfaces and refine them through an iterative and semi-automatic process. This type of process could be of interest for the AEC collaborative context, in which it is necessary to design visualization services adapted to their users and to the specificities of each collaborative situation. Our approach is based on model-driven engineering, and we make an extensive use of several models such as project (or domain) model, organizational model, user profile model and collaborative work model. All these models have to be productive and lead us to UI models that will be implemented through UsiXML as the most versatile language for context-aware and model-driven UI design. The usability question will mainly be addressed through two ways: a priori we will constrain our model to the validation of ergonomic criteria formalized as rules, and a posteriori with the study of user behaviors and the adaptation of our models.

Our main objective is now to refine our method and metamodels and to build tools assisting the work of UI designers and business experts in the design of the solution. Then we plan to experiment our main assumptions in real cases of UI design for the AEC sector.

## ACKNOWLEDGMENTS

The research in this paper was carried out within the project GENIUS, funded by the "Fonds National de la Recherche" (Luxembourg). The authors would also like to thank Vincent Koenig (University of Luxembourg) for his contribution in the research project.

[14] www.usixml.org

[15] UsiXML Specification v1.8

[16] ITEA2. Project profile available on the website: www.itea2.org/public/project_leaflets/UsiXML_profile_oct-09.pdf

**REFERENCES**

1. Abowd, G.D., Atkeson, C.G., Hong, J., Long, S., Kooper, R., and Pinkerton, M. Cyberguide: A mobile context-aware tour guide. *Wireless Networks 3*, 5 (2004), 421-433.
2. Baldauf, M., Dustdar, S., and Rosenberg, F. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing 2*, 4 (2007), 263-277.
3. Bastien, J.M.C, Scapin, D.L., and Leulier, C. The ergonomic criteria and the ISO/DIS 9241-10 dialogue principles: a pilot comparison in an evaluation task. *Interacting with Computers 11*, 3 (1999), 299-322.
4. Beirekdar, A., Vanderdonckt, J., and Noirhomme-Fraiture, M. A Framework and a Language for Usability Automatic Evaluation of Web Sites by Static Analysis of HTML Source Code. In *Proc. of 4th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2002* (Valenciennes, 15-17 May 2002). Kluwer Academics Pub., Dordrecht, 2002, pp. 337-348.
5. Bolchini, C., Curino, C.A., Quintarelli, E., Schreiber, F. A., and Tanca, L. A data-oriented survey of context models, *ACM SIGMOD Record 36*, 4 (2007), 19-26.
6. Brusilovsky, P. Adaptive hypermedia: an attempt to analyse and generalize. *International Centre for Scientific and Technical Information, available on the website: http://wwwis.win.tue.nl/ah94/Brusilovsky.html*
7. Cheverst, K., Davies, N., Mitchell, K., Friday, A. and Efstratiou, C. Developing a context-aware electronic tourist guide: some issues and experiences. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (2000), 17-24.
8. Constantine, L.L., and Lockwood, L.A. *Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design*, Inc, Addison-Wesley Educational Publishers, 1999.
9. Constantine, L., and Lockwood, L..A. Usage-centered engineering for Web applications. *IEEE Software* (2002).
10. Coutaz, J., and Calvary, G. *HCI and Software Engineering: Designing for User Interface Plasticity*. In The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, 2008, 1107-1125.
11. Dey, A. K. Understanding and Using Context. *Personal Ubiquitous Computing. 5*, 1 (2001), 4-7.
12. Diaper, D., and Stanton, N. The handbook of task analysis for human-computer interaction. *Lawrence Erlbaum Associates Inc,* 2003.
13. Donahue, G.M., Weinschenk, S., and Nowicki, J. Usability is good business (1999), available on the website: www.compuware.com.
14. Dymácek, T., Hocová, P., and Kintr, M. Adaptable visualization service: through uniformity towards sustainability. *First International Workshop on Model Driven Interoperability for Sustainable Information Systems (MDISIS'08)*, Montpellier, France, June 16, 2008 (2008).
15. Guerrero García, J., Lemaigre, Ch., Vanderdonckt, J., and González Calleros, J.M. Model-Driven Engineering of Workflow User Interfaces. In *Proc. of 7th Int. Conf. on Computer-Aided Design of User Interfaces CADUI'2008* (Albacete, 11-13 June 2008), Springer, Berlin, 2009, pp. 9-22.
16. Hasan, B., and Ahmed, M.U. Effects of interface style on user perceptions and behavioral intention to use computer systems. *Computers in Human Behavior 23*, 6 (2007), 3025-3037.
17. Hong, J.Y., Suh, E.H., and Kim, S.J. Context-aware systems: A literature review and classification. *Expert Systems with Applications 36*, 4 (2009), 8509-8522.
18. ISO 13407. *Human-centred design processes for interactive systems. International Standards for Business, Government and Society*.
19. Jacko, J. *The human-computer interaction handbook*. Lawrence Erlbaum Associates Inc, 2002.
20. Koenig, V. Contribution à l'étude de l'utilisabilité dans le contexte des systèmes d'information à usage professionnel : conception d'un laboratoire d'utilisabilité et applications. Thèse de doctorat en sciences psychologiques, Université Libre de Bruxelles, Belgique, 2006.
21. Koenig, V., and van de Leemput, C. U-Lab: an out-of-the-box solution for any usability problem? A presentation on methodological and practical issues. *Annual Meeting of the Belgian Psychological Society*, Ronéo, Ghent (2005).
22. Kubicki, S., Bignon, J.C., Halin, G., and Humbert, P. Assistance to building construction coordination. Towards a multi-view cooperative platform. *ITcon Electronic Journal of Information Technology in Construction 11* (2006), 565-586.
23. Kubicki, S., and Halin, G. Usage-centered design of adaptable visualization services. Application to cooperation support services system in the AEC sector. *IESS 1.0. First International Conference on Exploring Services Sciences*, Geneva, Switzerland, 17-19 February 2010, Lecture Notes in Business Information Processing n° 53, (2010).
24. Malhotra, Y., and Galletta, D.F. Building systems that users want to use. *Communications of the ACM 47*, 12 (2004), 88-94.
25. Mayhew, D.J., and Bias, R.G. Cost-justifying web usability, in Ratner J. (Ed), *Human factors and web development. Second Edition* (pp. 63-87), USA: Lawrence Erlbaum Associates, 2003.

26. Newlands, A., Anderson, A.H., and Mullin, J. Adapting communicative strategies to computer-mediated communication: an analysis of task performance and dialogue structure. *Applied Cognitive Psychology 17* (2003), 325-348.

27. Nielsen, J.. *Usability Engineering*. Academic Press, San Diego, CA, 1993.

28. Pinelle, D., Gutwin, C., and Greenberg, S. Task Analysis for groupware usability evaluation: modelling shared-workspace tasks with the mechanics of collaboration. *ACM Transactions on Human Computer Interaction 10*, 4 (2003), 281-311.

29. Raijmakers, B. Usability-Tests ist ein Mittel, kein Ziel, in Beier M. and von Gizycki V. (Eds.). *Usability, 3th edition*, (pp.129-157), Berlin and Heidelberg: Springer, 2002.

30. Rich, E. User Modelling via Stereotypes. *Cognitive Science 3*, 4 (1979), 329-354.

31. Schilit, B., and Theimer, M. Disseminating active map information to mobile hosts. *IEEE 8*, 5 (1994), 22-32.

32. Schmidt, A., Beigl, M., and Gellersen, H.W. There is more to context than location. *Computers & Graphics 23*, 6 (1999), 893-901.

33. Tractinsky, N., Cokhavi, A., Kirschenbaum, M., and Sharfi, T. Evaluating the consistency of immediate aesthetic perceptions of web pages. *International Journal of Man-Machine Studies 64*, 11 (2006), 1071-1083.

34. Want, R., Hopper, A., Falcao, V., and Gibbons, J. The active badge location system. *ACM Transactions on Information Systems 10*, 1 (1992), 91-102.

35. Weiser, M. The computer for the 21st Century. *ACM SIGMOBILE Mobile Computing and Communications Review 3*, 3 (1999), 1559-1562.

36. Wu, J.H., Chen, Y.C., and Lin, L.M. Empirical evaluation of the revised end user computing acceptance model. *Computers in Human Behavior 23*, 1 (2007), 162-174.

47. Limbourg, Q., Vanderdonckt, J., Michotte, B., Bouillon, L., and Lopez, V. UsiXML: a Language Supporting Multi-Path Development of User Interfaces. In *Proc. of 9th IFIP Working Conference on Engineering for Human-Computer Interaction jointly with 11th Int. Workshop on Design, Specification, and Verification of Interactive Systems EHCI-DSVIS'2004* (Hamburg, July 11-13, 2004). Lecture Notes in Computer Science, Vol. 3425, Springer-Verlag, Berlin, 2005, pp. 200-220.

37. Vanderdonckt, J. A MDA-Compliant Environment for Developing User Interfaces of Information Systems. In *Proc. of 17th Conf. on Advanced Information Systems Engineering CAiSE'05 (Porto, 13-17 June 2005),* O. Pastor & J. Falcão e Cunha (eds.), Lecture Notes in Computer Science, Vol. 3520, Springer-Verlag, Berlin, 2005, pp. 16-31.

38. Vanderdonckt, J., Model-Driven Engineering of User Interfaces: Promises, Successes, and Failures. *Proc. of 5th Annual Romanian Conf. on Human-Computer Interaction ROCHI'2008 (Iasi, 18-19 September 2008), S. Buraga, I. Juvina (Eds.), Matrix ROM,* Bucarest, 2008, pp. 1-10. ISSN 1843-4460.