QUIMERA: A Quality Metamodel to Improve Design Rationale

Alfonso García Frey, Eric Céret, Sophie Dupuy-Chessa and Gaëlle Calvary
University of Grenoble, Grenoble INP, CNRS, LIG
385, avenue de la Bibliothèque, 38400, Saint-Martin d'Hères, France
{Alfonso.Garcia-Frey, Eric.Céret, Sophie.Dupuy, Gaelle.Calvary}@imag.fr

ABSTRACT
With the increasing complexity of User Interfaces (UI) it is more and more necessary to make users understand the UI. We promote a Model-Driven approach to improve the perceived quality through an explicit and observable design rationale. The design rationale is the logical reasons given to justify a designed artifact. The design decisions are not taken arbitrarily, but following some criteria. We propose a Quality Metamodel to justify these decisions along a Model-Driven Engineering approach.

Author Keywords
User Interfaces, Perceived quality, Quality Metamodel, Design Rationale, QOC, Self-Explanation, Model-Driven Engineering.

ACM Classification Keywords
H.5.2 User Interfaces: Theory and method.

General Terms
Design, Human Factors.

INTRODUCTION
User Interfaces (UIs) must deal with new features such as the capacity of adaptation to the context of use (<user, platform, environment>). As designers cannot anticipate all the contexts of use at design time, UIs are generated dynamically giving rise to lacks of quality. This lack can be overcome through explanations. Self-Explanatory UIs (SE-UIs) aim at answering end-user questions about the UI. One of the SE-UIs approaches [16] is based on Model-Driven Engineering (MDE): explanations are generated from design models such as the Task and Domain Models used for UI generation. Good explanations about the UI need additional crucial information such as justification of design decisions or quality measures of the UI. Thus, we need an argumentation model to convey this information. This paper proposes a solution for explaining design decisions through quality models in the context of SE-UIs. The proposition is illustrated on a Seats Booking System.

EICS'11, June 13–16, 2011, Pisa, Italy.

The paper is fourfold. In the first section, it provides a short vision of related works about Quality and Design Rationale. Then, the Quality Metamodel is introduced and depicted through an example. Third part deals with design rationale. Finally, the fourth part is devoted to the case study that combines all the necessary pieces for self-explanation: a quality model, a design rationale, and a UI through a MDE approach.

RELATED WORKS
As we need quality models to explain design decisions, we relate existing quality models in the first section. Then, we review some design rationale representations explaining which one we use and why.

Quality Models
Different quality models have been proposed in the literature. McCall's hierarchical quality model [12] focuses on product quality, organizing it in two views: the external view for end-users and the internal view for developers. Boehm's model [13] adds a third level named *primitive characteristics* to deal with metrics and evaluation. The ISO/IEC 9126 standard series divides metrics into internal, external and quality-in-use.

This quality-in-use, also called usability or perceived quality, has been the main focus of the HCI community. Usability has evolved through standards such as the ISO 9241-110 [9], ISO/IEC 9126-1 [10] and ISO/IEC 25010 [11] among others. As a synthesis, Seffah encompasses most of the usability works in QUIM [14].

However Software Engineering quality models are more than usability. They deal with other important aspects of general quality in the whole System Development Life Cycle. ISO standards deal also with these aspects. To cover them, different quality metamodels have been proposed such as [18] for data quality, [19] as a quality metamodel for MDE, or [20] that defines a five step process for building product-specific quality models.

However, whilst several quality models exist in Software Engineering, most of them are oriented to evaluating source code or final products and not models or modeling activities. Other models don't deal with evaluation aspects

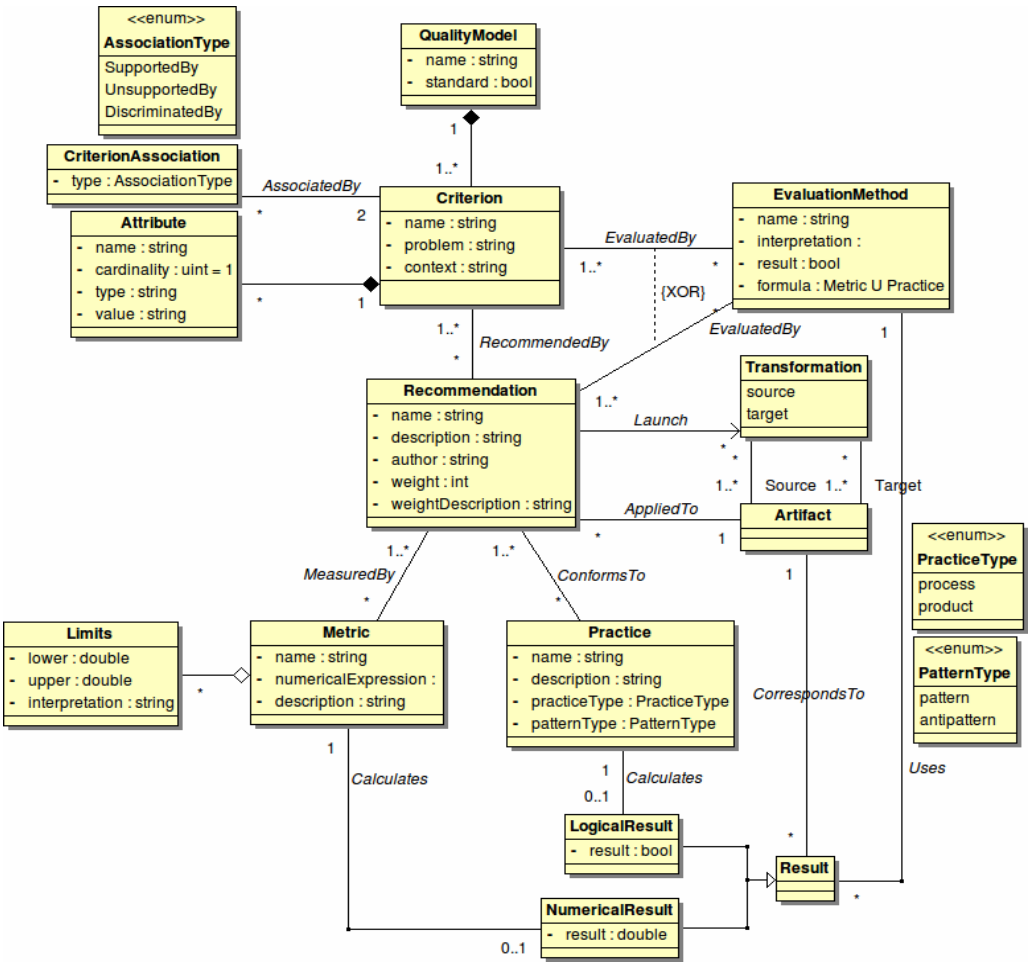

Figure 1: The Quality Metamodel

(evaluation methods, results...) or they just miss the different quality perspectives, as elicited in section two.

We propose QUIMERA, a quality metamodel, to take benefit from both HCI and Software Engineering while covering their requirements as well as our needs for UI quality. QUIMERA will be used to support design rational.

Design Rationale

Design Rationale is defined in [3] as: "An explanation of why a designed artifact (or some feature of an artifact) is the way it is."

Design rationale approaches like PHI [5] or QOC [6] (for Questions, Options and Criteria) are metamodel based. Non metamodel based approaches exist such as PDR [8]. For our purposes, we reuse QOC because it is the more expressive design rationale representation to work with different alternatives at the same time. QOC focuses directly on the discussion between those alternatives as shown in figure 4.

QUIMERA: THE QUALITY METAMODEL

In order to use a quality model to justify design decisions, we introduce a Quality Metamodel from which a quality model can be instantiated. QUIMERA stands for "QUality metamodel to IMprove the dEsign RAtionale". The first section shows the relations between QUIMERA and the system under study (SUS). The second section explains QUIMERA in detail. The final section shows an instantiation of QUIMERA that is put in practice later in the case study.

Quality Perspectives

QUIMERA (figures 1 and 2) has been designed to cover the needs of both Software Engineering and HCI. Quality can be expressed regarding four different perspectives [1]:

Expected Quality, or the quality the client needs. It is defined through the specification of the SUS.

Wished Quality is the degree of quality that the quality expert wants to achieve for the final version of the SUS. It is derived from the Expected Quality.

Achieved Quality is the quality obtained for a given implementation of the SUS. Ideally, it must satisfy the Wished Quality.

Perceived Quality is the perception of the results by the client, once the SUS has been delivered.

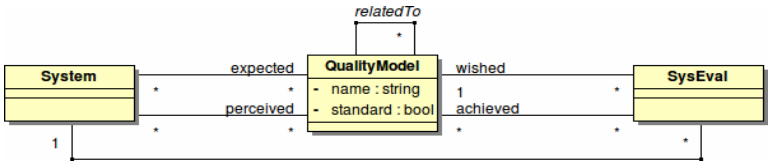

Figure 2: Quality perspectives in the Quality Metamodel

As stated in [2], these four perspectives can be related to the Systems Development Life Cycle by three dimensions. These dimensions are the Specification (related to the Expected and Wished Qualities), Implementation (related to the Achieved Quality) and Use (related to the Perceived Quality).

QUIMERA deals with these four perspectives as shown in figure 2. Here, the *System* entity represents the product to consider. *SysEval* represents a specific evaluation for that product. The four quality perspectives are four different uses of the same quality model. The attribute *standard* means that, when true, the quality model is not linked to *System* and *SysEval* as it only represents a quality standard such as ISO9241-110 or QUIM. In other words, the quality of these standards is not defined in terms of a product. Some internal parts of QUIMERA are not necessarily defined when *standard* is true.

Once the standard has been set, QUIMERA can be extended with the classes that are needed for each quality perspective, as we will see in the next section.

The Metamodel

Figure 1 shows QUIMERA in detail. A quality model is composed of criteria, that can be recursively decomposed into subcriteria through the class *CriterionAssociation*. Different recommendations can be specified for each criterion. A *Recommendation* is a positive assessment that characterizes Criteria. We can specify a weight for each recommendation to define which of them are more important than others for the considered system. Evaluations can be performed through *EvaluationMethods* that are specified by *Metrics* and/or *Practices*. In the first case, the measure is given by a *NumericalResult* that can be comprised between some *Limits* when defined. In the case of *Practices*, the result is a logical value, true or false, indicating if the *Practice* has been followed or not. Note that a *Practice* can be either a *pattern* or an *anti-pattern*, applied at the *process* level, or on a *product*. *Metrics* and *Practices* are directly evaluated on *Artifacts* through *Recommendations*. An *Artifact* can be no matter what element of the Software Development Life Cycle, such as code, classes of a model or the model itself.

Once a quality standard has been defined through *Criteria*, the metamodel can be reused with the association *relatedTo*, and extended with several classes such as *EvaluationMethods*, *Transformations* or *Artifacts,* to represent the four quality perspectives. For instance, *Metrics* can be defined in order to obtain some desired values (*Wished Quality*). The importance of every *Recommendation* can be customized using *Weights*. This allows designers to adjust the global quality precisely. Then, evaluations of the current quality of the SUS can be performed. When a *Result* of the evaluation (*Numerical* from *Metrics* or *Logical* from *Practices*) does not satisfy the expectations of the quality expert, this is, the *Achieved Quality* does not satisfy the *Wished Quality* (for instance, the value for a metric is not within the desired *Limits)*, the designer will need to increase the quality. This can be done by setting a *Transformation* or a set of *Transformations*. These *Transformations* are performed on the related *Artifacts* on which the *Result* has been previously calculated. Iterations can be done until the desired values defined by the quality expert (*Wished Quality*) are reached.

A Quality Model for the Ergonomic Criteria of UIs

Figure 3 shows a quality model representing Ergonomic Criteria in HCI [9]. For the sake of brevity, we explain only the three of them that are used later in the example:

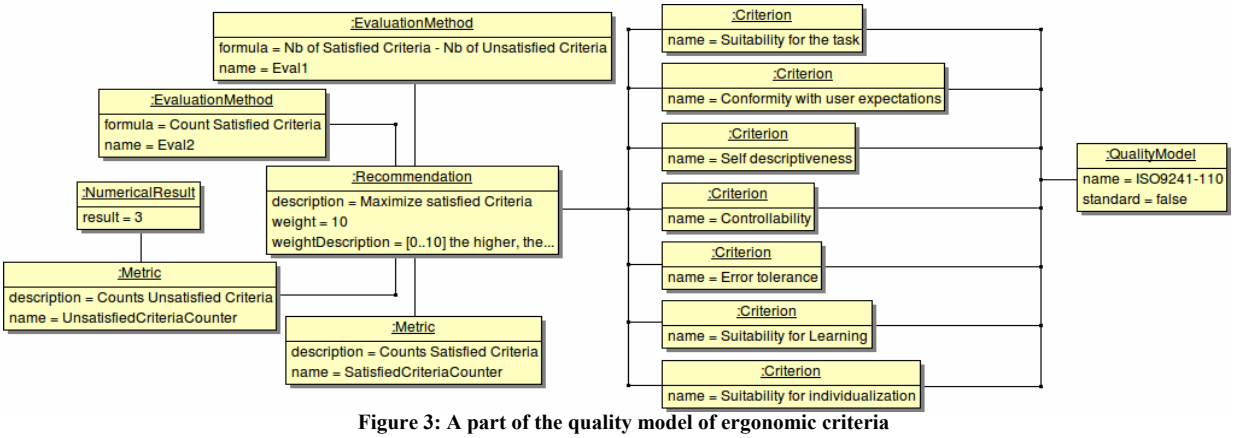

Figure 3: A part of the quality model of ergonomic criteria

Suitability for the task: A dialog is suitable for the task if the dialog helps the user to complete her/his task in an effective and efficient manner.

Self descriptiveness: A dialog is self descriptive if every single dialog step can immediately be understood by the user based on the information displayed by the system.

Error tolerance: A dialog is fault tolerant if a task can be completed without erroneous inputs with minimal overhead for corrections by the human user.

A *Recommendation* is a positive assessment that corresponds to one or more criteria. Figure 3 shows how different metrics are used for the same recommendation. For instance, in figure 3, the recommendation says that good quality can be achieved by maximizing the number of criteria that are satisfied by the UI. To evaluate Criteria, two different *EvaluationMethods* are defined. A detailed explanation about the left part of figure 3 including the *Recommendation*, the *Metrics*, the *EvaluationMethods* and the *Result*, is given later in the case study.

DESIGN RATIONALE

The main objective of QOC is the discussion of alternatives on specific artifact features. For our purposes, we consider only the following QOC elements:

Options that are artifact features under discussion.

Questions that are means of organizing the various Options, since every artifact feature responds to a specific design issue that can be framed as a Question.

Criteria that are used to determine the choice between Options. Equivalently, they can be seen as requirements or goals that have to be accomplished.

Assessments are links between Options and Criteria. If they satisfy a Criterion then the link is represented with a normal line. If not, a dotted line is used.

Figure 4 shows an example of QOC in which designers propose several interactors to let the user enter a date. The first interactor is composed of three input fields for the day, month and year respectively, and a label indicating format notations. The second interactor is a calendar. As shown in figure 4, the first interactor does satisfy the three criteria whilst the first interactor does not. This

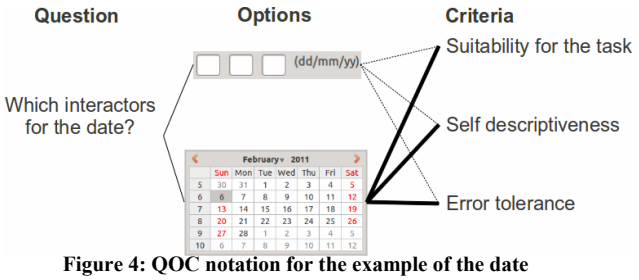

Figure 4: QOC notation for the example of the date

example is taken from the case study depicted in the next section.

PUTTING THE PIECES TOGETHER: A CASE STUDY

Figure 5 shows a booking system dialog inspired from [17]. With this UI, the end-user can book seats for a cinema session by entering the name, address, date of the session, time of the session (morning or evening) and the desired number of seats. The dialog has been derived in a MDE process from the task model shown in the same figure. The connection of various artifacts such as prototypes and tasks has also been proposed by previous authors [4,7]. The information provided in this UI is not clear enough. Some of its main problems are:

1. The prompting is insufficient. For instance, the label *Name* stands for First and Last names. (*Self descriptiveness*)
2. The guidance is ambiguous. Should the user type a ',' between names? (*Self descriptiveness*)
3. There is no prevention against errors. Users can enter any value because the verification is done in a later step. (*Error tolerance*)

This particular design has also two negative implications:

1. In case there is no seat available, the end-user has entered useless information.
2. If the end-user needs to book several seats at different times, for instance one in the morning and one in the evening, then the end-user needs to enter the same data several times.

A good design should ask for the information related to the seats first, and only if there are enough seats available, ask the end-user to provide the personal information.

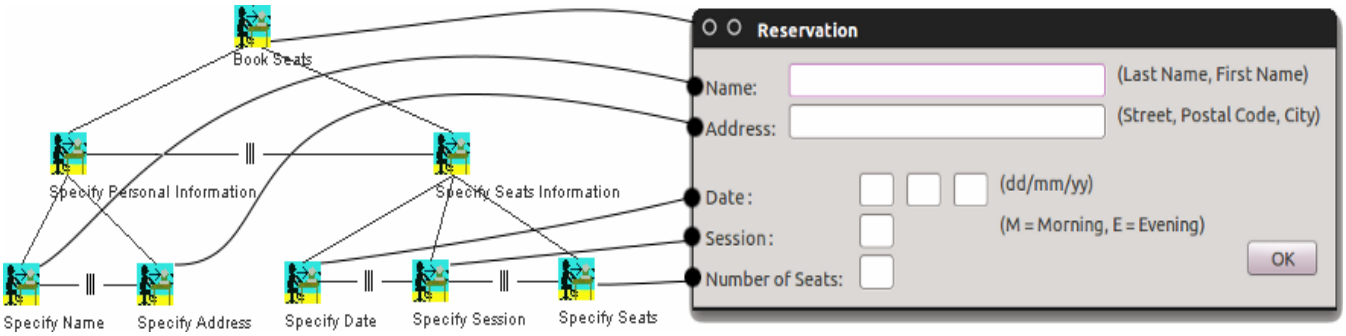

Figure 5: The UI of a Seats Booking System is obtained from a Task Model through a MDE process

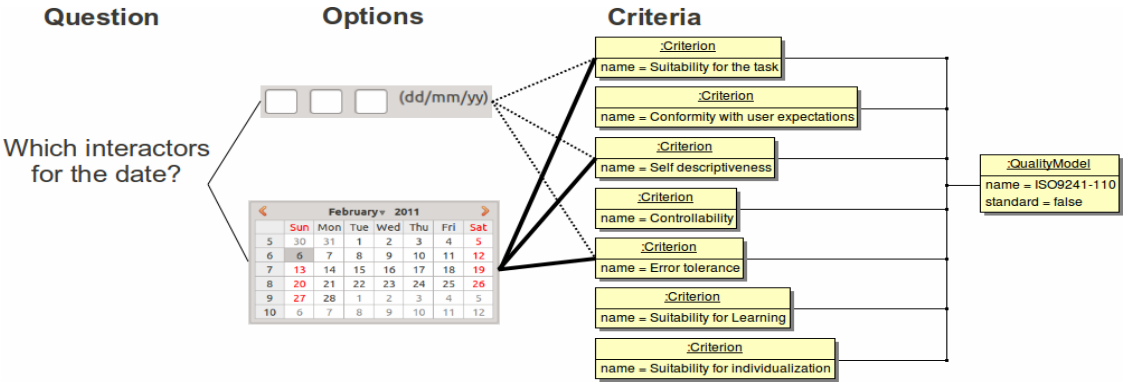

Figure 6: The quality model is used as the Criteria of QOC

In order to explain design decisions through quality models, designers use our approach as follows. First, the designers define a quality model based on QUIMERA. For this case study, we consider the quality model of ergonomic criteria of figure 3. Once the quality model is set, designers can keep trace of the design rationale through QOC. They need to describe the design rationale through questions, options and criteria. Designers write down the necessary questions to cover all the precedent problems that they have identified on the UI in figure 5, in the same way as it has been done in figure 4 for the question *Which interactor for the date?*

Our proposition is to use the quality model as the Criteria when describing the design rationale with QOC. Figure 6 shows this principle. With this approach, the three problems listed before about *Self descriptiveness* and *Error tolerance* are directly related to quality through the quality model. Note that the quality model is not a merely representation of the ergonomic criteria from [9], as it has been shown in other works like [15,21]. Ergonomic criteria play different roles becoming active for each quality perspective. For instance, the UI in figure 7 is better than UI in figure 5. The comparison between both UIs is based on *EvaluationMethods* depicted in the left part of figure 3. These methods use the specific formulas: "Satisfied Criteria minus Unsatisfied Criteria" for Eval1, and "Number of Satisfied Criteria" for Eval2". For Eval1 and regarding figure 6, we have Eval1(Calendar) = 3 - 0 = 3 and Eval1(Text-Fields) = 3 - 3 = 0, showing that the Calendar is better (3>0). The same conclusion is obtained for Eval2.

Advantages

The main advantages of this approach are:

1. Quality in design decisions becomes measurable.
2. Design decisions can be explained directly through quality models.
3. As a design rationale can be directly evaluated, two different solutions can be compared.
4. The quality model can be used not only for evaluation purposes, but as an active agent of the design rationale and the MDE process. As QUIMERA can launch transformations if the desired quality is not achieved, the MDE process for generating UIs can take benefit of it regarding how a transformation increases or decreases the achieved quality. For instance, figure 7 shows an improved version of the Seats Booking System. In this figure, two UIs have been generated to avoid the problem of typing personal information when there are no seats available. Here, the Task Model has been transformed (operator >>) and two UIs are generated now, maximizing the criterion *Suitability for the task*. Note that in figure 7, the task *Specify Name* is transformed into two sets of

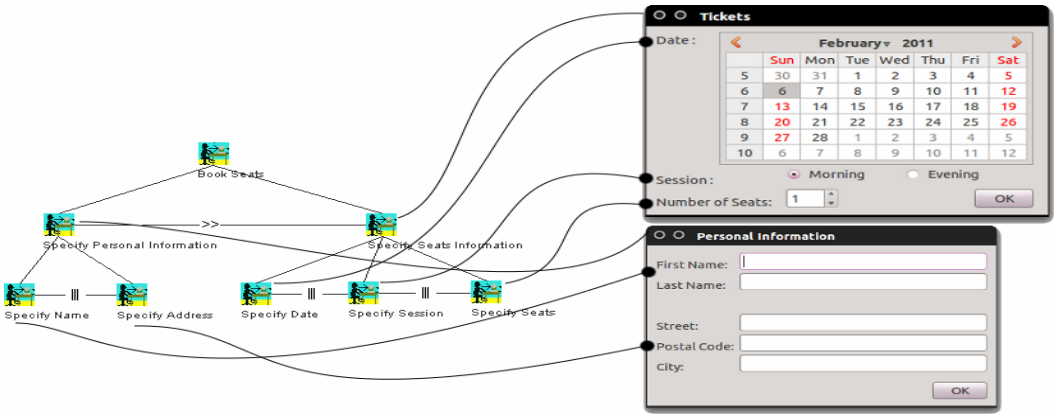

Figure 7: Two UIs are derived from the Task Model

Label + Text-Fields based on the two concepts (First name, Last name) that are manipulated in the task.

5. As a consequence of the previous point, adaptation of UIs can be quality driven.

Following this approach, designers can easily quantify design decisions regarding quality, and quality standards become active agents of the design process.

CONCLUSION AND FUTURE WORK

This paper presents QUIMERA, a quality metamodel that unifies quality aspects from HCI and Software Engineering, setting the bases for a quality driven adaption of UIs through quality models. Although QUIMERA is used to explain design decisions through quality models, it is domain independent, i.e. not only devoted to HCI.

We have detailed our approach through a case study, in which the metamodel is instantiated first, and used later as an active agent of the design rationale. The main advantages of this approach have been listed.

Future work will focus on implementing the proposed approach for evaluation purposes.

ACKNOWLEDGMENTS

The work is funded by the european ITEA UsiXML project (2009-2012).

REFERENCES

1.Carlier A. Management de la qualité pour la maîtrise du SI, Paris, Hermès, p. 28, 2006.

2.Si-Saïd Cherfi S., Akoka J., Comyn-Wattiau I. Conceptual Modeling Quality - From EER to UML Schemas Evaluation, Lecture Notes in Computer Science, Vol. 2503, p 414-428, January 2002.

3.Moran, T. P. and Carroll, J. M. Overview of design rationale. In Design Rationale: Concepts, Techniques, and Use, T. P. Moran and J. M. Carroll, Eds. LEA computers, cognition, and work series. Lawrence Erlbaum Associates, Inc., Mahwah, NJ, p 1–19, 1996.

4.Palanque P. and Lacaze, X. DREAM-TEAM: A Tool and a Notation Supporting Exploration of Options and Traceability of Choices for Safety Critical Interactive Systems. In Proceedings of INTERACT 2007, Rio, Brazil, Lecture Notes in Computer Science, Springer Verlag, September 2007.

5.McCall, R. J. PHI: A conceptual foundation for design hypermedia. Des. Stud. 12, 1, p 30 – 41, 1991.

6.MacLean, A., Young, R. M., Bellotti, V. M. E., and Moran, T. P. Questions, options, and criteria: Elements of design space analysis. Human-Comput. Interact. 6, 3-4, p 201–250, 1991.

7.Bramwell, C. Formal Development Methods for Interactive System: Combining Interactors and Design Ratio- nale. Ph.D. Thesis. University of York. 1995.

8.Carroll, J. M. and Rosson, M. B. Getting around the task-artifact cycle: How to make claims and design by scenario. ACM Trans. Inf. Syst. 10, 2 (Apr.), p 181–212, 1991.

9.ISO 9241-110:Ergonomics of human-system interaction - Part 110: Dialogue principles. ISO, 2006.

10. ISO/IEC 9126-1: Software engineering. Product quality - Part 1: Quality model. ISO, 2001.

11. ISO/IEC CD 25010.3: Systems and software engineering - Software product Quality Requirements and Evaluation (SQuaRE) - Software product quality and system quality in use models. ISO, 2009.

12. McCall, J. A., Richards, P. K., and Walters, G. F. Factors in Software Quality, Nat'l Tech. Information Service, no. Vol. 1, 2 and 3, 1977.

13. Boehm, B. W., Brown, J. R., Kaspar, H., Lipow, M., McLeod, G., and Merritt, M. Characteristics of Software Quality, North Holland, 1978.

14. Seffah, A., Donyaee, M., Kline, R. and Padda, H. Usability measurement and metrics: A consolidated model. Software Quality Journal, 14(2), p 159–178, June 2006.

15. Lacaze, X., Palanque, P., Barboni, E., Bastide, R., Navarre, D. From DREAM to Realitiy: Specificities of Interactive Systems Development with respect to Rationale Management. In: Dutoit, A.H., McCall, R., Mistrik, I., Paech, B. (eds.) Rationale Management in Software Engineering, pp. 155–172. Springer, Heidelberg, 2006.

16. García Frey, A. Self-explanatory user interfaces by model-driven engineering. In Proceedings of the 2nd ACM SIGCHI symposium on Engineering interactive computing systems (EICS '10). ACM, New York, NY, USA, p 341-344, 2010.

17. Nogier, J.F. De l'ergonomie du logiciel au design des sites Web, Third edition, Dunod 2005.

18. Kashif M, Si-Saïd Cherfi S., Comyn-Wattiau I. Data Quality Through Conceptual Model Quality-Reconciling Researchers and Practitioners Through a Customizable Quality Model. In International Conference on Information Quality (ICIQ), 2009.

19. Mohagheghi, P. and Dehlen, V. A Metamodel for Specifying Quality Models in Model-Driven Engineering. Nordic Workshop on Model Driven Engineering NW-MoDE '08, Reykjavik Iceland, p 20-22, August 2008.

20. Dromey, R.G. Concerning the Chimera. IEEE Software 13 (1), p 33- 43, 1996.

21. Martinie De Almeida, C., Ladry, J.F., Navarre D., Palanque P., Winckler, M. A. Embedding Requirements in Design Rationale to Deal Explicitly with User eXperience and Usability in an “intensive” Model-Based Development Approach (regular paper). In: Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI 2010), Atlanta Georgia USA, Vol. 617, (Eds.), CEUR Workshop Proceedings, p. 29-32, 2010.