A “call for answers” around the proposition of an HCI-enriched model

Christophe Kolski

A version of this paper has been published in : *ACM SIGSOFT Software Engineering Notes*, 23 (3), pp. 93-96, 1998.

A "Call for Answers" around the proposition of an HCI-enriched model

Christophe KOLSKI
LAMIH - URA CNRS 1775, University of Valenciennes,
Le Mont Houy - B.P. 311 - 59304 Valenciennes Cedex, FRANCE
E-mail:kolski@univ-valenciennes.fr

Abstract. The engineering of Human-Computer Interface (HCI) is a wide-ranging and huge research field. However development models stemming from Software Engineering overlook important aspects in terms of interactive systems development. That is why an HCI-enriched model, called ∇ model (pronounced nabla model), is envisaged in this paper.

Keywords. Software Engineering, development model, HCI-enriched development model, Interactive system, Human-computer interface, ∇ model

Introduction

First, I want to point out again that the main development models stemming from Software Engineering (Waterfall, Spiral, V...) are too general and largely insufficient in terms of interactive systems development, particularly for the applications in which human errors inherent in using the human-computer interface(s) can lead to safety problems, and/or ecological and/or economic consequences [14] [18] [21] [22]; the first part of this paper consists in a brief critical study of the main models issued from Software Engineering, in term of interactive systems development.

A solution consists in proposing so-called HCI-enriched models. Such a model is proposed in the second part. It is called ∇ (pronounced nabla model). This model integrates the standard steps found in the traditional life-cycle models stemming from Software Engineering, but it is also intended for the development of interactive systems. The development of HCI requires specific skills and means; that is the reason why the aspects linked to HCI will clearly appear in the □ model, contrary to the waterfall, V and spiral models (and their variants).

Finally, in the conclusion, I ask some questions about the validity and the usability of such a model. These questions should be considered as a "Call for Answers" for the readers of Software Engineering Notes.

Global critical study of the best known development models

The three most widely used models at the present time for software development are waterfall, V and spiral models (and their variants). Needless to say they will be just briefly described in so far as they have been often detailed in many Software Engineering books ([16] [28] [30]...).

The well-known Waterfall model [4] is currently certainly the most widespread in companies. In terms of HCI development, a fundamental criticism can be expressed: proposed at the end of the seventies, when taking into account the human-machine interactions was in fact relegated to a position of second importance [26], it is not surprising to notice that these aspects are not emphasized within the system as a whole, even if the latter has a strong interactive component. What is more, apart from the first step where requirement-analysis must logically concern the users, they are not considered any longer in the next steps where more designers are implied. However it is possible to find implicitly the users in the final steps, and the evaluation of the end-product is made possible only once it has been delivered to the users. What is more, even if the software must be used for very complex tasks, no task analysis and modeling is recommended. As a confirmation of this remark, the user analysis and modeling is not alluded to. In fact, whether these very important notions are (implicitly) considered, during the first step, will depend upon the common sense of the most skilled designers, in a very informal way.

The V model [12], figure 1, is often used in France and praised by Quality organisms. The steps presented in the waterfall model remain roughly valid for the V model. Meanwhile specification and design stages are integrated in a descending approach, where tests and validation steps are located in an ascending approach. So, in each descending stage, the planning, means and method allowing to (technically)assess and validate it must to be foreseen beforehand. This preoccupation to foresee the system evaluation as far upstream as possible, considering each stage, represents a strength of the V model.

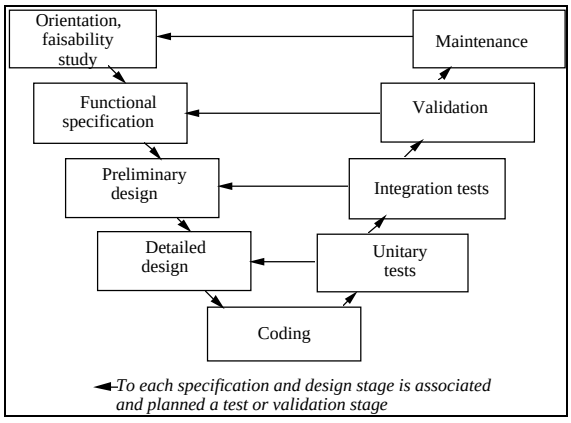

Figure 1. V model

Despite this strength, and like in the case of the waterfall model, criticisms still remain when the objective is the development of a system with a strong interactive component: taking into account the human factors is not - implicitly or explicitly - located. Nevertheless, on account of the importance granted to (technical) evaluation in this model, the user(s) can and must play a more determining part than in the waterfall model. In conclusion, this model must also be refined if the development concerns interactive systems.

The well-known Spiral model has been proposed by Boehm et al. [5]. Unlike the first two models, it is based on the notion of risk analysis .It has an advantage in respect of the development of very

A version of this paper has been published in : *ACM SIGSOFT Software Engineering Notes*, 23 (3), pp. 93-96, 1998.

interactive software: it helps to define specifications which must not be so exhaustive as in the two other models. Indeed the requirements are progressively expressed, and the different risks are solved as soon as they are met. It is only when the risks have been stabilized that one proceeds to a more closed stage, with the effect of delaying the detailed setting-up of low-risk software-components as long as the high-risk items have not been solved yet. This is why numerous authors concerned by the development of interactive software systems, such as James [15], Nielsen [22], Lichter et al. [20] all agree to think that prototyping provides a very practical solution to avoid or at least to limit design deficiencies and ergonomic errors, in order to improve the quality of the products and cut the production, tuning and maintenance costs. The Spiral model immediately introduces, right at the beginning of the cycle, the possibility to evaluate the first choices from a first prototype; it thus gives us a glimpse of extremely promising perspectives for the development of interactive systems. However, this model is all the same open to criticism: like in the case of the two other models, the HCI are not alluded to even if they are implied in the approach.

Considering the development of strongly interactive systems, some gaps appear in the main development models (and their variants) issued from Software Engineering. A possible solution could consist in proposing so-called "HCI-enriched models".

Proposition of a HCI-enriched model: ∇ model

The notion of HCI-enriched model is not new. Many discussions and ideas concerning this notion are found in [3], [8], [9], [11], [17], [19] or [23]. But no real and new model considered as a "Software Engineering model" has been really proposed.

In this part, a model, referred to as ∇ (pronounced "nabla") model, is presented, in connection with interactive system development. Its first version is given in figure 2. Its look is inspired by the V model. Its objective is to locate the several steps necessary to develop an interactive system, by distinguishing the HCI strictly speaking (left part of the model) and the application module(s)eventually accessible from the HCI (right part). One of the outstanding characteristics of the model is to locate stages - nonexistent in the standard models derived from Software Engineering- where the human factors have to be considered by the development team.

The description of the model is as follows. The first step is quite common in Software Engineering, and marks the beginning of the project by giving an orientation to the work to realize (objectives, project organization, constraints, and so on).

Then, the model emphasizes the importance of the analysis of the whole human-machine system during the project; this analysis deals more particularly with the system, the human tasks and the user(s). Indeed user analysis and modeling on the one hand, and human tasks analysis and modeling on the other hand are closely linked .Indeed, the users' cognitive and physical limits and resources have a direct influence over their efficiency and reliability as regards the tasks to perform. The importance granted to task analysis and modeling keeps increasing, and it has led to active researches since the eighties in computer science and in the cognitive sciences, with the goal of interactive systems development [6] [7] [10]. Several methods are usable for the modeling of the different tasks to perform by the user(s), and to make the requirements analysis and the specification of HCI and assistance tools easier. Amongst the most famous examples, we find GOMS [6], the method based on SADT and Petri Nets [1], DIANE[2], TAG [24], and so on.

Modeling must be slanted towards:

- A real model corresponding to the current (existing or virtual) human-machine system, with its constraints, its strengths and weaknesses. Three cases can be considered. When the objective consists in studying an already existing human-machine system to ultimately end with a new one, the modeling is of course carried out from the existing system. When the objective consists in creating a new human-machine system from other already existing systems, the modeling is based on a synthesis of the data issued from each analysis. When there is NO previously existing human-machine system and when the system is entirely to be designed from scratch, this model needs to be designed.
- A reference (or ideal) model corresponding to those of a human-machine system considered as ideal, by considering all the points of view and requirements of the different human partners concerned by the planned human-machine system. This model must in particular lists a set of criteria which have to be abided by. The nature of these criteria can be extremely wide-ranging (safety of human beings, of the facilities, of the environment, production, software ergonomics), following the considered application field [18].

By comparing progressively the two models during the analysis of the human-machine system, and by reaching compromises aiming at satisfying a maximum of criteria, the data must be sufficiently relevant for the specification of an interactive system adapted to the users' requirements. Then, the task consists in specifying the HCI on the one hand, the identified application modules on the other hand. This set of specifications will have to be evaluated and validated from a socio-ergonomical point of view, so as to verify the relevance of the new solutions being integrated into the aimed human-machine system; indeed, in most cases this includes several human beings, inter-connected software and hardware packages. We will have to consider the collective aspects of the work, aspects which are generally neglected by the development teams [27][33].

After the specification of the HCI and the application modules, and in order to reach the coding stage, the preliminary and detailed design stages, respectively associated links in the V model with integration tests and unitary tests, are carried out in the usual way.

A version of this paper has been published in : *ACM SIGSOFT Software Engineering Notes*, 23 (3), pp. 93-96, 1998.

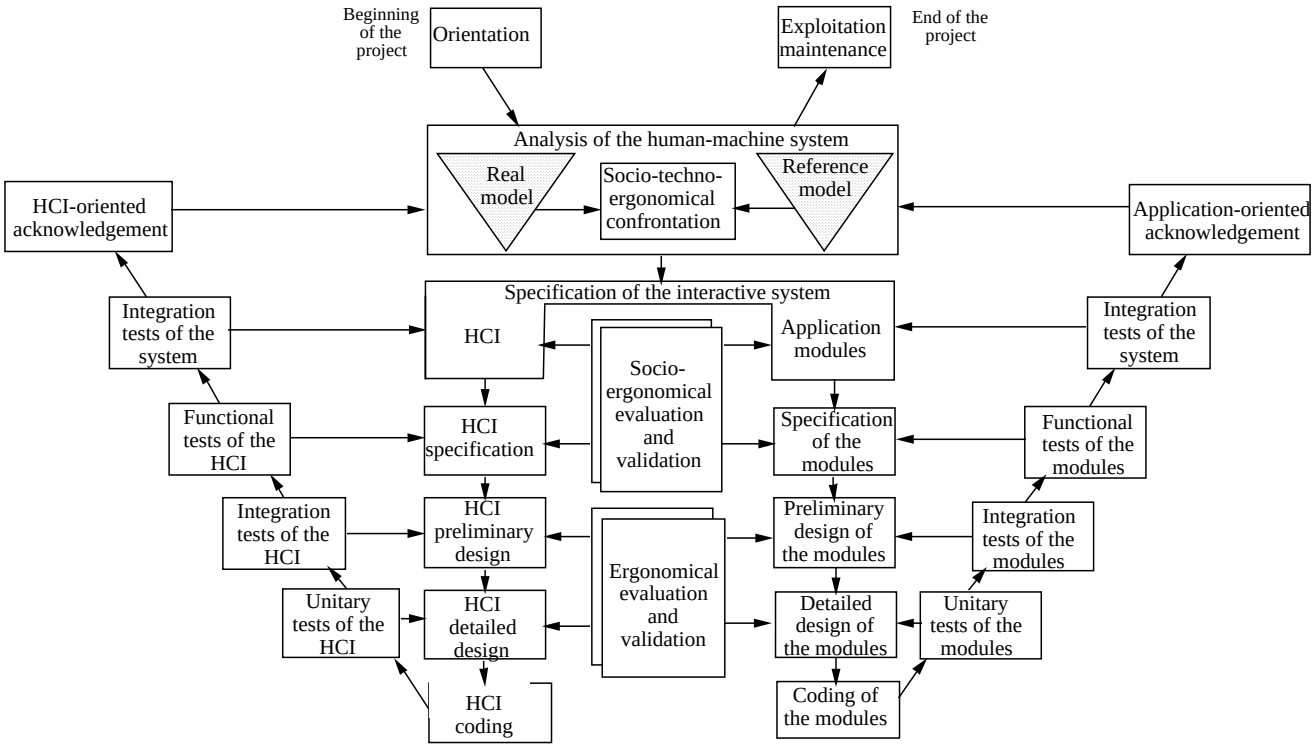

Figure 2. ∇ model [18].

In relation with the design stages, it is important to ergonomically evaluate and validate the components of the interactive system. Indeed, one can notice than in a great many industrial projects, HCI utility and usability evaluation is often skimped, awkwardly conducted, or even worse ignored or left aside. In these cases, the HCI cannot meet the user's demands. However, there are many methods derived from engineering and cognitive sciences which can contribute to the HCI evaluation (see for instance [13] [18] [22] [29] [31] [32].

These methods can be classified in three different approaches:

(1) User-centered approaches which consist in measuring the performances of users by acquiring and observing data, representative of the human-computer interaction, and then by analyzing them (observations, questionnaires, protocol analysis, video, monitoring, oculometric method, critical incidents, prototyping approach, final test bench...);

(2) Expertise-centered approaches which are based on the judgement of human factors experts, or can be based on evaluation grids or questionnaires listing the qualities of a "suitable" HCI;

(3) Modeling-centered approaches which consist in making the evaluation on a model of the HCI or of the human-computer interaction; these approaches use formal models (such as task models) associated with metrics, and allow to predict some difficulties or deficiencies concerning the HCI.

Some of these evaluation methods are usable at this level. The user-centered approaches, as well as expertise-centered approaches, are the most known and are more and more widely used. As regards the formal models issued from the modeling-centered approaches, they should ultimately provide the engineers and researchers with practical methods. Note also that, as indicated in the ∇ model, the chaining-process between the stages propitious to a prototyping approach, whose importance was expressed previously.

Like in each existing model, the acknowledgement stage has been located. In order to insist on interactive system development, we have chosen to split up this stage by symbolically distinguishing it into a HCI-oriented acknowledgement and an application-oriented one. These two stages should be minimized if the complete interactive system is conform to the data issued from the modeling of the human-machine system, and if each solution has been effectively evaluated and validated.

Finally this cycle ends with another stage, quite common in Software Engineering: the exploitation and maintenance stage.

In our mind, the ∇ model could now become a new theoretical and methodological framework for the researchers, concerned with the development of interactive systems. This framework is probably not perfect (NO model is perfect) and it will be necessary to validate it during industrial projects and to refine it over the next few years.

Conclusion and "Call For Answers"...

Many deficiencies applying to models and methods issued from Software Engineering -in terms of interactive systems development - logically lead to propose new ideas in this field. In this context, a development model which explains the stages linked with the technical and human aspects to be considered has been proposed. This model, called ∇ (nabla), is not a definitive model and must be considered as a guideline-schemes for future researches to be carried out in Software Engineering.

A version of this paper has been published in : *ACM SIGSOFT Software Engineering Notes*, 23 (3), pp. 93-96, 1998.

In relation with this model, several fundamental questions can be expressed:

- Is the ∇ model practical or impractical for (1) software engineering specialists, (2) HCI specialists, (3) a team regrouping specialists of these both fields?
- Is the ∇ model sufficiently, too or not enough explicit concerning (1) the consideration of the HCI aspects, (2) how to conduct a project?
- Is the decomposition in two parts (HCI and application) sufficient? (For instance, an interactive system is decomposed in the well-known SEEHEIM model [25] in four components: presentation, dialogue controller, interfaces with the application, application; three to five components are considered in other models).
- Must the notions concerning HCI-enriched models be integrated in Software Engineering lectures?

The readers of the journal are invited to study such ideas and/or to apply it in real (and/or simulated) situations. Some answers to the fundamental following questions could be provided from the use of such a model in different applications fields. A compilation of these answers could be the object of a next paper published in Software Engineering Notes.

References

[1] Abed, M. and J.C. Angue (1994): A new method for conception, realisationand evaluation of man-machine systems. In *Proceedings IEEE International Conference on Systems, Man and Cybernetics*, San Antonio, Texas, October 2-4.

[2] Barthet, M.F. (1995): The DIANE Method and its connection with MERISE Method. In *Proceedings IEA World Conference Ergonomic design, interfaces, products, Information*, October 16-20, Rio de Janeiro, Brazil, pp.106-110.

[3] Bass, L. and J. Coutaz (1991): *Developing Software for the user interface*. Addison-Wesley.

[4] Boehm, B.W. (1981): *Software Engineering Economics*. Englewoods Cliffs N.J.Prentice Hall.

[5] Boehm, B.W., T.E., Gray and T. Seewaldt (1984): Prototyping versus specifying: a multiproject experiment. *IEEE transactions on Software Engineering*, 10 (3), May.

[6] Card, S.K., T.P., Moran and A. Newell (1983): *The psychology of human-computer interaction*. Hillsdale, NJ: Erlbaum.

[7] Carter, J.A. (1991): Combining task analysis with software engineering in a methodology for designing interactive systems. In *Taking Software Design Seriously*, J. Karat (Ed.), Academic Press.

[8] Collins, D. (1995): *Designing Object-oriented user interfaces*. Benjamin/Cummings Publishing Company Inc., Redwood City, CA.

[9] Curtis, B. and B. Hefley (1994): A WIMP no more, the maturing of user interface engineering. *Interactions*, January, pp. 22-34.

[10] Diaper, D. (1989): *Task analysis for human-computer interaction.* Ellis Horwood Limited, Chichester, United Kingdom.

[11] Dix, A., A. Finlay, G.Abowd and R. Beale (1992): *Human-Computer Interaction.* Prentice-Hall, Inc., Englewood Cliffs, NJ.

[12] Goldberg, A. (1984): *Smalltalk-80, the interactive programming environment*.Addison-Wesley.

[13] Grislin, M., C., Kolski and J.C.Angue (1995): Human-computer interface evaluation in industrial complex systems: a review of usable techniques. In *Proceedings 6th IFAC/IFIP/IFORS/IEA Symposium on Analysis, Design and Evaluation of Man-Machine Systems*, M.I.T., Cambridge, USA, June 27-29.

[14] Helander, M. (1988): *Handbook of Human-Computer Interaction*. Elsevier Science Publishers B.V., North-Holland.

[15] James, M.G. (1991): PRODUSER: PROcessfor Developing USER Interfaces. In *Taking Software Design Seriously*, J. Karat (Ed.), Academic Press.

[16] Jaulent, P. (1990): *Génie logiciel, les méthodes*. Armand Colin,Paris.

[17] Karat, J. (1991): *Taking software design seriously*. Academic Press Limited, London.

[18] Kolski, C. (1997): *Interfaces homme-machine, application aux systèmes industriels complexes*. Editions Hermès, Paris.

[19] Larson, J.A. (1992): *Interactive software :tools for building interactive user interfaces*. Prentice-Hall, Inc.

[20] Lichter, H., M., Schneider-Hufschmidt and H. Zullighoven (1994): Prototyping in industrial software projects, bridging the gap between theory and practice. *IEEE Transactions on Software Engineering*, vol. 20, n. 11, pp. 825-832, November.

[21] Lim, K.Y. and J.B. Long (1994): *The MUSE Method for Usability Engineering*. Cambridge University Press, UK.

[22] Nielsen, J.(1993): *Usability engineering*. Academic Press.

[23] Nielsen, J. (1992): *The usability engineering life cycle*. *IEEE Computer*, 25 (3), pp. 12-22.

[24] Payne, S.J. and T.R.G. Green(1989): Task-Action Grammar: the model and developments. In *Task analysis for HCI*, Diaper (Ed), John Wiley and Sons, pp. 75-107.

[25] Pfaff, G.E. (1985): *User interface management system*. Springer-Verlag.

[26] Raccoon, L.B.S. (1997): Fifty years ofprogress in software engineering. *Software Engineering Notes*, vol. 22, no 1, pp 88-104.

[27] Rasmussen, J., B. Brehmerand J. Leplat (Eds.) (1991):*Distributed decision making*. Chichester, J. Wiley.

[28] Sommerville, I.(1994): *Software engineering*. Addison-Wesley.

[29] Sweeney, M., M., Maguire and B.Shackel (1993): Evaluating user-computer interaction: a framework. *International Journal of Man-Machine Studies*, 38, pp. 689-711.

[30] Thayer, R.H. and A.D. McGettrick (Eds.)(1993): *Software Engineering: A European Perspective*. IEEE Computer Society Press.

[31] Whitefield, A., F., Wilsonand J. Dowell (1991): Aframework for human factors evaluation. *Behaviour and Information Technology*, vol10, n°1, pp. 65-79.

[32] Wilson, J.R. and E.N. Corlett (eds.)(1990): *Evaluation of human works: a practical ergonomics methodology*. Taylor & Francis.

[33] Zorola-Villarreal, R., B., Pavard and R. Bastide (1995): SIM-COOP: A tool to analyse and predict cooperation in complex environments, a case study: the introduction of a datalink between controllers and pilots. In *Proceedings Fifth International Conference on Human-Machine Interaction and artificial Intelligence in Aerospace, HMI-AI-AS 95*,Toulouse, France, September.