Accepted at the Wokshop *Model for Developing High Impact Formatice Usability Evaluation Methods*, CHI'94, Boston

Supporting Usability Evaluation through Software Engineering Tools

Joëlle Coutaz, Daniel Salber, Sandrine Balbo, Francis Jambon

Laboratoire de Génie Informatique (LGI-IMAG)
BP 53, 38041 Grenoble Cedex 9, France
Tel. +33 76 51 48 54, +33 76 51 44 40, E-mail: Joelle.Coutaz@imag.fr, Daniel.Salber@imag.fr, Sandrine.Balbo@imag.fr, Francis.Jambon@imag.fr

1. Introduction

Software Engineering has provided the HCI community with tools and techniques for the software design and implementation of interactive systems. Although UIMS's, user interface toolkits, and architectural models improve usability in their own way, they do not explicitly address evaluation. On the other hand, HCI tools, methods, and notations are available but not incorporated into software engineering practice. In [Coutaz 94] we discuss how HCI and software engineering may cooperate through usability testing along the main phases of the development process. In this position paper, we propose an initial taxonomical framework for classifying and assessing usability methods and show how, as user interface software engineers, we are contributing to usability testing.

2. Classification frameworks

Techniques and methods for usability testing differ widely depending on the development stage, resources available, and the inclination and competence of the assessors.

- Theory-based tools such as GOMS [Card 83], CCT [Kieras 85], PUM [Young 90] and ICS [Barnard 87], should be favoured in the early stages of the development process as predictive tools. One may also resort to them as explanation tools after implementation has occurred: It is important to detect usability problems but it is equally important to understand the causes of the defects in order to take corrective actions.

- HCI heuristics and cognitive walkthrough are applicable once external specifications have been produced. They are cheap to perform because they do not require actual user testing. On the other hand, they rely on the competence of the assessors [Nielsen 90, Nielsen 92].

- Experimental approaches based on capturing behavioral data for well-targeted tests are favoured once implementation has occurred [Siocchi 91, Hammontree 92]. Implementation may be performed "by hand" from toolkits and application frameworks, or produced automatically via user interface generators.

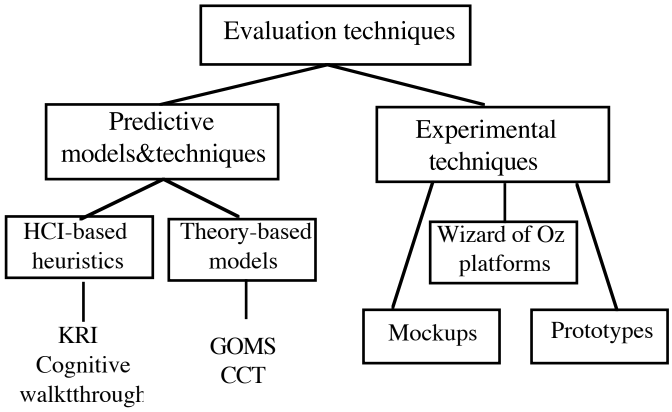

Figure 1: An overview of HCI evaluation techniques.

Figure 1 illustrates one general classification scheme. In Figure 2, we propose a more detailed view of the methodological space for usability testing. This framework makes explicit the presence or absence of the end-user, the design stage of the system development, the type of knowledge used to perform evaluation and the type of computer support for usability testing. The first two dimensions cover Whitefield et al.'s classification [Whitefield 91]. The type of knowledge used may be heuristic as in usability inspection by experts or may be formal as in PUM and GOMS. Formal knowledge may model the user(s), the user interface, the task and the environment. Computer support for usability testing includes capturing behavioral data at various levels of abstraction (from physical actions to system functions) and presentation of behavioral data as in DRUM [Macleod 93], critique such as KRI [Löwgren 90], and corrective action proposals.

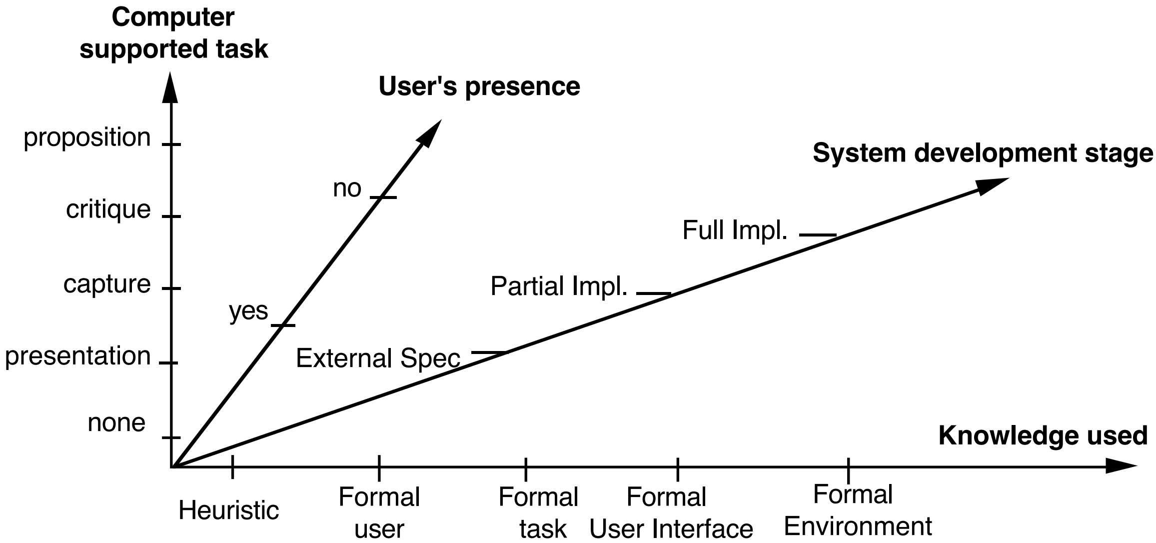

Figure 2: Proposed taxonomical model.

This framework can be used as a first step towards the classification and assessment of existing methods. In this framework, each method is modelled as a t-uple (U, S, K, A) where:
- U denotes the presence or absence of the end-user,
- S, the development stage of the system,
- K, the knowledge used,
- A, the type of computer-supported evaluation task.

Each element of the t-uple has some implications on the "usability" of the method. For example, usage of heuristics knowledge implies the availability of at least 3 well trained evaluators [Nielsen 90]. In some development contexts, access to this type of resource would be redhibitory. Given the value of a particular t-uple, one could make recommendations on the applicability/scope of the method in terms of cost. Identifying the criteria for measuring cost is one possible activity that we propose for the workshop. To do so, we may need to improve the taxonomical model by refining some of the dimensions proposed in Figure 2.

3. Our experience in Usability Testing

Figure 3 shows the framework of our current research agenda in developing tools for usability testing. Our tooling activity is directed at formal specifications as well as observational experiments: from the formal description of an interactive system and a set of generic but formalized properties, a predictive analysis tool will be able to detect usability problems. A correct specification will be used by a user interface generator to produce an executable version of the user interface as in Adept [Johnson 93]. In addition to adept, the code will be instrumented and plugged into a tool that captures behavioral data. These data can then be used as input to an automatic analyser. In the next paragraph, we motivate our choices and indicate what we have already built.

Formal notations for external specifications can be used as predictive instruments based on theoretically motivated principles. This initiative is not new (e.g., Payne's, Green's, and Reisner's work) but must be pursued. Usage at the external specifications stage of formal notations such as

UAN [Hartson 92], opens the way to automatic usability testing before implementation proceeds. To illustrate the potential of the approach, we consider one example of rules drawn from our own experience using UAN to describe the user interface of MATIS [Nigay 93], a Multimodal Air Travel Information System: If the precondition for a task execution cannot be expressed in terms of perceivable features, then the "observability" criteria is broken: the user is not aware that the task can be performed. Properties such as observability, predictability, honesty, are derived from the literature [Dix 91] as well as from our own early work in this area [Abowd 92]. In the taxonomy of Figure 2, the predictive tool is characterized by the t-uple : <No End-User required, External specifications of the system required, (Knowledge about the user interface, Knowledge about the task), Computer-supported critique>.

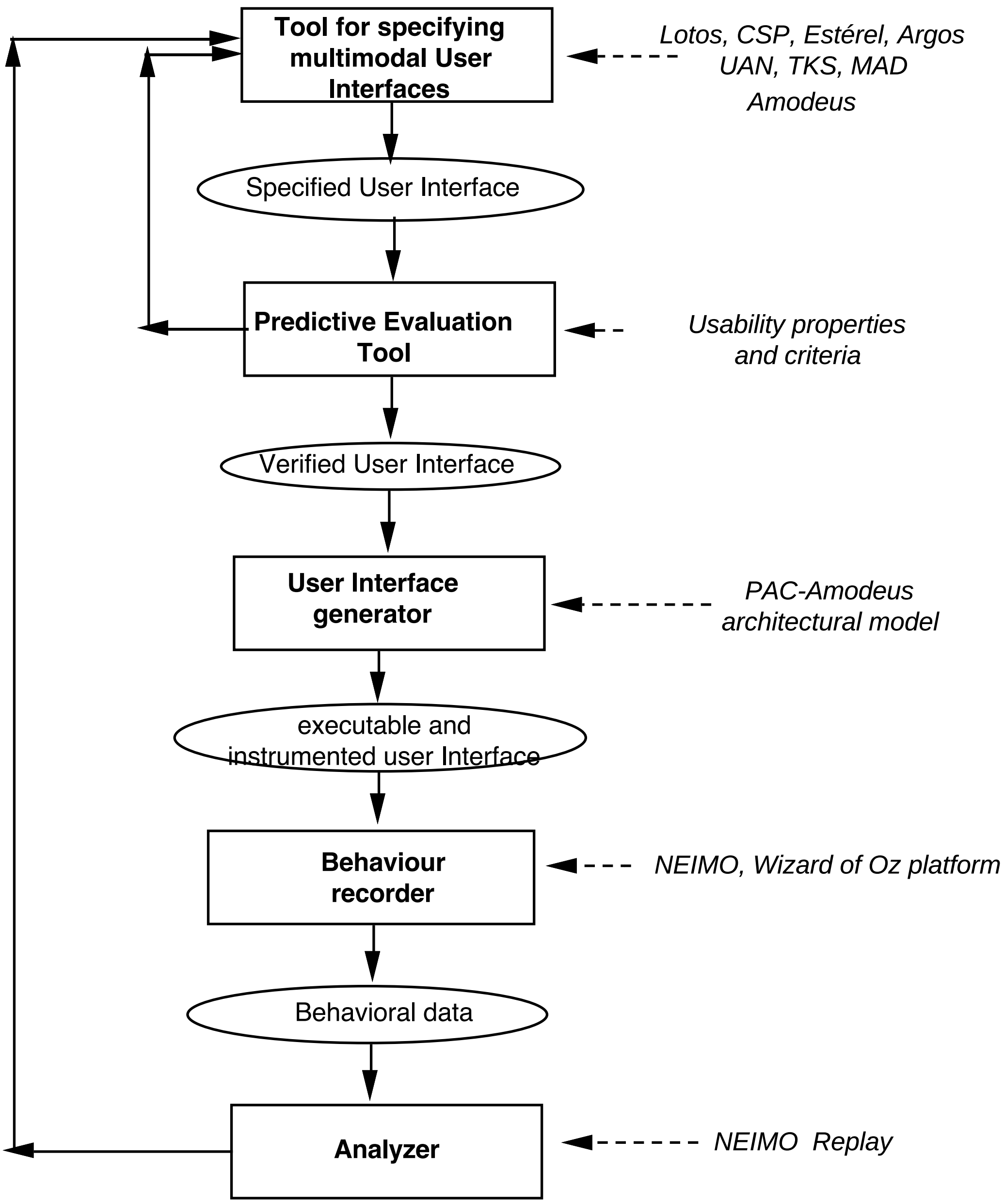

Figure 3: Framework of our current research agenda for usability testing. Italic denotes the expertise we have developed or we are using to achieve the goal.

A correct specification checked through the predictive tool will be fed into a user interface generator that will produce a user interface automatically. This prototype will be automatically instrumented and plugged into the NEIMO observation platform. By doing so, formal specifications join the camp of experimental approaches forming an integrated environment for engineering usability.

NEIMO is a multi-wizard generic Wizard of Oz platform that automatically digitizes and records behavioral data ranging from low level physical actions to high level commands performed by a subject using a multimodal user interface [Coutaz 93, Salber 93a, Salber 93b]. Observational experiments are unavoidable as long as predictive theories are limited in scope. Observations are currently performed within usability laboratories. These laboratories are well equipped with analog recording which requires time consuming analysis. The aim of NEIMO is to computerize this task by extending the current Wizard of Oz (WOz) techniques, by automatically instrumenting user interfaces and by providing high level critics based on formal descriptions of the intended task model and of the user interface.

In his current state, NEIMO is characterized as <(End-User required, possibly multiple wizards), partially running prototype, (Formal knowledge of the user interface, Formal knowledge of the task), capture>. The goal is to extend the analysis tools towards an automatic critique. An early work, based on formal task modelling has been developed towards this goal [Balbo 93].

4. Summary of contribution

In summary, we can contribute to the workshop in three ways:

- use the taxonomic framework of Figure 2 as a starting point for assessing evaluation methods;

- provide HCI participants of the workshop with a software engineering perspective. In particular we favour Long's engineering approach [Dowell 89] similar to the methodology advocated in software quality assurance: define a quality goal and its metrics (i.e., identify criterias to assess the goal, specify the methods to measure the criterias, document results and corrective actions).

- discuss how software tools can support usability testing.

5. Acknowledgement

This work has been partly supported by project ESPRIT BR 7040 Amodeus II.

6. References

[Abowd 92]
G. Abowd, J. Coutaz, L. Nigay
Structuring the Space of Interactive System Properties, IFIP TC2/WG2.7 Working Conference on Engineering for Human Computer Interaction, Elsevier Publ., Ellivuori, Finland, August, 1992.

[Balbo 93]
S. Balbo, J. Coutaz, "Automatic Evaluation in Human Computer Interaction"; The Ergonomics Society 1993 Annual Conference, Edinburgh, 13-16 April, 1993.

[Barnard 87]
P.J. Barnard, "Cognitive Resources and the Learning of Humnan-Computer Dialogs", in Interfacing Thought, Cognitive Aspects of Human-Computer Interaction, J.M. Carroll Ed., MIT Press Publ., 1987, pp.112-158

[Card 83]
S.K. Card, T.P. Moran & A. Newell, "The psychology of Human Computer Interaction", Lawrence Erlbaum Associates, 1983

[Coutaz 93]
J. Coutaz, D. Salber, S. Balbo
Towards the Automatic Evaluation of Multimodal User Interfaces, numéro spécial "Intelligent User Interfaces", Knowledge-Based Systems, 1993.

[Coutaz 94]
J. Coutaz, "Evaluation techniques: Exploring the intersection of HCI and Software engineering", to appear in the proceedings of the workshop "Research Issues in the Intersection between HCI and Software Engineering", International Conference on Software Engiineering, ICSE'94, Sorrento, May, 1994.

[Dix 91]
A. J. Dix, "Formal Methods for interactive Systems", Computers and People series, Academic Press, 1991.

[Dowell 89]
J. Dowell, J. Long, "Towards a Conception for an Engineering Discipline of Human Factors", in Ergonomics, 32, 1513-1535

[Hammontree 92]
M.L. Hammontree, J.J. Hendrickson, B.W. Hensley, "Integrated Data Capture and Analysis Tools for Research and Testing on Graphical User Interfaces", in the CHI'92 Conference Proceedings, ACM Press Publ., 1992, pp. 431-432

[Hartson 92]
R. Hartson, P.D. Gray, “Temporal Aspects of Tasks in the User Action Notation”, Human-Computer Interaction, Laurence Erlbaum, vol. 7, No 1, pp. 1-45, 1992
[Johnson et al. 93]
P. Johnson, S. Wilson, P. Markopoulos, J. Pycock, “ADEPT-Advanced Design Environment fror Prototyping With Task Models”, In adjunct Proceedings, INTERCHI’93, 1993, pp. 56-56
[Kieras 85]
D. Kieras, P.G. Polson “An Approach to the Formal Analysis of User Complexity”, International Journal of Man-Machine Studies, 22, 1985, pp. 365-394
[Lewis 90]
C. Lewis, P. Polson, C. Wharton, J. Rieman, “Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces, in Proceedings of CHI’90, ACM Press, 1990, pp. 235-241
[Löwgren 90]
J. Löwgren & T. Nordqvist, “A Knowledge-Based Tool for User Interface Evaluation and its Integration in a UIMS”, Human-Computer Interaction-INTERACT'90, pp. 395-400
[Macleod 93]
M. Macleod, “DRUM, Diagnostic Recorder fro Usability Measurment”, NPL, DITC HCI Group, Teddington, Middlesex, TW11 0LW, UK, 1993
[Nielsen 90]
J. Nielsen, R. Molich, “Heuristic Evaluation of User Interfaces”, in CHI’90 Proceedings, ACM Press Publ., 1990, pp. 249-256
[Nielsen 92]
J. Nielsen, V. L. Phillips, “Estimating the relative usability of two interfaces:Heuristic, formal and empirical methods compared”, in CHI’92 Proceedings, ACM Press Publ., 1992, pp. 214-221
[Nigay et al. 93]
L. Nigay, J. Coutaz, “A design space for multimodal interfaces: concurrent processing and data fusion”. INTERCHI’93 Proceedings, Amsterdam, May, 1993, pp. 172-178.
[Salber 93a]
D. Salber, J. Coutaz, “Applying the Wizard of Oz Technique to the Study of Multimodal Systems”. In Human Computer Interaction, 3rd International Conference EWHCI’93, East/West Human Computer Interaction, Moscow. L. Bass, J. Gornostaaev, C. Unger Eds. Springer Verlag Publ., Lecture notes in Computer Science, Vol. 753, 1993.
[Salber 93b]
D. Salber, J. Coutaz, "A Wizard of Oz Platform for the Study of Multimodal Systems; INTERCHI’93 Adjunct Proceedings, Amsterdam, May, 1993, pp. 95-96.
[Siochi 91]
A. Siochi & D. Hix, “A study of Computer-Supported User interface Evaluation Using Maximal Repeating Patern Analysis”, in Proceedings of the CHI’91 Conference, ACM Press, pp. 301-305, 1991.
[Whitefield 91]
A. Whitefield, F. Wilson & J. Dowell, "A framework for human factors evaluation", Behaviour and information technology, vol. 10, no. 1, pp. 65-79, 1991
[Young 90]
R.M. Young & J. Whittington, “Interim Report on the Instruction Langage”, AMODEUS Project Document: Deliverable D5, ESPRIT Basic Research Action 3066, 11th August 1990