



ELSEVIER

Interacting with Computers 14 (2002) 407–409

www.elsevier.com/locate/intcom

**Interacting
with
Computers**

Discussion

Commentary on ‘scenarios and task analysis’ by Dan Diaper

Fabio Paternò*

ISTI, C.N.R., Via G. Moruzzi 1, 56100 Pisa, Italy

Accepted 4 February 2002

Both Carroll’s book and Diaper’s paper address fundamental issues for the HCI field and stimulate important discussion that can help to clarify at least some of them. I have noticed that in recent years some tension has arisen between scenario-based approaches and task analysis, with the risk of degenerating into a kind of religious war. To some extent, the solution to this tension is simple: both approaches are important and complementary. However, this solution is often interpreted as a diplomatic, polite position in which people do not actually believe. So, some more in-depth reflections should prove to be useful.

We can start with scenarios. I think the importance of scenarios is generally recognized and their adoption in current practice is impressive. However, I have some concerns on this topic. I have seen scenarios being used even by people who have no knowledge of the HCI literature. When people want to discuss design decisions or requirements for a new system it is very intuitive to propose scenarios in terms of examples of possible use. So, I would expect that the role of researchers in HCI be, not be recall the importance of scenarios, but to provide techniques to identify scenarios relevant to current aims, to extend and elaborate scenarios and better use the information that they contain. Unfortunately, I feel such aspects have not been dealt with extensively enough in Carroll’s 350-page long book. There is a section devoted to seven techniques for eliciting scenarios, but I agree with Diaper when he comments that these are general HCI methods that have been developed in a broader context (such as ethnographic studies and participatory design. Chapter 10 (which I found to be the book’s most interesting) also addresses some aspects of this issue but I would have expected this to be a central theme in the book.

Another fundamental question is: what is a scenario? Reading Carroll’s book I got the impression that any type of description of system use can be a scenario. At some point, we are told that a scenario is a story. However, some scenarios in the book do not seem like

* Tel.: +39-50-3153066; fax: +39-50-3138091.

E-mail address: fabio.paterno@cnuce.cnr.it (F. Paternò).

stories at all. Elsewhere in the book we are told variously that use cases are ‘simplified views of task scenarios’ (pp. 237), that a description of a single system use is a scenario, and that a description of a group of system uses is also a scenario. Actually, some authors disagree with this and make an explicit distinction between stories that also include some emotional aspects and, at least, use cases, for example, in Ref. Imaz and Benyon (1999). All this reminds me of a similar discussion on use cases; see for example Cockburn (1997) where the authors have classified 18 definitions of use cases depending on four issues (purpose, contents, plurality and structure). The book’s author seems to adopt an eclectic approach. More generally, the importance of eclecticism is highlighted in the book. I fear that eclectic approaches can sometimes degenerate into opportunism that is quite understandable in practitioners’ work, less so when researchers aim to create a science of design.

Having had such concerns, I appreciate Diaper’s paper, which aims to balance the contribution of scenarios and better position them with respect other HCI techniques such as task analysis. In the abstract, he claims that scenarios and their use in software engineering can fit into the broader framework of task analysis. I think this statement is too strong, because I do not believe that either of the two techniques should be fit into the other. Actually each of them can be applied without the other, even though I would recommend using both of them to solve the scenario dilemma and ‘the dynamic tension between specification and imagination’.

A related topic dealt with by Carroll is claims analysis, which he considers the desirable and undesirable consequences stemming from the scenario. While I found the concept useful and helpful, I agree with Diaper’s comments that more structure would be needed as would better indications of the level of abstractions that should be considered in each phase of design.

More generally, I find it telling that the practice of software engineering applies scenarios mainly in the requirements phase (pp. 319). Thus, their strong point is the ability to highlight issues and stimulate discussion while requiring limited effort to develop, at least as compared to other more formal techniques. However, since I believe a scenario describes a single system use (whereas Carroll sometimes seems to tend to overstate their coverage), there is a need for other techniques that may provide less detail but broader scope. Here is where task analysis comes in. Moreover, I would distinguish between task analysis and task modeling which Diaper does not do. While the purpose of task analysis is to understand what tasks should be supported and what their related attributes are, the aim of task modeling is to identify more precisely the relationships among such tasks (Paternò, 1999). The need for modeling is most acutely felt when the design aims to support system implementation as well. If we gave developers only a set of scenarios, they would have to make many design decisions on their own in order to obtain a complete system. This is why UML has widely been adopted in software engineering: its purpose is to provide a set of related representations (ranging from use cases to class diagrams) to support designers and developers across the various phases of the design cycle.

The most-structured representation proposed by Carroll is the interaction script, which reminds me of the UAN (Hartson and Gray, 1992) tables associated with basic tasks but, once again, Diaper highlights some potential limitations of this representation.

In closing, I found both the book and the paper enlightening. The book has given me some useful insights as to how I can use scenarios. The final part has also highlighted some

interesting research issues. Diaper's critique helps to better define scenarios' position within the broader context of HCI.

References

- Cockburn, A., 1997. Structuring use cases with goals. *J Object-oriented Program*. <http://members.aol.com/acockburn>.
- Hartson, R., Gray, P., 1992. Temporal aspects of tasks in the user action notation. *Human Comput Interaction* 7, 1–45.
- Imaz, M., Benyon, D., 1999. How stories capture interactions. *Proceedings Interact'99*, pp. 321–328.
- Paternò, F., 1999. *Model-based Design and Evaluation of Interactive Applications*. Springer, Berlin ISBN 1-85233-155-0.