Editors: Tiziana Catarci and Thomas D.C. Little

Guidelines for Hypermedia Usability Inspection

Maria F. Costabile and Maristella Matera
University of Bari, Italy

It is generally acknowledged that the only way to actually determine the quality of interactive systems is to perform a usability evaluation. This proves especially true for multimedia systems, where using concurrent media generates further usability problems. We aim to develop effective evaluation methods to meet industry demand and address concerns about the lack of cost-effective methods. These issues prevent most companies from performing usability evaluation, resulting in poorly designed and unusable software.

Among the various usability evaluation approaches, usability inspection methods are gaining popularity because they cost less than traditional lab-based usability evaluations. These methods involve expert evaluators only, who inspect the application and, based on their knowledge, provide judgments about the usability of the different application elements. (Examples of inspection methods include heuristic evaluation, cognitive walkthrough, guideline review, and formal usability inspection.[1])

A drawback of these inspection methods is that the results depend on the inspectors' skills, experience, and ability. On the other hand, training inspectors proves difficult and expensive. Another limitation is that the techniques mostly focus on surface-oriented features related to the graphical interface. Few approaches focus on the application structure or on the organization of information elements and functionality.[2] Also, most methods are too general. Although they're valid for any type of interactive system, they often neglect some intrinsic features.

Our research addresses evaluating the usability of hypermedia systems—both offline (CD-ROMs) and online (Web)—and tries to capture the features that most characterize the specific nature of these systems. Here, we describe an inspection technique that lets evaluators concentrate on the usability of specific aspects of hypermedia applications, such as information and navigation structuring, media integration and synchronization, and so on, without neglecting the surface aspects. Our technique uses operational guidelines, called Abstract Tasks (ATs), which systematically drive the inspection activities, allowing even less experienced evaluators to come up with valuable results.

The AT-based inspection

We originally developed this inspection technique as part of an overall methodology for usability evaluation, called SUE.[3] We hoped to help usability inspectors share and transfer their evaluation know-how, make the hypermedia inspection process easier for newcomers who have little experience in how to conduct the inspection, and achieve more effective and efficient evaluations.

Using a library of 40 ATs is the most innovative feature introduced by the SUE inspection.[3–5] An AT describes the operational activities (or tasks) that the expert evaluators should perform during the inspection. We use the term abstract, because the activity specifications are formulated independently from a particular application, and they refer to categories, or types, of application constituents more than to specific constituents.

We formulate an AT by following a precise pattern template, which includes five items:

- The Classification Code and Title identify the AT and convey its essence.
- The Focus of Action briefly describes the AT's context, or focus, by listing the application constituents that represent the evaluation entities.
- The Intent describes the problem addressed by the AT and its rationale, trying to clarify what goal the AT application should achieve.
- The Activity Description describes the activi-

1070-986X/01/$10.00 © 2001 IEEE

An AT Example

Here's a sample Abstract Task taken from the AT library that we defined for hypermedia applications.[1] The reported Example of Inspection Findings refers to the Italian commercial CD-ROM *Camminare nella pittura* (or *Walking through Painting*), and DS-1 refers to the first AT for the evaluation of dynamic slots.

DS-1: "Control on Dynamic Slots"

Focus of Action

A dynamic slot is a time-based multimedia object, such as videos, sounds, or animations.

Intent

Dynamic slot control refers to the activation commands and to the feedback mechanisms allowing users to control the dynamic slots' activation. The intent of this AT is, therefore, to evaluate

- commands for the dynamic slot activation and
- mechanisms supporting the state visibility, or the identification of any intermediate state of the dynamic slot activation.

Activity Description

Given a dynamic slot, execute commands such as play, suspend, continue, stop, replay, and get to an intermediate state. At a given instant, during the activation of the dynamic slot, verify if it's possible to identify its current state, as well as its evolution up to the end.

Output

Describe the set of control commands and the set of mechanisms supporting the state visibility. Then, answer the following questions:

- Are the type and number of commands appropriate, with respect to the intrinsic nature of the dynamic slot?
- Would further commands make the dynamic slot control more effective?
- Are the mechanisms supporting the state visibility evident and effective?

Figure A. In Camminare nella Pittura, a General Theme (or "Temi Generali" in Italian) node. (Courtesy of Mondadori New Media, Copyright 1997.)

Example of Inspection Findings

In *Camminare nella pittura*, General Theme nodes include a composite dynamic slot, made by an image slide show synchronized with an audio comment. This slot is automatically activated when the page is entered. A slider, displayed on the right of the images, moves automatically as the images and the audio comment go on (see Figure A). Users can also manually move the slider to reach intermediate positions within the sequence of images in the slide show. Unfortunately, moving to an intermediate state is the only way to control the slot. In fact, it's not possible to pause or stop it. Also, when users put the slider in its last position, instead of stopping, the slide show becomes activated again.

We identified additional cases where dynamic slots control are lacking in other pages—for example, the General Index node, where users can start the audio comment with a button but can't pause or stop it. Moreover, this slot doesn't show any feedback mechanism giving indications about the evolution of its activation.

Reference

1. M. Matera, *SUE: A Systematic Methodology for Evaluating Hypermedia Usability*, doctoral dissertation, Dipartimento di Elettronica e Informazione, Politecnico di Milano, 2000.

ties evaluators will perform during the AT application.

- The Output indicates what the inspectors should provide in the evaluation report.
- The Examples of Inspection Findings clarify which situations the evaluators should look for while applying the AT activity. They also provide proof of the AT applicability. (See the sidebar for an example.)

During inspection, evaluators first identify the different application components whose design

A preliminary look at the data seems to confirm the advantage of using ATs, even without a model-based specification of the application.

must be verified. They select which ATs to apply from the library, depending on the application objects mentioned in their Focus of Action. They start the inspection by applying some basic ATs that support the detailed analysis of the structure and dynamics for basic objects often occurring in hypermedia applications. Such ATs also help evaluators better understand the application's overall organization. Then, evaluators proceed with ATs referring to more advanced features. During the application of each AT, evaluators perform the actions specified in the Activity Description and annotate the observed structural or dynamical patterns. Thus, they generate an evaluation report.

Validating the AT-based inspection

The original AT formulation reflected concepts and terminology of a hypermedia design model (HDM).[6] This model suggested which application objects the ATs should focus on. It also helped us define a set of usability attributes[5] as criteria to guide the evaluation process and clarify how the model primitives should be composed to obtain usable structures. Moreover, the SUE inspection prescribed using an HDM application schema to highlight the main constituents in which the application could be decomposed according to the model primitives. Therefore, the SUE inspection applied the ATs while walking through the application. Evaluators had an HDM schema highlighting the application objects to look for and evaluate available on paper. When the HDM schema wasn't already available, the evaluators had to generate it.

To evaluate this technique, we first performed a controlled experiment, involving 26 novice evaluators.[7] We divided them into two groups and asked them to evaluate a commercial CD-ROM. One group (HI) performed a heuristic evaluation,[1] using our usability attributes as heuristics to inspect the application. The other group (SI) performed the SUE inspection, basing their evaluation on using a set of ATs, together with an HDM application schema. The experiment confirmed that the SUE inspection enhanced the evaluation's effectiveness and efficiency, as well as the evaluators' satisfaction.[7]

However, from a postexperiment questionnaire, we found that evaluators using ATs had difficulties fully understanding both the HDM application schema and the ATs. While analyzing the experimental results, we realized that exploiting HDM could be our approach's strength and weakness. Indeed, although exploiting some HDM concepts let us define a systematic method for guiding evaluators in identifying critical objects and evaluating them, forcing them to manage a model and a terminology they weren't familiar with could reduce the method's efficiency and the evaluators' satisfaction. Therefore, we revised the AT formulation to help evaluators identify and evaluate some application components without forcing them to learn the HDM model.

To verify the relevance of the role played by the HDM application schema with respect to the role the ATs played alone, we performed a second experiment involving 20 novice evaluators. We followed the same design as the first experiment; the only difference is that the second group performed the inspection without using the HDM application schema, only applying the ATs. We called this AT-based inspection (ATI).

We're still analyzing the results of the second experiment, but a preliminary look at the data seems to confirm the advantage of using ATs, even without a model-based specification of the application. The graphs in Figure 1 show the Nielsen's cost-benefit curve[1] for the two experiments. We derived such a curve, which relates the proportion of usability problems to the number of evaluators, from a mathematical model based on the following prediction formula for the number of usability problems that evaluators can find during a usability inspection: $\text{Found}(i) = n(1 - (1 - (\lambda)^i)$. $\text{Found}(i)$ is the number of problems found by aggregating evaluation reports from i independent evaluators, n is the total number of problems in the application, and λ is the probability of finding the average usability problem when using a single, average evaluator. One possible use of this model is the estimation of the number of inspectors needed to identify a given percentage of usability errors.

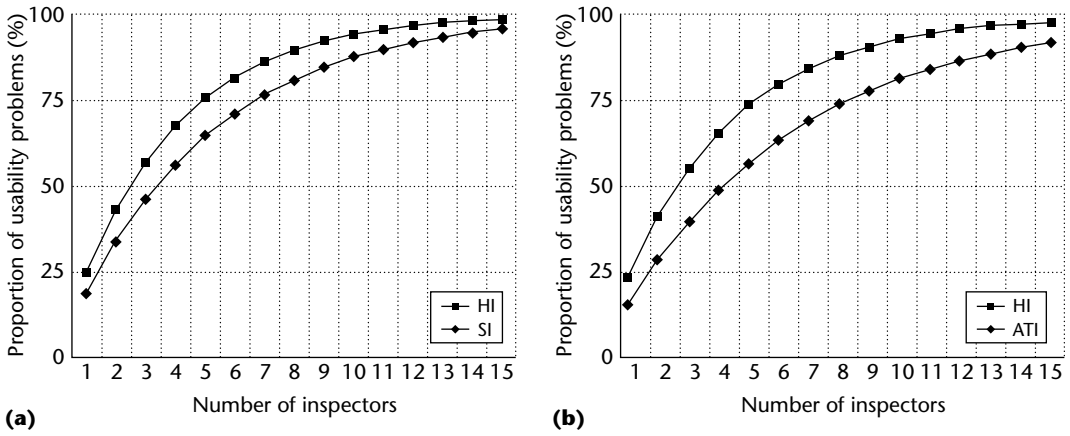

Figure 1. The Nielsen's cost-benefit curves show the proportion of usability problems found by evaluators using heuristic evaluation (HI) and others using the SUE inspection (SI) in the first experiment (a), and HI and Abstract Task insection (ATI) evaluators in the second experiment (b). In Figure 1a, n = 38, $\lambda_{HI} = 0.19$, and $\lambda_{SI} = 0.24$; in Figure 1b, n = 38, $\lambda_{HI} = 0.15$, and $\lambda_{ATI} = 0.23$.

Therefore, we used the mathematical model to determine the required number of inspectors (for the techniques used) to detect a reasonable percentage of problems in the application.

As the graphs show, in both experiments, ATs tend to reach better performance with a low number of evaluators. For example, assuming the Nielsen's 75 percent threshold is a reference point,[1] in the second experiment, the AT-based inspection could reach it with five evaluators. The heuristic evaluation would need eight evaluators, which confirms the intrinsic power of ATs in driving the evaluators' activities.

Conclusion

As they currently exist, we see ATs as evaluation patterns, which make it possible to maximize the reuse of an evaluator's expertise. That is, reuse takes advantage of previous work, thus reducing the effort to create a new one. ATs support the reuse of an evaluator's know-how. Their goal is to capture usability inspection expertise and to express it in a precise and understandable form, so that others can easily reproduce, communicate, and exploit it.

Our current work is devoted to extending the AT library to cover specific usability issues related to e-commerce Web sites. **MM**

References

1. J. Nielsen and R.L. Mack, *Usability Inspection Methods,* John Wiley & Sons, New York, 1994.
2. T.R.G. Green and D.R. Benyon, "The Skull Beneath the Skin; Entity-Relationship Modelling of Information Artifacts," *Int'l J. Human-Computer Studies*, vol. 44, no. 6, 1996, pp. 801-828.
3. M.F.Costabile et al., *SUE: A Systematic Usability Evaluation*, tech. report 19-97, Dipartimento di Elettronica e Informazione, Politecnico di Milano, Milan, Italy, 1997.
4. M. Matera, *SUE: A Systematic Methodology for Evaluating Hypermedia Usability*, doctoral dissertation, Dipartimento di Elettronica e Informazione, Politecnico di Milano, 2000.
5. F. Garzotto and M. Matera, "A Systematic Method for Hypermedia Usability Inspection," *The New Review of Hypermedia and Multimedia*, vol. 3, 1997, pp. 39-65.
6. F. Garzotto, P. Paolini, and D. Schwabe, "HDM— A Model-Based Approach to Hypermedia Application Design," *ACM Trans. Information Systems*, vol. 11, no. 1, Jan. 1993, pp. 1-26.
7. A. De Angeli et al., "Validating the SUE Inspection Technique," *Proc. Conf. Advanced Visual Interfaces* (AVI 2000), ACM Press, New York, 2000, pp. 143-150.

Readers may contact Matera at the Department of Information Systems, University of Bari, Via Orabona, 4, 70125, Bari, Italy, email matera@di.uniba,it.