Human–Computer Interaction — INTERACT'99
Angela Sasse and Chris Johnson (Editors)
Published by IOS Press, © IFIP TC.13, 1999

Desperado: Three-in-one Indexing for Innovative Design

Thomas C Ormerod[1], John Mariani[2], Linden J Ball[3] & Nicki Lambell[3]

[1] Department of Psychology,
[2] Department of Computing
Lancaster University, Lancaster LA1 4YF, UK.
[3] Division of Psychology, Derby University DE3 5GX, UK.
{T.Ormerod,J.Mariani}@lancaster.ac.uk,{L.J.Ball,N.Lambell}@derby.ac.uk

Abstract: Despite the potential offered by reuse of information from previous projects in ongoing design work for greater cost-effectiveness and innovation, it is beset by problems, which are not resolved by existing design documentation approaches. We describe Desperado, an indexing system that adopts a novel approach to supporting design reuse. Desperado offers three concurrent facilities — component encoding, rationale capture and guided retrieval — within a single environment. Each facility is supported by a system-initiated search of an object-oriented database of previous design episodes.

Keywords: design rationale, documentation, solution reuse, information retrieval, information indexing, ethnography.

1 Introduction

Expert designers working in commercial environments face a constant dilemma in which they must balance innovation against efficiency. In such an environment, the reuse of previous design work is an attractive proposition. Reuse avoids repetition of design effort, and maintains upward product compatibility and consistency with legal or company standards. Furthermore, with appropriate tools to support the process, reuse might actively enhance innovation. The notion of innovative design reuse sounds at first like an oxymoron. However, providing access to previously considered design options enables the maintenance of innovative ideas over time. When designers know that ideas, if not implemented immediately, may be useful to future projects, they may explore design problems more creatively.

Two types of design reuse information can be identified. The first is product-oriented, focusing upon reuse of solution or component information. The second is process-oriented, focusing upon reuse of ideas, conversations, arguments, decisions and critiques, an orientation associated with 'design rationale'. Design rationale is the documentation of the underlying reasoning behind an artifact design, in which attempts to improve productivity are centred around encouraging a reflective examination of the design process itself. Design rationale might aid understanding amongst stakeholders in the design process and improve the quality of their reasoning. Also it can aid redesign and modification. There is evidence that previous design concepts and prototypes do get reused in many 'routine' design situations (Gero, 1990). However, despite their promise, solution and rationale reuse are limited in application. This is partly because the effort needed to carry them out outweighs the apparent benefits. Often the people who create a design rationale are not motivated because they do not directly benefit from it (Grudin, 1988). There is evidence that designers suffer from 'design rationale fatigue' (Conklin & Begeman, 1988). Existing methods for eliciting design rationales have been shown to be difficult to grasp at first and can cause disruption during the 'construction phase' (Buckingham Shum, 1996). Also, existing solutions can restrict the range of options pursued by designers. Even experts engage in 'satisficing' behaviours, becoming fixated upon single solutions rather than exploring alternatives in order to optimise choices (Ball et al., 1997).

Despite these problems, we believe that there is great potential for effective design reuse. In this paper, we describe an ongoing project to develop Desperado,

a computer-based indexing system for supporting the reuse of rationale and solution-based design information. Our approach is to provide a method for capturing solution and rationale information within an information retrieval environment. This contrasts with approaches in which solution reuse, design rationale and information retrieval are separate, either by time or by artifact.

In the next section we report an ethnographic study of documentation and reuse practices in industrial design groups. Then, the features of Desperado are described, a walk-through of a typical user interaction is outlined.

2 A Study of Reuse Practices

We carried out ethnographic studies of documentation and reuse practices in four industrial, software and aerospace engineering companies. Although design activities differed substantially across companies, our data (videotapes, conversation and meeting transcripts and field notes) reveal consistencies of both problems faced in, and opportunities for, reuse of design information. In this section we focus upon main aspects of the data: existing documentation practices, information retrieval strategies and situated design activities in each company. For a detailed description see (Ball & Ormerod, to appear).

2.1 Existing Documentation Practices

Assessment of existing documentation practices was important for two reasons. First, in designing Desperado we had to ensure that it supports existing documentation requirements, since designers would be unlikely to adopt a new system that did not also deliver existing functionality. Second, we needed to assess the extent to which designers were willing to engage in documentation, and to investigate aspects of existing systems that frustrate documentation behaviours.

Each company had in place a number of information repositories, such as paper-based archives, computer-aided design databases and diary-based file stores. The extent to which each repository was used successfully was extremely variable, though an overall evaluation (not only by ourselves but more importantly by designers and design managers who used the systems) is that they failed to support systematic and optimal reuse practices.

In many instances, documentation was sporadic and unsuccessful. For example, one company had installed a project management system based around a computer-networked diary, which automatically created folders in a project file hierarchy for designers to deposit project-related information. In demonstrating the system to our researcher, the team manager was unable to find a single folder that had any contents, despite the fact that the system had been running for two years prior to our visit. It became clear, from subsequent conversations with members of the design team that they maintained their own information repositories, only occasionally shifting files to the designated project folders immediately prior to team meetings. In this case, the benefits of central organisation of project-related information, although recognised by each member of the design group, were not regarded as sufficient to overcome the additional burdens faced by individual designers in re-assigning their files during an ongoing design process to a central repository.

In all companies information was typically encoded in a piecemeal fashion, with file names specifying either the players, project names, components under design, the date, or some combination of these categories. There was no evidence of encoding by process, criteria or question-based categories. This contrasts with transcripts of conversations and meetings, in which process, criteria and question topics were typically the focus of discussions.

While encoding was piecemeal, it nonetheless was regularly practised. Informal conversations with the designers indicated a frustration with existing systems, and a desire for better documentation technologies. It appears, then, that designers are willing to invest time in encoding for reuse and that they do recognise the benefits of such practices. Thus, an important outcome from the studies was an increased motivation to develop an effective documentation technology.

Observations of documentation problems revealed a need to support the process of encoding in three main ways. First, problems in naming consistency that have been observed elsewhere were apparent in many of our companies. Second, the extent to which documentation took place was variable. In the two companies where extensive documentation was kept reliably, design work was only documented formally at a late stage when a component or design approach had received managerial approval. Records of design alternatives or criteria for rejection and acceptance were rarely recorded. Thus there is a need to provide structure to the documentation process, through some form of concurrent model that steps designers through the documentation process. Third, it was apparent that, despite the willingness of designers to engage in documentation, it was determined in

large part by response to organisation or managerial demands. Thus, system design needs to be flexible enough to embody existing organisational practices (e.g. company-specific stages).

2.2 Information Retrieval Strategies

The presence of existing reuse practices is potentially a useful resource for modelling system-initiated retrieval. In particular, we were interested in the knowledge sources and types that designers relied upon to cue reuse.

We encountered many examples of retrieval failure. For example, we observed one designer engaging in a week-long search for a piece of design information. It took him a day to locate the 'owner' of the project to which the information pertained, and a further two days to find that this individual had retired from the company some years previously, taking with him the knowledge of how to retrieve the information sought by the designer (though he was sure that is was held "somewhere in the central files of the group"). We also encountered many examples of successful retrieval. The first, and most common type was for individual designers to recall from memory previous work in which they were involved or components and solution options (and occasionally, critiques) that they had experienced. The second type was essentially managerial, in that design team managers often made suggestions as to the information designers might seek to solve particular problems as well as suggested ways of locating relevant information.

The key point emerging from these observations of reuse is that none of them is based upon retrieval from existing repositories. This negative result must be seen in the light of both the designers' willingness to undertake documentation, and also the failures of existing technologies to facilitate either the encoding or retrieval of documentation. In all the companies, encoding and retrieval of design information were separated by time and device from ongoing design activity. Two important design considerations are: first, that a system to promote reuse needs to make encoding and retrieval contemporaneous, and second, that designers require data-oriented encoding support (e.g. recording of component and project names) to supplement process encoding elicited through design rationale.

2.3 Situated Design Activities

A key issue was to understand the consequences of sub-optimal reuse strategies such as satisficing. The results of our ethnographic research contrast markedly with previous studies of individual and de-contextualized designers (Ball et al., 1997). Almost all team-based episodes reflect a motivated attempt to generate and evaluate multiple solutions options. For example, a major function of review meetings was for the design team to critically appraise alternative design concepts. Experienced designers entered review meetings armed with alternative design options for discussion together with detailed knowledge of their associated costs and benefits. The role played by the team manager was also important, as a safeguard against premature commitment to single solution options. For example, in one session, where a project champion is describing the unsatisfactory aspects of a solution that they are committed to, the team manager interjects "I think the thing to do is look at all the other options". This simple interjection is striking as it is only the third statement that he made in the first 30 minutes or so of the session. Thus, an important outcome is the need for the system to act as a 'surrogate manager', encouraging designers working in individual contexts to consider alternative options.

Another key question concerns the currency of reuse, particularly for rationale-based design information. Our analysis was driven by the goal of identifying the nature and size of a design episode. From inspection of the transcripts, shifts in *question* focus were an apparently natural transition point between episodes. Designers work upon criteria and options in parallel with pursuing a specific question, in what we term a *focus constellation*.

Identification of the focus constellation gives us a framework for choosing between different approaches for capturing design rationale. The two most widely reported of these are Issue Based Information Systems (Conklin & Begeman, 1988) and Questions, Options and Criteria, or QOC (MacLean et al., 1991). While either representation is feasible, the QOC notation was chosen for Desperado for two reasons. First, the QOC notation is consistent with our observation of the focus constellation as defining a design episode. Second, evaluation of options in relation to plausible alternatives is fundamental to QOC, and is especially useful for reuse as it specifies not only the reasoning behind an artifact but also offers a host of plausible alternatives which can be reused, thereby challenging satisficing behaviours. MacLean et al. propose a set of nine heuristics that capture advice on optimal design as practised through concurrent design rationale elicitation. One of our objectives for system design was to implement these heuristics, either implicitly through the processes of encoding and retrieval, or explicitly through system prompts.

3 Desperado: An Overview

Desperado is an environment for component encoding and design rationale elicitation during an ongoing design episode. Design activities are supported by provision of automatically selected episodes or design object names from a database of previous episodes. The basic system structure is shown in Figure 1.

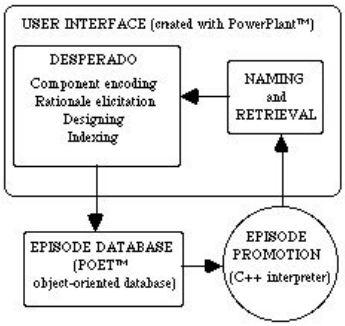

Figure 1: Users encode in Desperado using names retrieved from stored episodes. Episodes that inform design are retrieved concurrently by the promotion interpreter recognizing the state of the current episode.

3.1 The Episode Cycle

We chose the episode as the unit of encoding, defined as pursuing a focus constellation centred around a single design question. Our approach contrasts with reuse systems which take the component as the unit of encoding (Duffy & Duffy, 1996). Sectioning of episode by focus constellation gives a manageable chunk size for indexing reuse information. For example, two weeks of data from the aerospace company gave us 50 episodes that subsequently seeded the database for that company.

Segmenting episodes by focus constellation affords considerable advantages since Desperado can support structured, goal-oriented shifts in activity which we argue are a hallmark of design expertise. Furthermore, it makes explicit the elicitation of design rationale. Pursuit of an episode is the place at which the encoding of component-based and rationale-based information and the retrieval of potentially valuable previous episodes take place. The processes of encoding, design and retrieval are therefore interactive: encoding is supported by using previous episodes to supply terminology for naming, design is supported by prompting through system-initiated retrieval of episodes that are likely to be of value, and retrieval is guided by the current state of information encoded by the designer. The point is for all aspects of reuse, from the designer's perspective, to be singular and seamless.

3.2 Guided Encoding

Encoding is guided by Desperado in three ways. First, it offers a sequenced dialogue, in which design information is elicited in four main phases:

1. data-oriented encoding (e.g. project, user, component information);
2. stage (e.g. requirements vs. conceptual vs. detailed design) and scope (e.g. project-specific, organisational, standards);
3. focus constellation (i.e. QOC information); and
4. location of documentation (e.g. CAD files, requirements specifications).
5. The value of a procedural dialogue is in providing a seamless environment for solution and rationale encoding during ongoing design and in ensuring that encoding is continuous rather than postponed or only from partial aspects of ongoing work.

Second, Desperado offers a naming window (see Figure 2) in which labels supplied in previous episodes for the current encoding event are promulgated using a prioritising mechanism described below. The user clicks on a relevant label in the naming window which places the label in the current episode event text box, which can subsequently be modified if necessary. The naming window addresses two encoding problems, enhancing naming consistency and reducing data entry requirements.

Third, encoding is managed by system prompts, in which the exploration of one event (e.g. a solution option from a previous project) elicits a prompt to explore related events (e.g. criteria concerning that option, or alternative solution options). In essence, prompts serve as surrogate managers to encourage designers to encode reasons behind choices, a key factor identified in our examination of existing documentation practices.

3.3 Guided Retrieval

Rather than relying solely on user-initiated retrieval, Desperado prompts the user as to when, what, and from where, to retrieve episodes during an ongoing design process. This is done through provision of a retrieval window (see Figure 2), in which episodes that are deemed to be of potential value are promulgated continuously. Episodes are promulgated to a 'retrieval window' and refined continuously during an episode cycle. There are three modes of episode retrieval. First, users can select a re-use class name (e.g. 'the

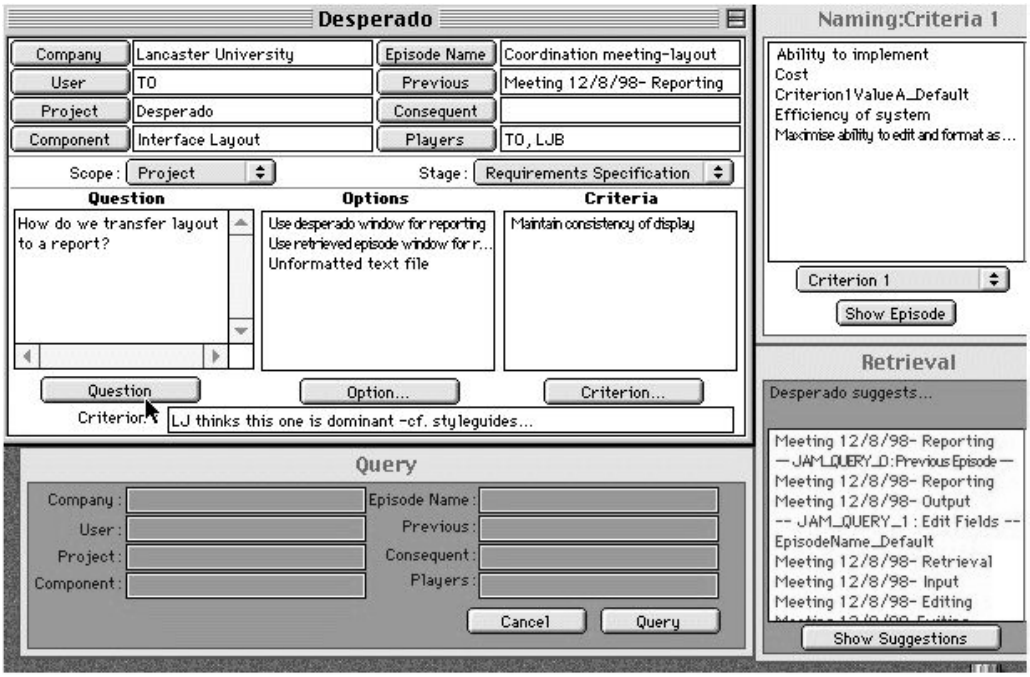

Figure 2: The Desperado window presents the encoding cycle, specifying Company, User, Project, Component, Episode name, Previous episode, Consequents, Players, Scope, Stage and QOC components. The user steps through these with a keystroke command. The Naming window offers a list of event labels from previous episodes that can be selected for and modified in the event text boxes in the Desperado window. The Retrieval window provides a list of system-prompted priority episodes. The user clicks on one for retrieval, and a summary is shown (see Figure 5) along with records of the location of additional documentation (e.g. CAD files). The Query window can be used to search for episodes containing specific terms.

T900 project') from the list of unique names, and view a list of episodes that invoke this name. Second, the user can use Boolean queries to search for episodes. Third, a set of surrogate manager suggestions are made available and continuously updated. In all cases, episodes are prioritised for display so as to maximise their potential *impact*.

The mechanism for promoting previous episodes in the retrieval window and for promoting relevant labels in the naming window are identical. An interpreter (written in CodeWarrior C++) uses prioritization data to rank order previous episodes or event names from within previous episodes. There are five sources of prioritization data: time of episode, defaults ('must see' episodes specified during encoding or querying), key word matches, frequency of retrieval, and weightings derived as a by-product of design rationale elicitation. The interpreter contains an algorithm that evaluates the values of promotion data from each source for each episode. The current algorithm treats data from each source as being of equal value (though see control flexibility, below). However, a re-designed interpreter currently under construction uses a simple learning algorithm to assess the success of each promotion data source for each user, and increments or decrements its relative value accordingly. Part of maintaining *impact* is to introduce perturbation into prioritization, so that designers do not simply revisit the same set of episodes. Therefore, prioritization data can work in two directions, depending upon the stage of ongoing design. For example, frequency of retrieval data can be used to promote or demote frequently retrieved episodes depending upon whether or not promoted episodes are regularly retrieved, and infrequently retrieved or 'lost' episodes can be promoted after set time periods both within and between projects.

One of the key types of promotion data are user-derived weightings elicited during encoding. This is a novel addition to design rationale methods. Weightings consist of evaluations by users of the extent to which an episode event exemplifies a fixed set of general criteria. The criteria are attributes (e.g. novelty, quality, standard practice), alarms (unexpected problem, pre-technological idea) and dependencies (pre-conditions and implications). Users

select general criteria as relevant markers. This does not capture a full rationale for each criterion judgement, but it elicits valuable retrieval cues while minimising the requirement to provide justifications.

As well as being sensitive to the promotion data within previous episodes, the interpreter additionally prioritises previous episodes that are impactful for the current encoding state, as well as the stage and scope of an ongoing episode. For example, different types of reuse information are appropriate in each phase of design. The presentation of questions or criteria-related information in the phase of problem understanding facilitates the orderly decomposition of a complex design problem, whereas the early prompting of option-related information. Thus, Desperado promotes episodes that have heavily weighted questions and criteria early in the design process, and demotes episodes that have only weighted options.

3.4 Control Flexibility

An outcome of our iterative prototyping approach to the development of Desperado has been the finding that system-initiated encoding, retrieval and prompts (e.g. suggestions and warnings regarding un-encoded elements) need to be flexible. For example, the validity of these mechanisms changes depending on whether system use is concurrent during an ongoing design episode or retrospective. Thus, we have added switchability to many aspects of the system. For example, users can choose prioritization to be either entirely system-determined or a user-specified combination of weighting, key-word, date/time, default or simple alphabetical ordering.

4 A Walk-through of an Episode

We have been using Desperado to document our own project. This has lead us to modify later versions of the system in the light of our own project needs, but has also demonstrated the value of system-initiated retrieval. This walk-through describes encoding that occurred during a meeting between the first and third authors, who were discussing changes that were needed for the layout of the user interface. Figures 2–5 show some of the episode data that were encoded and retrieved during this meeting. Prior to this meeting, Desperado had been used in a two-hour meeting where all five project members attended, during which 13 episodes were encoded, and then by the first author working on his own on a specification for reporting functions (i.e. a facility for users to collate reports over projects, users, components etc.) in which two episodes were encoded. Immediately prior to the current episode, the first author used the database to encode an episode pertaining to another research project.

Figure 3: The Criterion capture dialogue. After stating their own episode specific criterion (e.g. by selecting a previous criterion from the Naming window), the user checks those fixed criterion weights that apply.

On launching the system, the main encoding window (labelled Desperado) was shown, along with the naming and retrieval windows. The only components of the encoding window that were visible were the data items *company* and *user*, whose text boxes showed the encodings from the previous episode ('Lancaster University' and 'TO'). The naming window showed a list of all the other company names in the database prioritised according to frequency of selection in previous episodes by TO. Since neither needed modification, the user (TO) moved forward (by keystroke selection) and the next items (*project* and *component*) were shown with defaults to TO's previous project ('Reasoning experiment' and 'Timing software'). The naming window showed a list of all the other project names, prioritised by the frequency of TO's involvement. Since the project had changed between episodes, the user selected 'Desperado' (the most highly prioritised label) from the naming window. At this point the *component* text box defaulted to 'Interface layout'. This was not the previous component worked upon by TO in the Desperado project, but was promoted on the strength of other data, namely the frequency with which TO had encoded episodes in the first meeting to do with the interface layout, and the weightings that focus constellations within these episodes received. This was fortuitous, since the component that TO and LJB were engaged with concerned the interface layout.

The episode continued with increasingly sensitive defaults being selected by the system interpreter for event text boxes, and with continual updating of the naming and retrieval windows. At the stage of encoding the focus constellation (effectively the point at which design rationale is elicited), the designers (TO and LJB) specified the question that formed the focus of the episode ("How do we transfer layout to a report?"). A number of options were suggested verbally in rapid succession. An

attempt to encode the first of these in Desperado prompted a system message to encourage the user to consider criteria before engaging in further option specification (see Figure 4). The system message can be overridden, or the advice followed, depending upon user needs. The designers switched their attention to considering criteria for selecting between options, and developed two key criteria (flexibility for output for text-editing reports etc., and consistency of display within Desperado). After the options were added, a discussion arose as to which of these criteria was the more important.

To resolve this issue, the designers looked at the retrieval window and saw that the highest promoted episode in the window was one from the first meeting that mentioned 'reports'. On selection of this episode (see Figure 5) and browsing through the additional documentation (whose file location was given on retrieval) they found that a similar discussion had taken place at that meeting, in this case with respect to the consistency or flexibility of formatting within a report itself. Using the retrieved notes the designers were able to see that in this instance having flexible output outweighed the need for display consistency, especially since most users will view reports outside Desperado itself. The episode then continued to completion, which was marked by the recognition that a new question had arisen, necessitating the start of a new episode. By stepping through to the end of the cycle, the episode was stored in the database and its data became available for default and episode promotion for the next episode.

Figure 4: A prompt to encourage the consideration of criteria. The prompt is shown when the user proceeds to state options without encoding any evaluative criteria.

Not all episode encoding cycles work as smoothly or demonstrate so convincingly the added value of the system-initiated retrieval. However, we have engaged in a number of single-session evaluation trials with professional designers at the companies studied in the ethnographic study. In general the system has been favourably received. Though a number of modifications were suggested by each design group we visited, these did not implicate the basic functionality of the episode cycle and the guided encoding and retrieval. Typically they involved things like replacing a generic 'waterfall' model of design stages with company-specific stages.

Figure 5: Display of a retrieved episode. The user can select buttons to get further detail (e.g. notes, locations of documentation and weighting information).

5 Conclusions

Desperado provides a single environment in which designers encode component-related and design rationale information, whilst making use of system-initiated episode retrieval to guide these encodings and to promote effective exploration of the design space. In developing Desperado, we have endeavoured to minimise the effort required of the designer in encoding and retrieval, whilst recognising that to remove these activities entirely from user control is to hide a crucial aspect of design.

We have carried out single-session evaluations of Desperado with individual designers, and are using the system continuously in our ongoing work. Some measure of the effectiveness of the system is that we captured over 400 episodes in just five weeks of design work on the project. To fully judge the effectiveness of our approach will entail a longitudinal study of Desperado in professional design environments. Before we commit designers to using a new system, we need to validate it in a less commercially sensitive environment. To this end we are currently evaluating the use of the system by postgraduate Engineering students at Lancaster who are using Desperado to document ongoing assignments.

Acknowledgements

This research was supported by an ESRC Cognitive Engineering Programme grant (L127251027). We are grateful to the managers and designers in our end-user companies for their extensive assistance. We also thank Gary Spiers for his assistance with the programming.

References

Ball, L. J. & Ormerod, T. C. (to appear), "Applying Ethnography in the Analysis and Support of Expertise in Engineering Design", *Design Studies* *****VOLUME*****(***NUMBER***), ***PAGES***.

Ball, L. J., Evans, J., Dennis, I. & Ormerod, T. C. (1997), "Problem Solving Strategies and Expertise in Engineering Design", *Thinking and Reasoning* **3**(***NUMBER***), 247–70.

Buckingham Shum, S. (1996), Analyzing the Usability of a Design Rationale Notation, *in* T. P. Moran & J. M. Carroll (eds.), *Design Rationale: Concepts, Techniques, and Use*, Lawrence Erlbaum Associates, pp.185–215.

Conklin, E. J. & Begeman, M. L. (1988), "gIBIS, A Hypertext Tool for Exploratory Policy Discussion", *ACM Transactions on Office Information Systems* **6**(***NUMBER***), 303–31.

Duffy, A. H. B. & Duffy, S. M. (1996), "Learning for Design Reuse", *AI-EDAM* **10**(***NUMBER***), 139–42.

Gero, J. S. (1990), "Design Prototypes: A Knowledge Representation Schema", *AI Magazine* **11**(***NUMBER***), 26–36.

Grudin, J. (1988), Why CSCW Applications Fail. Problems in the Design and Evaluation of Organizational Interfaces, *in* D. G. Tatar (ed.), *Proceedings of CSCW'88: Conference on Computer Supported Cooperative Work*, ACM Press, pp.85–93.

MacLean, A., Young, R. M., Bellotti, V. M. E. & Moran, T. P. (1991), "Questions, Options and Criteria: Elements of Design Space Analysis", *Human–Computer Interaction* **6**(3-4), 201–50.

Human–Computer Interaction — INTERACT'99
Angela Sasse and Chris Johnson (Editors)
Published by IOS Press, © IFIP TC.13, 1999

Author Index

Human–Computer Interaction — INTERACT'99
Angela Sasse and Chris Johnson (Editors)
Published by IOS Press, © IFIP TC.13, 1999

Keyword Index