Computers and People Series

Edited by
B. R. GAINES and A. MONK

Monographs

Communicating with Microcomputers: An introduction to the technology of man–computer communication, *Ian H. Witten* 1980

The Computer in Experimental Psychology, *R. Bird* 1981

Principles of Computer Speech, *I. H. Witten* 1982

Cognitive Psychology of Planning, *J-M. Hoc* 1988

Edited Works

Computing Skills and the User Interface, *M. J. Coombs and J. L. Alty (eds)* 1981

Fuzzy Reasoning and Its Applications, *E. H. Mamdani and B. R. Gaines (eds)* 1981

Intelligent Tutoring Systems, *D. Sleeman and J. S. Brown (eds)* 1982 (1986 paperback)

Designing for Human–Computer Communication, *M. E. Sime and M. J. Coombs (eds)* 1983

The Psychology of Computer Use, *T. R. G. Green, S. J. Payne and G. C. van der Veer (eds)* 1983

Fundamentals of Human–Computer Interaction, *Andrew Monk (ed)* 1984, 1985

Working with Computers: Theory versus Outcome, *G. C. van der Veer, T. R. G. Green, J-M. Hoc and D. Murray (eds)* 1988

Cognitive Engineering in Complex Dynamic Worlds, *E. Hollnagel, G. Mancini and D. D. Woods (eds)* 1988

Computers and Conversation, *P. Luff, N. Gilbert and D. Frohlich (eds)* 1990

Adaptive User Interfaces, *D. Browne, P. Totterdell and M. Norman (eds)* 1990

EACE Publications

(Consulting Editors: *Y. WAERN and J-M. HOC*)

Cognitive Ergonomics, *P. Falzon (ed)* 1990

Psychology of Programming, *J-M. Hoc, T. R. G. Green, R. Samurçay and D. Gilmore (eds)* 1990

Adaptive User Interfaces

Edited by

DERMOT BROWNE

KPMG Peat Marwick Management Consultants
231 Blackfriars Road, London, SE1, UK

PETER TOTTERDELL

MRC Social and Applied Psychology Unit,
Department of Psychology, University of Sheffield,
Sheffield, S10 2TN, UK

MIKE NORMAN

Department of Computer Science,
University of Hull, Hull, HU6 7RX, UK

ACADEMIC PRESS

Harcourt Brace Jovanovich, Publishers
London San Diego New York
Boston Sydney Tokyo Toronto

ACADEMIC PRESS LTD.
24/28 Oval Road,
London NW1 7DX

United States Edition published by
ACADEMIC PRESS INC.
San Diego, California 92101-4311

British Library Cataloguing in Publication Data

Is available

ISBN 0-12-137755-5

Printed in Great Britain by St Edmundsbury Press Ltd,
Bury St Edmunds, Suffolk

Authors

Editors

Dermot Browne is now Executive Consultant at KPMG Peat Marwick Management Consultants, London and was formerly Technical Consultant at Data Logic, Harrow.

Peter Totterdell is Research Officer at the MRC/ESRC Social and Applied Psychology Unit, University of Sheffield, and was formerly Senior Research Engineer at STC Technology Ltd, Harlow.

Mike Norman is Rank Xerox Professor of Information Technology in the Department of Computer Science at Hull University.

Contributors

Stewart Anderson is a Lecturer in Computer Science at Edinburgh University and was formerley Lecturer in Computer Science at Heriott Watt University.

Ebby Adhami is a Consultant with Ernst & Young, London and was formerly Technical Consultant at Data Logic, Harrow.

Liz Boyle is a Research Fellow at the Human Communication Research Centre, Glasgow and was formerly Research Associate at the Scottish HCI Centre.

Paul Rautenbach is a Project Engineer at Computer Vision, Cambridge working on CAD/CAM systems and was previously Senior Research Engineer at STC Technology Ltd, Harlow.

Dave Riches is Senior Software Engineer at STC Technology Ltd, Harlow and was formerly Senior Research Officer at Essex University.

Alan Wilkinson is Senior Research Engineer at STC Technology Ltd, Harlow.

Preface

It is widely believed that "everyone should be computer literate" and as a consequence vast human and financial resources are being expended internationally on training individuals to adapt to using computers within their lives. The premise of the research that is reported in this volume is that "computers should be user literate". Comparatively small resources have been devoted to building flexibility into computers; flexibility which would enable computers to adapt to the diverse needs of the individuals who use them.

The research reported here was part of a project which looked at the problem of designing adaptive computer interfaces. An adaptive computer intereface is one which can change its behaviour to suit the individual or group using it. This can range from an interface which switches fonts to suit the preference of a user, to an interface which builds and evaluates a model of the user in order to improve the effectiveness of communication between the computer and the user.

The Adaptive Intelligent Dialogues project, also known as the AID project, was a four-year project which researched and developed techniques for designing and building adaptive computer interfaces. The AID project was part of the United Kingdom's Advanced Technology Alvey Programme. The collaborators on the project were: STC Technology Ltd, Data Logic Ltd, British Telecommunications plc, The University of Hull, The University of Strathclyde and The University of Essex.

Special mention should go to the project managers Phil O'Donovan and his successor Paul Cooper of STC Technology Ltd who managed to keep the project on course. We would also like to thank Tom Stewart of System Concepts Ltd who was the external monitor for the project. His contributions at workshops were always very positive, apt and well received.

We would like to acknowledge the contribution of all those that worked on the project, whether it was for four weeks or four years. In keeping with the spirit of the venture, we will simply list their names without mentioning affiliation, position or time on the project:

Ebby Adhami, Jim Alty, Bruce Anderson, Stuart Anderson, Farhat Arshad, Bob Benton, Len Botacci, Peter Boucherat, Liz Boyle, Pamela Brody, Andrew Brooks, Dermot Browne, Nigel Cliffe, Ian Clowes, Gilbert Cockton, Ian Cole, Stephanie Cookson, Martin Cooper, Paul Cooper, Tony Cox, Colin Davenport, Graham Dunkling, Tony Fountain, John Friend, Andrew Hockley, Colin Hopkins, Bernard Horan, Safwan J'Affra, Peter Jones, Neil Lawrence, Richard Lawrence, Graham Leedham, Pat Leisner, Hamid Lesan, Andrew Marshall, Terry Mayes, Phil McEachen, Swapan Mitra, Linda Moxey, Dave Moynaghan, Brian Murphy, Mike Norman, Phil O'Donovan, Robin Pyburn, Paul Rautenbach, Dave Riches, Colin Robertson, Osnat Ron, Nigel Seel, Briun Sharratt, Andrew Stewart, Mike Thornton, Peter Totterdell, Robert Trevellyan, Alan Wilkinson, Romualdas Viliunas, Albert Wong.

We would also like to acknowledge the support provided by Alvey and SERC in part funding the AID project. STC Technology would also like to acknowledge International Computers Ltd for their support.

One caveat, although much reference is made to work from within the project, the thoughts and ideas expressed here are the responsibility of the editors and authors alone and do not necessarily reflect the views of all the individuals and organisations involved in the project. Some of the ideas reported here have been reported elsewhere - in papers, journals, workshops and conferences - but this volume is intended to bring together the results of the project under one cover.

It is hoped that the work will be of interest to anyone who is seriously interested in Human Computer Interaction; both the issues and the practice.

Chapter 1
Introduction

P. Totterdell

The following chapter is an introduction to the Adaptive Intelligent Dialogues (AID) project which was the source of the research reported in this book. The chapter begins by describing the objectives, participants, and development of the project. During the course of the project a number of adaptive interfaces appeared as a result of work outside the project. Some of these interfaces are briefly reviewed in this chapter. The editors then describe what it is they hope to achieve by disseminating the results of the AID project to a wider audience, and hence they describe what you the reader might hope to gain. The chapter ends with a brief summary of the contents and rationale of other chapters.

This book is the result of a four-year collaborative venture between three industrial and three academic partners who, with the help of government funds, allowed a group of researchers to tackle the problem of designing adaptive user interfaces for computers. An adaptive user interface is an interface which can change its behaviour to suit an individual or group of individuals.

Our account concentrates on the technical outcomes of this venture but an equally interesting account could have described the sociological development of a project of this type. Indeed we might have chosen to illustrate the processes of adaptation by referring to the course of change within the project itself. But we start more mundanely with some details of the project.

The AID Project

The Adaptive Intelligent Dialogues (AID) project was part of the United Kingdom's Advanced Technology Alvey program which sponsored collaborative computing and information technology projects involving both industry and academia in the UK. The AID project was the largest project of the Man-Machine Interface (MMI) section of the Alvey programme. The project ran from October 1984 to September 1988. The collaborators on the project were: STC Technology Ltd, Data Logic Ltd, British Telecommunications plc, The University of Hull, The University of Strathclyde and The University of Essex.

The project was originally set up to "research the techniques appropriate to the development of user interfaces that adapt to a particular user's ability and knowledge of a given system". The term adaptive referred both to self adaptive and user tailored systems (terms which will be explored in depth later on). The justification for the project was that with computers becoming more widespread and having to accommodate a wider range of users, it was no longer satisfactory for designers to aim the level of interaction at an average user because nobody conformed to the stereotype of the average user.

The specific objectives of the project were to:

i) Research the principles underlying intelligent adaptive interaction.

ii) Build software exemplars to demonstrate this research.

iii) Evaluate the effectiveness of the exemplars.

iv) Produce tools to assist in developing adaptive user interfaces.

The project was divided into three phases. The first phase developed an adaptive interface to an electronic mail system using the (at the time) state-of-the-art tools and techniques. The second phase of the project researched the key issues of adaptive interaction. As well as producing theoretical frameworks for understanding adaptation, the second phase also produced a number of software exemplars. The third phase of the project consolidated the research by building improved exemplars which illustrated general techniques for constructing adaptive interfaces. The final phase also developed tools to facilitate future development of adaptive interfaces.

Phase 1

Phase 1 of the project designed and built an adaptive front end to the British Telecom electronic mail system Telecom Gold. It was hoped that the existing interface to the application would provide a suitable baseline against which to measure the performance of the adaptive interface. The application also had the advantage that it was a "real world" rather than a "toy" application and would therefore be a good test of the adequacy of current Human Computer Interface (HCI) wisdom and computer technology.

The adaptive interface was built in a single design, build and test cycle, with no opportunity for redesign. A description of the interface architecture and its components can be found in later chapters but essentially it comprised a dialogue controller (which interacted with the user), a user model and an application expert. The application expert interfaced the whole system to the application via a modem. The phase 1 system adapted along a number of dimensions, the most important of which was the level of help it gave a user. The system was effective in as much as it provided user assistance related to user competence, but this was at the expense of unacceptably high response times.

It has been said that the sum of the components in the phase 1 system was greater than the whole. And certainly one of the

successes of the system was the design of the application expert. With hindsight one might question whether the design effort required to interface the system via a slow and noisy telephone line to an application with unknown states might have been better spent on designing the user interface to an easier application. But as with any research, the spinoffs are often as valuable as they are unanticipated; and the application expert was a valuable spinoff.

Evaluation of the phase 1 system proved difficult and it became apparent that adaptation is of superficial value unless the non-adaptive components of an interface are also of sufficient design quality. In an effort to make a controlled comparison with a non-adaptive version of the interface, the designers of the phase 1 system had neglected this requirement and had built adaptive facilities on top of a rather poor user interface.

Phase 2

Phase 2 of the project ran from January 1986 to September 1987. The original objective for phase 2 had been to produce a commercially viable exemplar. However, following the experience of phase 1, it was clear that phase 2 needed to address some very fundamental issues concerning the characteristics and classification of adaptive systems. This included looking at the use of the concept of adaptation in other disciplines such as biology and cybernetics, and led on to a clarification of its usage within the field of HCI.

In phase 2 adaptation took on the meaning of an approach to design rather than being seen as a universal architecture. In particular it was viewed as a framework for deferring design decisions. A theory- based approach to the design and evaluation of adaptive systems via metrics also emerged in this phase.

It was decided after a few months of the phase that the project needed a single domain for its exemplars and experiments. This was seen as a way of bringing various strands of work together. The project chose document preparation as its domain because it seemed to offer problems requiring adaptive solutions. The exemplars which followed seemed to offer alternatives to the cognitive user modelling approach which had dominated the first phase. They demonstrated that in some cases adaptation can be provided without a sophisticated model of the user. This is important given that the bandwidth of

communication between user and interface is sometimes too narrow to justify an elaborate user model.

Phase 3

The final phase of the project used the groundwork of phase 2 to produce improved exemplars. Again the exemplars were within the domain of document preparation and they were chosen to illustrate more general techniques for constructing adaptive interfaces. The exemplars showed considerable diversity in their demonstration of adaptation. For example: the Task Organiser adapted to linguistic context, Groupie 2 adapted to the help preferences of a community of users, Reference Information Provider adapted to relevance, and Adaptive Menu Structure adapted to frequency of use. These exemplars as well as others are described in later chapters, especially Chapter 5.

Phases 1 and 2 had shown that it was hard to identify user or task variability, the sources for adaptation, in the course of using conventional system design techniques. What was needed was a method of systematically assessing situation specific requirements to see whether or not they could profitably be realised at run-time by an adaptive interface. The method also needed to address the practicality of using an adaptive solution. The computer has only limited access to the user's situation and can therefore only detect and monitor a limited set of stimuli. A method called MAID (Methodology for Adaptive Interface Design) was developed to satisfy these needs. This was complemented by using the evaluation experiences of the project to devise a detailed protocol for formative and summative evaluation of adaptive interfaces.

Phase 3 also delivered some tools to support the construction of adaptive interfaces. The ideas of application modelling developed in phase 1 were later incorporated into a tool (HIFI) for integrating user interface design with system functionality. And two other tools, Deferred Design Tool (DDT) and Task Description Language (TDL) were developed to support the project's design method.

Management

Research on the scale of the AID project requires good management. Whilst good management cannot guarantee success to a research project, bad management usually guarantees failure. And in as much as the results reported here are a measure of success or failure, it may be of use to record some of the conditions which were the context for that outcome.

The work of the project was divided into "workpackages" with personnel from different sites participating in workpackages which accorded with their interests. Meetings to organise these workpackages were arranged as necessary. Every quarter year a review workshop was held for all staff on the project. The responsibility for organising workshops rotated around sites. The workshops were an opportunity for discussing general progress on the project and for organising the following three months of work.

Management meetings were held on a monthly basis and involved a project manager from each site. These meetings gave way to technical management meetings in the later stages of the project. These involved more junior and technical interested representatives from each site. The project was reviewed internally by a monitoring officer chosen every three months from the staff of the site organising the next workshop. Externally the project was reviewed both by an independent monitoring officer who attended project workshops and reported to the funding body, and also by major review with an Alvey Committee at the end of each phase.

The three phases of the project were important in structuring the research. Phase 1 of the project, as well as providing valuable technical insights, also served to facilitate collaboration within the project. Working in cross site design teams to tight deadlines on concrete problems undoubtedly helped to break down organisation barriers. Evidence of this cohesiveness came from workshop guests who commented that they could not tell who came from which site. The only drawback, however, to this style of research is that it discourages divergence and it prevented the project tackling fundamental questions until phase 2.

In contrast, the more *laissez faire* style in the initial stages of phase 2 meant that there was little prospect of integrating the

various lines of work that were happening at different sites. And some of the work was on the margins of usefulness to the central adaptive interface problem. Hence, the decision to choose a common problem domain ie document preparation.

In the project's original conception, it was envisaged that the industrial partners would contribute their experience and expertise of system building to the project whilst the academic partners would contribute more to the theory and evaluation issues. In reality this proved to be an artificial separation. Much of the "thinking" came from the industrialist and much of the "building" from the academics. In part this was a reflection of recruitment outcomes at the different sites. But perhaps it also exposes the myth of the industrial/academic stereotypes.

A project of the size and length of AID develops its own culture in the form of personal allegiances, schools and even generations of thought. Only a small core of researchers were part of the project throughout its duration and even then the project occupied a varying percentage of their time. A full list of the people that worked on the project is given at the start of the book. It would be an interesting exercise to look at how their individual interests and time on the project shaped its course.

Collaboration on a project such as AID should not be measured simply in terms of its deliverables. Some of the other benefits include:

The acquisition of a wider skill base for participants.

Increased contacts in relevant fields and communities.

An increase in status for collaborative groups within their own organisations.

Increased recruitment of skilled human factors staff.

A wider awareness and appreciation of human factors issues.

The increased adoption of a user centred design process within the organisations.

However collaboration does have its costs, particularly in terms of its resource intensity and administrative overheads.

Other Adaptive Interfaces

The AID project proposal recognised that adaptiveness is required in the interface because no single fixed solution is suitable for all users or even one user over a period of time or range of applications. The interface needs to adapt to the user's changing skills and requirements, and the assistance provided by the system needs to be relevant to the tasks the user is performing. This requirement for adaptive interfaces had been recognised for some time (e.g. Edmonds, 1982) and yet there have been relatively few examples of adaptive interfaces. Here we provide a brief review of the work of others in this area and name some of the interfaces which do exist.

Edmonds (1982) described three modes of adaptation: adaptation by a specialist, adaptation by a trained user, and adaptation by any user. The SYNICS system (Edmonds and Guest, 1978) translates strings according to specified transformational rules and is an example of a system which can be used in the first mode, for example to allow a specialist to produce alternative error messages. Edmonds also gives early examples of systems in the other modes. All of these modes are examples of tailorable systems, that is they are adapted by something outside the system whether it be the designer or user. But Edmonds also introduced the concept of a self-adaptive interface.

A self adaptive interface changes automatically in response to its experience with users. Three types of self-adaptive interface were distinguished. Those which collect information about the user and tailor the interface responses either during a session or between sessions. Those which identify a user as belonging to a particular category and set the interface's parameters accordingly (once only). And those in which the interface doesn't change but performance improves, for example by dealing with errors more quickly. Early examples of all three types are given by Edmonds. Innocent (1982) also described some of the basic principles of self adaptive user interfaces. He gave a possible architecture for a self adaptive user interface incorporating an expert modifier which monitors and evaluates user and system behaviour with the purpose of reshaping the 'soft facade' of the interface.

Some of the adaptive interfaces to appear in recent years include the following:

Connect (Alty, 1984a). Connect is an adaptable dialogue delivery vehicle that achieves adaptability through a production system which monitors user interaction and consequently opens or closes arcs in the dialogue network.

Dialog (Maskery, 1984). Maskery undertook some experiments with an adaptive interface to a package of statistical tools called Dialog. The interface had three levels of dialogue, forced choice system led, free choice system led, and free choice user led. Users experienced difficulties when they were transferred to the user led interface.

Adaptable help manual (Mason & Thomas, 1984). The adaptive part of the interface models the user by quantifying their experience of the system using a weighted set of user descriptive variables. The model then determines what type of help should be retrieved for the user.

Monitor (Benyon, 1984). Monitor selects dialogue scripts based on information about the user collected in a user model. For example, the user model records whether or not a user has performed a task before and a more verbose script is selected if the user has not previously performed the task. The prototype of Monitor was in the domain of computer assisted learning but its design was intended to be general purpose. LS-1 (Smith, 1984) uses a genetic algorithm to learn a set of problem solving heuristics. It has been applied to the problems of simple maze walking and draw poker, and has been shown to improve its performance over time in both domains. Its performance at draw poker was better than another program judged to perform at the same level as an experienced human player.

Pal (Pickering *et al.*, 1984). Pal is a communication aid for the disabled which adapts by predicting the stems of words being typed by the user. By allowing the user to accept its predictions it is able to reduce the length of the keying sequence.

Poise (Croft, 1984). The Poise system provides assistance to the users of an office system based on models of office tasks. Poise infers users' tasks from their actions and provides context sensitive assistance.

Document retrieval (Croft, 1984). The document retrieval system uses associative search networks (ASN) which work on feedback from the user to change the set of weights on the system's search strategies.

Experiments have shown that the ASN reliably learns to select the appropriate strategy.

Personalised directory (Greenberg & Witten, 1985). Greenberg and Witten constructed an interface to a menu driven telephone directory system. The interface reduced selection time and error rate by ordering the menus according to the probability of selection which was based on the user's frequency of retrieval for each number.

Adaptive indexing (Furnas, 1985). This system monitors the words that a community of users employ to refer to system objects. It then increases the weight of association between the words and the objects. Where necessary it adds new words to the index. In this way the system increases its chance of successfully recognising subsequent user commands.

We will meet some of these systems again in Chapter 3 when we start to classify adaptive systems.

Aims

Who is this book aimed at? Firstly, it is intended to help anyone that might be considering designing an adaptive interface. Inevitably any such person will cross many of the same hurdles that we did. For example, we had long debates on: the distinction between and relative merits of user tailored and self adaptive interfaces; what is and what isn't an adaptive interface (a very slippery slope this one); good HCI vs adaptation; the best architecture for an adaptive interface; the domain specifity of results; appropriate evaluation procedures etc. Hopefully this book will help to lower those hurdles through its description of frameworks, techniques and experimental results.

The book should appeal not only to those looking for an adaptive solution to a problem but also to anyone that is interested in learning more about the issues and techniqes of HCI. The techniques required for an adaptive interface commonly apply also to non adaptive interfaces. HCI has often been seen to be about building fancy interface gadgets or about choosing screen colours. It is neither ; it is a serious design activity. And designing adaptive interfaces are part of that activity.

At the start of the project, adaptation was seen as something which could be packaged together and bolted on to a system. It was to be an extra facility, something which would differentiate the system from its market competitiors. However, it became clear that adaptation was much more part of the design process than it was a product in itself.

Adaptive mechanisms have evolved to play an important role in human physiology, e.g. the adaptation of the circadian system to changes in environmental time (Hildebrant and Moog, 1988). Adaptive behaviour is also critical in successful human to human interaction. It is perhaps not unreasonable, therefore, to think that successful interaction between computer and human will also require the computer system to exhibit adaptive behaviour. In humans the long term outcome of failure to adapt is disease (Mackay, 1984). There may come a time when we refer to computers which don't adapt as diseased. On this count we have some very unhealthy computers at present! The same preconditions and symptoms of failure to adapt that we identify in humans, such as life event changes, stressors, coping behaviour and health status may also begin to apply to computers. But that is for the future.

We end this particular chapter with an overview of the chapters that follow. But first an apology. Much of the terminology in the field of HCI is used very loosely. Sometimes terms are used interchangeably and sometimes they are used to make distinctions. We haven't helped this situation. For example, we sometimes refer to adaptation as if it were a thing not a process, despite the claim that it is the latter. It is awkward to do otherwise. We also use "interface" and "system" interchangeably. "Methods", "models" and "techniques" too. Probably others. Hopefully the reader will be able to understand our use of terms within the context (we rely on your adaptive capabilities). We have, however, made an effort to restrict ourselves to using the term "adaptation" rather than "adaption". But others in the field also use adaption (Edmonds, 1982).

Overview

Chapter 2. John Bloggs is a rather unpleasant fictional character who employed a team of extremely pleasant user interface designers to provide an adaptive interface. His story illustrates some of the

problems and benefits of adaptive user interfaces. Some of these problems can be solved by using the frameworks described in the chapter. There are frameworks for: identifying differences in the end user population that might be reasons for using adaptation; identifying the purposes or likely benefits that might be obtained; and identifying changes that could be made to the interface. A set of metrics and a methodology for designing adaptive interfaces are then described.

Chapter 3. This chapter takes a step back from the practicalities of designing adaptive interfaces to adopt a theoretical stance on the problem. Adaptation is viewed as a set of alternative design choices for relating a system to its environment. The game "Prisoner's Dilemma" is used to construct a taxonomy of adaptive system design, within which current adaptive systems are classified. Some parallels with evolution are noted and the implications for designers are described. A general architecture for adaptive systems is then inferred from this view of adaptation.

Chapter 4. An interface designer will need to know what methods are available to support the design of an adaptive (or non adaptive) interface. Chapter 4 reviews these methods under the headings of user models, dialogue models, task models and application models. These methods are not restricted to adaptive interfaces and the text should therefore also provide a useful review of the current status of methods in HCI. The specific relevance of a method to adaptive interface design is highlighted where necessary. The application expert of the AID phase 1 system is described in considerably more detail than other methods. This is not because we feel that it is more important than the alternatives (well, perhaps we do!) but because it is a result of the project and therefore needs fuller reporting.

Chapter 5. The methods described in Chapter 4 were general to interface design but could be used as components of an adaptive interface. In contrast, this chapter looks specifically at techniques for providing adaptation at an interface. This includes: genetic algorithms, adaptive scheduling, pattern matching, context and user models. Most of these are illustrated by examples from the AID project. The chapter begins by taking another look at architecture, and shows how three exemplars conform to the general architecture

described in Chapter 3. The chapter finishes on two specific issues, namely adaptation to a group of users and the weighting of evidence.

Chapter 6. Interface designers ought to be able to show that the techniques they are using are of benefit to the user and that they are an improvement on other techniques. This chapter describes the evaluation methods that can be used to evaluate the performance of an adaptive interface. It also describes evaluation methods that can be used to support the design process itself. Many of the methods have been developed to deal specifically with adaptive interfaces, but most will also apply to the evaluation of non adaptive interfaces.

Chapter 7. This chapter summarises some of the lessons of the AID project and looks into the crystal ball for the future of adaptive interfaces.

Chapter 2
Why Build Adaptive Systems?

D. Browne, M. Norman and D. Riches

Two major assumptions underly the building of any adaptive system. Firstly, that there are differences in the end-user population that can provide good REASON for adaptation. Secondly, that there is some PURPOSE or benefit to be obtained. It is insufficient to describe the purpose as one of improving the user interface of the system. Further, given the identifiable user differences and that they can form the basis for purposeful adaptive change, the question remains as to what those changes should actually be? Finally, discussion will turn to the adoption of a suitable methodology for developing adaptive systems.