CONTEXTUALIZING USER INTERFACES FOR COMPLEX SYSTEMS

By

JOHN FOSTER MEECH

A thesis submitted to
The University of Birmingham
for the degree of
DOCTOR OF PHILOSOPHY

School of Psychology
Faculty of Science
The University of Birmingham
September 1999

Abstract

This thesis addresses the difficulties that people have in interacting with complex, computer-based systems. The fields of intelligent and adaptive interfaces and agent-based systems are surveyed and critiqued to identify how intelligent human-computer interfaces can be used to improve interaction. The need to manage context is identified as a key element in intelligent interfaces used to manage complexity. A model of *contextualization* is developed to encompass a range of interface design and implementation paradigms, with the objective of improving the design of dynamic interactive systems. Viewing the process of contextualization as part of the interaction process provides a powerful conceptual methodology for the design of agent-based intelligent user interfaces. A model of contextualization is developed consisting of several components that are intended to promote contextualization in user and interface. An experimental evaluation of these components shows that elements of *dialogue instigation* and *adaptation of the user interface via user preferences* provide the expected performance advantages in both objective and subjective evaluations. In addition, the experiments show that contextualization is affected by factors such as the personality of the user. The interaction of the various components of the model of contextualization is discussed and proposals for future work are presented.

Acknowledgements

This thesis would not have been completed without the help and support of many people. My supervisors, Dr. Mike Harris (Psychology) and Dr. William Edmondson (Computing) have supported me and provided stimulating discussion throughout my graduate work. As I moved through various research positions, they continued this support in what turned out to be more of a marathon than I expected (even allowing for part-time studying).

Many colleagues (too numerous to mention) have encouraged me and taken time to discuss various aspects of this work (and a lot of topics that never made it into the final thesis). To all of you, a heartfelt thanks.

Towards the end of my thesis work, I would like to thank Dr. Sue Abu-Hakima (and later Dr. Roger Impey), my Group Leaders in the Network Computing Group at the National Research Council of Canada. They gave me the time to run and write up experiments, and generally encouraged me to finish. Thanks also to Dr. Dick Dillon and the Faculty of Psychology at Carleton University, Ottawa, Canada for allowing me access to students as I began to run out of experimental subjects. My gratitude also goes to Dr. Steve Marsh at the National Research Council of Canada for reading final drafts of the thesis to make sure it was somewhat coherent.

Finally, I would first like to thank my Mother and Father for their tireless support and encouragement, not only in completing this thesis, but in everything that I do.

Dedication

To my parents, Margaret and Barrie Meech, with love and thanks.

Table of Contents

Appendix B:

List of Illustrations

List of Tables

Chapter 1

Interacting With Complex Systems

1.0 Introduction

This thesis is the product of extensive experience studying large-scale commercial systems and trying to understand how it is possible to reconcile their complexity with the capabilities of the people who must use and operate them. Consequently the goals of this thesis are to examine the commonalties shared by a wide range of complex, dynamic systems and to address emergent complexity as perceived by the user. The motivation is to identify the factors that make it difficult for users to interact with complex systems, and provide a *theory* that both explains this and suggests ways of addressing the design of future systems to minimise problems.

1.1 Complex, Real-Time Systems

It has become clear that some human-designed systems have become *complex* and consequently difficult to manage by their users and operators (Cohen et al., 1986). This complexity has been clearly demonstrated by several high-profile accidents, errors in operation and brittleness in system functioning.

One such high-profile accident was the Kegworth Air Disaster (Department of Transport, 1990). In this incident a Boeing 737-400 commercial airliner crashed on approach to East Midlands Airport with the loss of 47 lives. The crew, detecting a problem, shut down the wrong engine and did not realize their mistake until it was too late to take further action. The combination of heavy engine vibration, noise, shuddering and smell of fire was outside the training and experience of the crew. The pilots also failed to re-assess their initial, incorrect, diagnosis of the fault on the approach to East Midlands airport. This was due to high workload, compounded by the co-pilot attempting (unsuccessfully) to re-program the Flight Management Systems (FMS) with the new destination and frequent interruptions by air traffic control. In short, the complexity of the system resulted in the crew being unable to understand its operation to the degree required to correctly interpret the symptoms in the time available, with disastrous consequences. In another example from the world of aviation, in 1985 a China Airlines Boeing 747 suffered a slow loss of power from its starboard engine. The autopilot compensated automatically, preventing the aircraft from yawing to the right, but eventually the autopilot could no longer compensate. The crew, who had not noticed the action of the autopilot, did not have enough time to determine the cause of the problem and take action. The aircraft rolled into a vertical dive for 13500 feet before it could be recovered (Norman, 1990).

Similar control problems exist with other complex systems such as process control, notable for high-profile incidents such as Three-Mile Island (Brooks and Siddall, 1980) and Chernobyl (Bailey 1989). In the Three Mile Island (TMI) nuclear generating incident the major operator error was throttling back on two high-pressure injection pumps to decrease water pressure. This allowed the core to be uncovered and overheat. However, there were several other contributing factors (including a valve stuck open, which was indicated to the operators as being closed), which combined with the fact that the operators did not fully understand the principles of plant operation, only narrowly avoided a core-meltdown. In the Chernobyl incident operators were conducting a test on the nuclear reactor which required several automatic control systems to be disconnected. A range of factors led to explosions and fire within the plant, resulting in fission products being released into the atmosphere. The operators were blamed for being complacent (because of the plant's hitherto good operating record) and for operating the nuclear reactor outside safe limits. Networked process control systems seem to be equally vulnerable; In 1977 New York City suffered a massive and costly power blackout (see Perrow, 1984). The operator of the electricity grid followed prescribed procedures to handle the initial symptoms, but because of two relay failures (of which the operator was unaware) the entire distribution system failed.

It is usually systems which have a high degree of public visibility that receive such publicity, though this does not mean that other systems do not suffer from similar problems. Complex, dynamic systems are becoming increasingly more common and as the previous set of examples shows, automation in such systems is now the norm. In the design of new systems, complexity due to physical size and interconnectedness is being replaced by complexity related to the use of large software programs in controlling systems. For example, the management of telecommunications networks is also becoming problematical (Lucas et al, 1997), as is the usability of the services provided by such networks (Nielsen, 1997). Telecommunications networks encompass elements of both physical size and software complexity, and include safety-critical systems such as Ambulance, Fire and Police dispatch and communication (see Leveson, 1995 for a wide range of examples).

Personal experience of a range of similar systems supports these conclusions. Initial investigations concerned pilots flying commercial and military aeroplanes, and gave an appreciation of the amount of data that must be monitored, the speed at which actions must be taken, and the deadly consequences of making a mistake (Meech, 1992a). Further studies concerned more general process control systems, particularly supervisory control systems (e.g., operators controlling nuclear power plants). Supervisory systems exhibit behaviour similar to modern, automated aircraft: they are very complex, but the user takes a supervisory role rather than constantly interacting with the system (Meech 1992b, 1994a). This intermittent interaction means that for successful interaction to take place, the user should be aware about what the system is doing at any given point in time; information obtained solely from passively monitoring displays. Analyses of a large scale, unique process control system (High Energy Physics experiments) confirmed the importance of providing the user with important information and making sure that interaction was easy and natural, particularly for inexperienced users (Meech, 1994b; Meech, Huuskonen et al., 1995). Further work in telecommunications showed that the control of both system-wide and personal systems share the characteristics of complex supervisory control (Lucas, Meech and Purcell, 1997; Meech and Abu-Hakima, 1998). Telecommunications networks operate as supervisory control systems from the network perspective, but also appear as complex systems to individual users, who must monitor the system for messages and respond to them dynamically.

The common thread running through the systems discussed above is that when things go wrong the users or operators of the systems concerned are usually blamed for making operational mistakes. It is this operational characteristic which is important as a defining element of complexity. One signifier of complexity in this sense is when a system is difficult to control; a difficulty which is usually manifested as "errors" which occur in system operation. Perrow (1984) considers complexity to be an inevitable result of the kinds of systems that operate in *high-risk* environments (*high-risk* in that continued operation is desirable, if not essential, as in aircraft and other continuous processes). However, it is unclear if this complexity is inherent in the system itself, and to what extent this is also dependent on operator knowledge, training, and other factors. For example, many accidents attributed to "operator error" occur in heavily automated systems (see previous examples). It has been widely suggested that such automation often makes systems more complex for the user (Bainbridge, 1987).

The evolution of complex systems has been identified by a number of researchers (Checkland, 1981; Ferry, 1988; Flood and Carson, 1988; Rasmussen and Vicente, 1990). Large systems with many interacting parts are not new; society itself is a prime example of such a system, as are the associated systems of finance, trade, and so on. DeGreene (1991) points out that the real world (as opposed to models of the real world) is fluid, dynamic, turbulent, evolutionary and constantly re-configuring itself. Fischer (1991) reinforces this view, but introduces a characteristic which is more indicative of designed systems, that of *high functionality.* High functionality is a property of systems such as the UNIX computer operating system which has many possible commands that may be combined in many ways. The sheer number of command compositions that are possible means that not even an expert is able to master the entire functionality of the system. In addition, the use of one function can change the system state, which will affect what happens when the next function is performed – transposing copy and delete functions will have radically different results, as will pressurising a reactor vessel without first closing an exhaust valve. An operator must know what functions are possible, as well as how functions can be composed, and this also contributes to complexity.

The introduction of high-functionality computer artefacts into general usage began with consumer products such as video recorders, which are notorious for being difficult to program (and users rarely make use of all their functions). This has extended into more everyday devices with the proliferation of cellphones, Personal Digital Assistants (PDAs), and other portable computing and communication devices. Software systems for commercial usage have functionalities that seem to increase with each new release, as indicated by ever-thickening manuals. However, commercial software is missing an important attribute of the systems previously discussed; that of real-time, dynamic behaviour. In most commercial (e.g., word-processing) software, the user drives the interaction. In controlling an external, physical world (such as an aeroplane) the external world prompts the user to take action – it is not possible to leave the machine to read the manual (nor is it possible to "undo" the last command!).

1.2 Complexity and Dynamic Behaviour

The field of systems analysis has developed to address the difficulty of understanding complex systems (Checkland, 1981). There have been efforts to identify objective system characteristics that contribute to complexity. Flood and Carson (1988) and Yates (1988) view complexity as arising from static system properties (e.g., number of functions, interactions, variables, and so on, along the lines proposed by Fischer). However, Perrow (1984, p.88) points out that limited operator understanding will obviously contribute to how complex the system appears to the operator. Thelwell (1994) and Folleso et al., (1995) extend these complexity criteria to include the dynamic behaviour of the system, as this factor reduces the time that users or operators have to understand the system behaviour and consequently also has a large impact on operator-perceived complexity. Thus complexity can be seen to have both objective and subjective contributing factors.

Weir (1991) divides system complexity into three parts which separate objective and subjective factors. *Domain Complexity* corresponds to the underlying system complexity based on the number of critical variables, possible system states, movement between states and subsystem interactions. *Control Complexity* relates to the factors that determine how system variables and states must be controlled in order for the system to work successfully. Control complexity is therefore dependent upon domain complexity to some extent, but also depends upon the control strategy adopted by the designers, as it is at this level automatic systems may be deployed to control some elements of the system in a closed-loop. It is therefore at this stage that what the user will be able to control or influence directly is decided. The third level of complexity is *Interaction Complexity*. Interaction complexity represents the demands placed upon the user of the system in terms of the information provided and the mechanisms provided to control the system. We may therefore view Domain and Control Complexity as objective measures of system complexity, and Interaction Complexity as the subjective complexity as seen by the user, as it relates to the ability of the user to perceive the operation of the system correctly and diagnose and correct faults appropriately. Interaction complexity is therefore also dependent on the cognitive and ergonomic demands imposed on the user by the human-system interface.

1.3 Subjective Interaction Complexity

Interaction complexity provides a conceptual interface between the system and the human operator. It is this interface which directly influences the subjective complexity of the system as viewed by the user. Hayek (1967) addresses this subjective view by defining phenomena as complex when it cannot be specified or predicted to any desired degree. In order to provide such an arbitrary degree of prediction it is necessary for the user to not only perceive and control the phenomena or system but also to understand the behaviour of the system. This in turn requires the user have the requisite knowledge about the system in order to predict and understand its behaviour.

Kieras and Polson (1985) suggest that the complexity of a device or system *as it appears to the user* depends upon the amount, content and structure of the knowledge that must be utilized by the user in order to operate the system successfully, and also the ease by which the user may learn these factors. They go on to define *Cognitive Complexity* as the complexity of a system from the point of view of the user. Knowles (1988) draws the distinction between the *quantity* and *type* of knowledge required to operate the system successfully, and terms this Cognitive Complexity Theory (CCT). Thus the effectiveness of control (and the subjective evaluation of complexity) depends both upon the user-system interface and the knowledge of the user. In designing the interface to a complex system it is therefore necessary to consider the knowledge of the anticipated user, and an expectation of how the user will behave in order to judge how complex the system will appear.

1.4 Issues in Designing for Interaction

Systems appear complex when the human users find it difficult to predict how such systems will behave in some circumstances. The human-computer interface is the system component that indicates both the state of the system and the effect of any controlling actions to the user. This human-computer interface is a conceptual dividing line that represents the intersection of the human and computer system boundaries. The human-computer interface is therefore central in showing what the system behaviour is (its *state*) by means of various representational techniques (e.g., diagrams, text, sound, and so on). It also provides the user with a means of entering *commands* to control or operate the system (e.g., by typing commands or using pull-down menus). Therefore we may view subjective complexity as arising from domain and control complexity, reflected by the interaction complexity at the user interface.

1.4.1 Interface and Interaction

At the human-computer interface information is exchanged in a way which relates to the *flow* of information (between user and computer) and the *representation* of that information (e.g., text, graphics, sound, speech). Until recently most human-computer interaction involved a turn-taking process in which text was entered and responses were displayed on a display screen. This form of interaction was termed a *dialogue*, as it was mostly a process of exchanging textual commands and responses. With the advent of graphical user interfaces and multimedia systems there has been a move away from characterising interactions as dialogue towards defining interaction *styles* (Preece et al., 1994). However, the use of interaction styles to characterise interaction is based primarily on commercial software which provides metaphor-based user interfaces to home and office computers. Such systems cater for a wide range of users (in terms of experience with computers and familiarity with other domains on which metaphors may be based, such as the desktop metaphor). These interfaces combine elements of representation and dialogue, often in an ad-hoc fashion.

Interaction and Representation are connected in that underlying human cognitive processes determine the preferred form of interaction and representation for a *task* or *set of tasks*. The concept of a task is used as an abstraction which maps user goals on to the means of achieving those goals. A task may be represented to the user in many different ways, and different representations of a task can force other styles of cognitive behaviour to be adopted by the user (e.g., to prompt a user to take some action, a warning light may be used, or a textual instruction, or an auditory cue). This is of particular importance in the operation of complex, real-time systems because unlike commercial software packages (which only respond to user input), systems such as aircraft, process control systems, etc. exhibit behaviour independently of user actions. It is therefore necessary to address the role of the user interface from first principles in order to evolve a theory of interaction that includes the requirements of complex, real-time systems.

1.4.2 Feedback

The provision of information about current system state and the state changes brought about by user interaction is termed *feedback*. Feedback relates to the provision of continuous monitoring of the result of an input to one system by another system, and is used in electronics and physiology to describe the interaction between systems (Figure 1.1). Feedback provides a closed loop of interaction that is used to maintain the two interacting systems in equilibrium. The concept of feedback between user and machine was extended from the feedback provided by mechanical devices that also served humans as tools (e.g., motor-vehicles). This notion of feedback relates to the continuous signalling of changes in system state to the user as the user interacts with the system. This involved analogue dials, noise etc. to show the dynamic behaviour of the system.

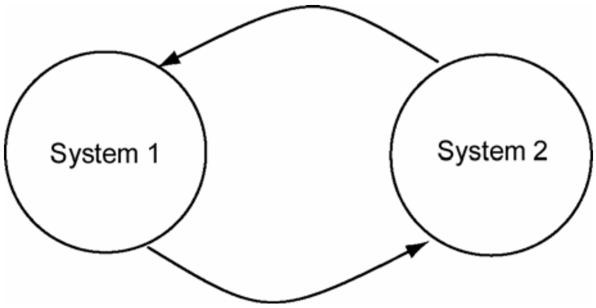


Figure 1.1 Feedback between two systems

Gradually various sorts of automation appeared to aid the operator of such systems. In aircraft this took the form of simple autopilots (Billings, 1991) and in the control of manufacturing processes such as the production of chemicals regulating devices were developed to allow a state chosen by the user to be maintained (e.g., in the very beginning of automation, governors on steam engines).

1.4.3 Providing Feedback Between User and Computer

Various models have been formulated to capture the roles adopted by human user and computer. These models attempt to capture the relationship between user and computer in terms of the information content and format which is transferred at the human-computer interface. Essentially the models describe the nature of the feedback between the system and the user.

Several models (Figures 1.2 – 1.5) can be used to illustrate the interaction roles adopted by human and computer based on models of supervisory control (Sheridan, 1984) and human interaction with expert systems and decision aids (Buck, 1989). Solid arrows represent *continuous* information transfer and dotted arrows *intermittent* information transfer.

In manual control (Figure 1.2) the operator interacts directly with the control systems without any computer support. This may take the form of a person driving a car, or a person operating a drill press.

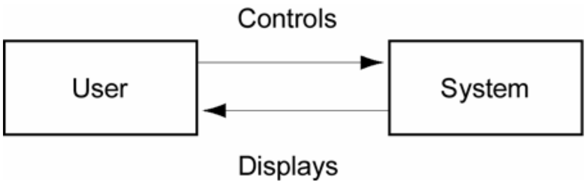


Figure 1.2 Manual Control

Advisory control (Figure 1.3) represents the control of a physical system directly by the operator but with an expert system providing on-line help. This is a classic view of expert system aiding in which the aiding system is not connected to the domain except via the operator, e.g., MYCIN (Shortliffe and Buchanen, 1984), PROSPECTOR (Reiter, 1984), GRADIENT (1989) etc.

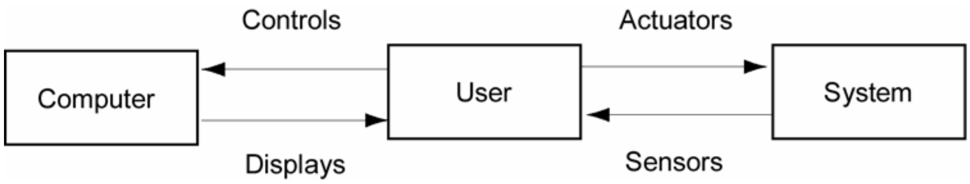


Figure 1.3 Advisory Control

The third model (Figure 1.4) represents the automatic control of a system around limits specified by the operator. In automatic control a specified set of parameters are maintained within specified limits without human intervention (e.g., Autopilot, automatic mixing of chemicals, etc.).

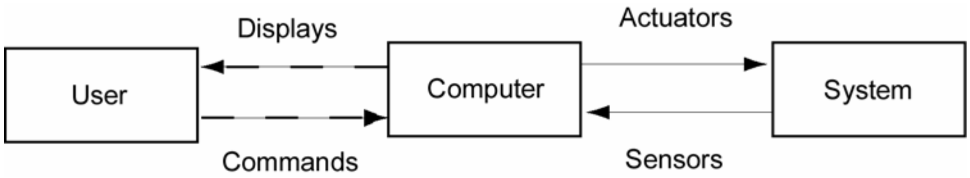


Figure 1.4 Automatic Control

The final model (Figure 1.5) represents supervisory control. Elements of the other three models are combined in a system that constantly displays information about its state, but only periodically interacts with the operator. Supervisory control systems include many types of system which are erroneously classified as automatic (e.g., medical life-support systems, process control tasks, agent based computer control systems).

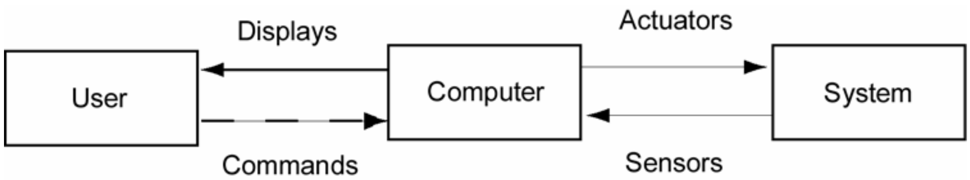


Figure1.5 Supervisory Control

These models are based on classical control theory using the concept of feedback to model interaction. The last two models are of primary interest as:

1. The user interacts with the physical system only via the computer;
2. The computer acts as an intermediary and conducts some form of dialogue (the continuous flow of data for feedback is absent) with the operator in order to implement control.

As automation is increased in the control of systems the number and interaction of feedback loops becomes very complex. In supervisory systems such as the tele-operation of remote vehicles the feedback loops may be illustrated as follows (Figure 1.6).

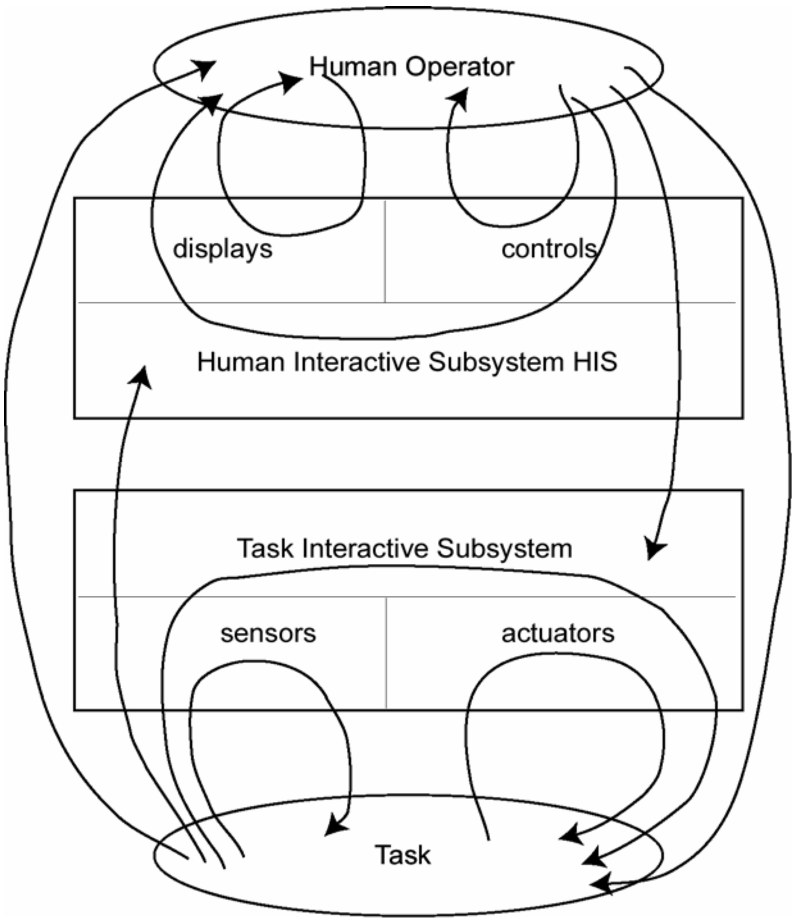


Figure 1.6 Multiplicity of control loops within a single tele-robotics system (Sheridan, 1984).

The important aspect of the change in level of automation (how directly the user interacts with a system) and the dynamics of interaction and information exchange is due to the user only experiencing the system through the user interface:

> "This means that the operator interacts with the process through the automatic control system and through the interface. The operator's understanding of the process therefore becomes contingent upon how it is represented at the interface." Hollnagel (1995).

The level of automation has a great impact on how a user interacts with a system and consequently the way the user understands the operation of the system. The way in which the information is represented to the user (the interface) and the way in which information is exchanged between user and computer (the interaction) are therefore bound up with how complex the system appears to the user.

1.4.4 Interaction Style

In addition to relating the meaning of fed-back information, embedded computer control changes the nature of the way feedback propagates through a system and hence consequently also changes the nature of the user's interaction with the system at the human-computer interface. In monitoring the multitude of feedback parameters it is necessary for the user to know the *meaning* and *relevance* of the information. This meaning and relevance must be understood in relation to the environment, the experience of the user and the relations between the user's perception of the environment and the environment itself. This information concerns the *relationship* between information rather than the indication of *outcomes* and is known as *cognitive feedback* (Balzer et al., 1989). The interaction between user and system is consequently also dependent on the perceived autonomy of the computer system, and on the user's expectation of what the system is capable of doing autonomously. The level of autonomy is linked to the meaning and relevance of indicated information as this is how the system indicates that the user should take some action.

Initially the metaphor that has been used for the design of artefacts has been that of a *tool* – something constructed by humans to facilitate the performance of certain tasks to reach certain goals. The understanding and usability of such tools depends upon the interpretation of the tool by the user (Winograd & Flores, 1986; Norman 1986). The way in which the properties of a tool are inferred and understood by the user is critical in deciding when the tool may be used. This is a reflexive process as what the tool can be used for affects how the user views the world in which the tool is used (i.e., how the tool user formulates goals). To be useful a tool must facilitate the goals of the user in a way that is readily apparent and easily achieved.

The interpretation of computers as tools depends on the appearance of the system to the user as represented at the human-computer interface. When computers are used as tools to perform tasks, the state of the computer and the domain in which the tool is being applied must be made clear to the user at all stages of task performance. The first true digital computers indicated the state of the computer and the problem space by showing the current instruction being performed and the address of that instruction in binary code. As computers-as-tools have become more sophisticated, the representation of what the tool is capable of and when it should be used have also become more complex. However, the main problem of the tool metaphor is that it gives little sense of the complexity involved in operating the tool. A tool in the traditional sense has no autonomous capability – it will not act unless manipulated by a user. Thus the tool model of interaction is the equivalent of manual control (Figures 1.2 and 1.7). Automated systems now available for plant control allow the system to prompt the user for information - to carry out a communicative dialogue, a form of interaction identified previously as being central to the supervisory control of systems.

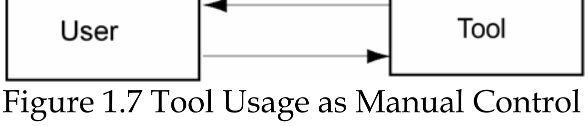


Figure 1.7 Tool Usage as Manual Control

The level of feedback complexity shown in Figure 1.6 raises several important questions in terms of the ability of a user to control such a system via an interface, and highlights problems with the computer as a tool metaphor. One concern is the ability of the system to perform some actions automatically, and how the system signals this to the user. The ability of the system to perform some actions autonomously is a critical factor in defining the role of the user. The use of automation also impacts the form of dialogue between interface and user; in commercial software the system will do almost nothing unless specifically instructed to do so by the user. However, the dialogue between user and system in supervisory control can take many forms. The levels of automation and consequently dialogue and roles may be summarized as follows (Rasmussen, 1990; Helander, 1988):

1. User does the whole task;
2. Computer helps in determining options;
3. Computer helps in determining options and suggests one which the user need not follow;
4. Computer selects an option and user may or may not follow it;
5. Computer selects option and implements if user approves;
6. Computer selects option and implements it unless user stops it;
7. Computer does task and tells user what it did;
8. Computer does task and tells user what it did, if asked;
9. Computer does task and tells user what it did, if it so decides;
10. Computer does whole job if it decides it should be done, and if so tells user what it did, if it so decides.

This is a more detailed analysis of automation and feedback that that previously presented (Section 1.4.3), and highlights both the level of automation and the changing role of the user. If we consider the communication between user and computer to be a *dialogue*, that is:

> "A temporally limited interaction between two or more relatively autonomous communicating entities through a sequence of exchanged messages" (Edwards and Mason, 1988),

then increasing automation changes the dialogue between human and computer by allowing the computer to take the initiative in *when* and *how* to communicate with the user; a dialogue style which reflects collaboration between two entities, rather than the use of a tool. This change from control to collaboration is a compromise between how much input the user can or will provide with the control needs of the system. Essentially it is a move towards a dialogue-based style of interaction, as might be conducted between two humans. Dialogue or Interaction *styles* are described by many authors (Shneiderman 1987; Sutcliffe 1988; Booth 1990; Dix et al, 1992) as a means of categorizing both the form of dialogue and the mechanisms by which the dialogue takes place simultaneously in conventional computer systems. Typically such taxonomies give styles such as *Menus*, *Command Entry, Question and Answer, Form Filling, Natural Language, Direct Manipulation*, etc (e.g., see Preece et al, 1994). However, there are several problems in using such definitions as a starting point for analyzing interfaces to complex, real-time systems:

1. The dialogue styles (in terms of user-system role) are almost always based on the user making all command decisions (controlling the dialogue structure);
2. Dialogue aspects are combined with representational factors;
3. Such categorisations are not based on cognitive theory but on a surface description of the interface (rather than interaction);
4. The styles must support interaction in a wide range of domains (rather than focus on specific applications);

Such dialogue style definitions are based on the behaviour we observe at the user interface, rather than being conceptually designed around the cognitive mechanisms of the user (Edmondson, 1993). Consequently, when considering the task demands placed on the users of complex, real-time systems, such styles do not allow different mechanisms to be provided for interacting with a system in different stages of behaviour. In moving from a supervisory role to one of problem diagnosis and rectification, such styles cannot provide the operator with any contextual cues about what to do next. In order to identify what features an interface should provide the user of complex, supervisory systems we must re-examine how interfaces are used to convey information about the systems under control.

Hutchins (1989) identifies several interaction metaphors within these styles:

1. Conversation Metaphor. This metaphor represents human-computer interaction as communication by conversation. Goals or intent are communicated with causal force to the domain of interest: i.e., commands are issued. However, feedback in this style of interaction is problematical, as the correct interpretation of causal intent is critical, as is the signaling of success and failure between the two dialogue partners.

2. Declaration Metaphor. Expressions at the interface appear as actions with causal force at the domain or world of interest. However, Hutchins notes that this metaphor, in which saying is doing, can only be supported if everything that can be said can be done.

3. Model-World Metaphor. The user takes action directly in the world of action which is itself the medium for the interface language. It is not possible to compose an expression which cannot be realized in the domain (e.g., menu-driven and direct manipulation interfaces).

4. Collaborative Manipulation Metaphor. The user may act directly on the domain or act on the domain via the computer. This combines elements of manual control with other styles involving computer mediation.

These interaction metaphors emphasise the move away from a model of the computer as a static tool to a metaphor in which the computer plays the part of an intelligent collaborator. This view of interaction divides the responsibility for recognising what the system can and cannot do, when and how to act, etc. between the user and the system. The implication is that the computer itself must exhibit some form of intelligence to do this. Storrs (1989) defines the terms *interaction* and *interface* in the following way:

1. An *interaction* is an exchange of information by which two agents modify the state of one-another;

2. This exchange of information takes place via a set of information channels which constitute the *interface* between the agents.

These definitions highlight the fact that the interaction takes place between two (at least partially) independent entities (*agents* in Stores' terms). The mechanisms for specifying actions (style of interaction) and interpreting displays (style of interface or representation provided) must be able to cater to the expectations of both agents in order for the interaction to be successful.

1.5 Interacting with Complex Systems

The interaction that a user has with a complex dynamic system is influenced by several factors. The first of these concerns the fact that a complex, dynamic system always contains some automation, and this changes the nature of the feedback between user and system. This results in a continuum of automation, with the system having the capability to dynamically alter the degree of automation (Section 1.4.4). The first important insight here is that the characteristics possessed by traditional, complex, dynamic systems are identical to those present in current software systems. Therefore by comparing the design of feedback in traditional systems with the design of human-computer interfaces for software it is possible to examine how feedback presents the system to the user. Feedback may consequently be viewed as providing a style of interaction that presents the computer as a dialogue partner, rather than as a tool. The structure of this dialogue is dynamic and because the system is driven by external factors - the computer (as well as the user) can instigate a dialogue. This may happen because of a change in the system being detected by the computer, and the computer requesting the user to consequently take action. Comparing interaction models with automation models, the conversation metaphor presents the computer as a dialogue partner as capable as another human user.

The agent model of the computer can be seen to parallel the model of supervisory control (Figure 1.8) as it is capable of providing elements of advisory control and automatic control (Figures 1.3 – 1.4) but with an asymmetric constraint concerning when dialogue can be initiated (i.e., only under certain conditions known by the user). Consequently the spectrum of feedback viewed as interaction becomes *Tool Usage*, through *Agent Mediation* to *Communication*.

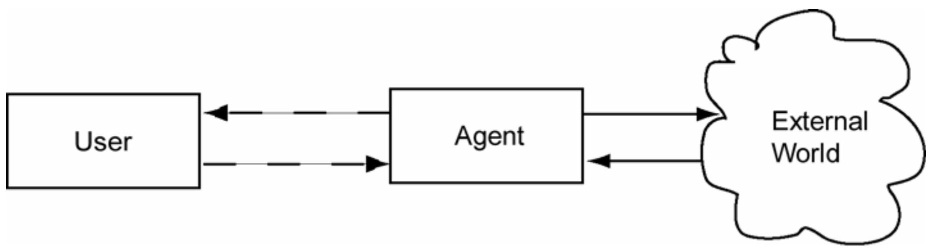


Figure 1.8 Agent Interaction as Supervisory Control

Another factor concerns the form that is used for the interaction – the interface style. The way in which the interface is presented (i.e., the interface style) will depend on the dialogue being undertaken, the nature of the feedback and the cognitive demands placed upon the user. The interface style may also change dynamically, based on the dialogue that takes place. The second insight is that the way in which feedback and interaction are managed contribute to the complexity of the system. The third insight is that complexity related to interaction is *subjective to the user*. Both the interaction and interface must consequently be carefully managed, as the implementation of these has been shown to contribute to the user-perceived complexity of the overall system. The management of interface, interaction, automation and complexity are bound together and must be designed and implemented in a cohesive manner to minimise possible user problems.

Implicit in both assigning automation to the computer and in allowing the computer to control the interaction and the interface is the concept that the software system itself is exhibiting intelligence in the way that it interacts with the user. The way in which this dialogue is managed consequently requires some form of intelligence on the part of the computer.

1.6 Summary and Overview of Thesis

This chapter has outlined the subject area of this thesis - computer systems applied to the control of complex dynamic systems, and the problems that users have dealing with the resulting subjective system complexity. Examining the design of these systems from the twin perspectives of the design of *feedback* and the design of human-computer *interaction*, it has been possible to compare these seemingly disparate approaches, and identify commonalties between them. These commonalties may be viewed as methods to intelligently manage the interaction/feedback between user and computer/system. This management changes the computer component of the system (the part that handles interaction with a user) from being a *tool* to being a *partner* in a control *dialogue*, with the computer mediating between the user and the environment which is being controlled. This mediation encompasses the models of automation which are used to place the user in a supervisory control role. In supervisory control the computer may be viewed as acting as a control agent on behalf of the user. Thus *feedback* becomes *dialogue*, and *automation* becomes *agency*.

The structure of this thesis becomes the following:

Chapter 2 presents a survey of methods that can be used to provide such an intelligent dialogue partner, examines the ways in which such an intelligent dialogue partner acts as an agent for the user, and evaluates them in regard to interacting with complex, real-time systems.

Chapter 3 analyses the findings of Chapter 2 and develops a new paradigm for the design of intelligent and adaptive interfaces using the notion of agency. The design paradigm is based on the notion of actively interpreting the context of use and is termed *contextualization*. The model of contextualization is decomposed into a number of components which are intended to structure the design of an interface in a way which promotes understanding and consequently improve user interaction with a system.

Chapter 4 examines the components of contextualization identified in Chapter 3 and evaluates them in an empirical setting.

Finally Chapter 5 analyses and discusses the findings of Chapter 4 and suggests directions for future work.

Chapter 2

Aiding Interaction

2.0 Providing an Intelligent Dialogue Partner

The use of computer automation in current systems allows the computer to behave autonomously and to have some control over the structure and content of the human-computer interaction. Storrs (1989) views this as the interface adopting the role of an *agent* - a relationship approaching that between two humans, rather than a human and an inanimate tool. In this agent-like operation the human user may signal the computer to perform some task and then report back on progress. Feedback no longer relates to individual or fused sets of system variables but to task relevant factors such as success, failure, deviations from anticipated results, etc. The form of information exchange at the human-computer interface begins to appear not as a simple indication of system state, but rather as a dialogue between two (or more) intelligent entities (human and computer) *communicating* in order to solve a problem *co-operatively*. This communication relates to both the content of the information (i.e., information about system states), and also the meaning (partly reliant on the representation) of this information in relation to the state of the system and the goals of the user.

The metaphor of user and computer engaged in communicating with one-another (carrying out a dialogue about the task in hand) is summarised by Hutchins (1989) (see Figure 2.1). Here the interface intermediary within the interface acts as an intelligent entity, mapping user goals onto the domain of interest (or world of action) via shared symbolic representations. Such a model requires some corresponding intelligence within the system to enable the interface to adapt to the requirements of the user and maintain a dialogue that gives appropriate feedback. This is usually referred to as an *adaptive* or *intelligent* user interface.

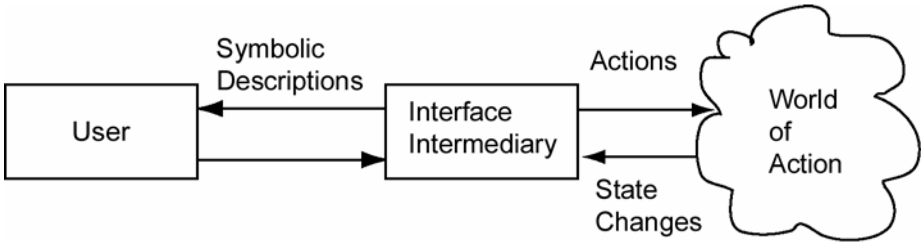


Figure 2.1 The role of the computer as an Interface Intermediary

It would therefore seem that adaptive-intelligent user interface techniques should provide the capability to manage the elements of feedback/dialogue and interaction/agency. This chapter surveys the literature on these technologies, and evaluates their utility as a means for addressing system complexity.

2.1 Intelligent and Adaptive User Interfaces

As computing technology became less specialised and used by people other than the constructors of the computers, researchers began to appreciate that machines of such complexity should be able to respond to the needs of the intended users. This premise extends many traditional design methodologies that recognise the target user group is not homogeneous and provide mechanisms for customising the device or product (Murray 1991). In addition to making the system easier to use, this provides a number of benefits which also make intelligent and adaptive interfaces attractive from a commercial viewpoint. These benefits increase the number of potential users by (Browne, Totterdell and Norman, 1990):

1. Allowing usage with varying amounts of training;
2. Increasing the life span of the system (by allowing it to adapt to external factors);
3. Reducing operational learning;

4. Increasing user satisfaction;
5. Catering to the preferences of diverse user populations (such as both shop floor and management).

Generally this adaptation is seen to increase the learnability and usability of a system (Tyler and Treu 1989), and to increase the overall performance of the human-machine system in control applications. This user-centred approach has other benefits, particularly allowing the re-use and integration of existing software and extending the life of existing systems (Edmonds et al., 1992).

Edmonds (1981) was one of the first to highlight the need to adapt the interface of the computer to the requirements of the users, pointing out that such adaptation can be done by a computer specialist, a trained user or, indeed, any user. However, this adaptation is considered as an off-line process to be done when the system is not actually being used (e.g., as part of an iterative design process). Innocent (1982) identified the need to allow the interface to be self-adapting, possibly as a means of separating interface and application. Adaptation as *customization* implies that the user has enough knowledge to select the best options from those available, and this may not be the case with novice or infrequent users. *Self-adaptation* allows the system itself to decide if and when to adapt, and what form the adaptation might take. Benyon and Murray (1988) also separate the type of adaptation from what is being adapted to - adaptation can be seen as being instigated by some feature of the user in order to improve the effectiveness of the interaction. At the simplest level the user may decide to customize the interface to their own preferences by changing screen layout, etc. At the other extreme the system or interface itself may decide to adapt to some behaviour of the user or system automatically without any instigation from the user (e.g. Alty, 1984), perhaps by changing the level of detail provided as help based on the experience level of the user. It is this "self-adaptation" which is referred to as an *adaptive interface*. Chignell and

Hancock (1988) see the concept of an *intelligent interface* as evolving from this recognition that there is a gap between the needs (and abilities) of the user and the capabilities of the machine which must in some way be *bridged* (Norman 1986, Card 1988). This can result in a range of features, such as automation of routine tasks, easy access to tools, provision of on-line assistance and documentation and may also involve tailoring user-interface style to a particular user or class of users (Rissland 1984). In domains in which safety and performance are of particular importance, factors such as speed of response, correctness of response (number of errors) and other decision-making elements play a greater part. Workload and situational awareness become important components in real-time systems such as aviation (Reed, 1990). In addition to adapting to capabilities, Sukaviriya (1993) also highlights the need for systems to adapt to the preferences of the users (regardless of whether this provides any quantitative performance gain). One of the possible problems of adaptive user interfaces is that they can appear inconsistent to the users (Hockley, 1986). To address this Hancock and Chignell (1989) include *naturalness of interaction* to the list of features of such an interface. Observing this rationale from another point of view we may say that an intelligent interface is required when user understanding is incomplete (i.e., when a system appears complex to the user), or user performance is poor, or requires large amounts of training (or both). Failure to understand the task model presented at the interface may also indicate the need for an adaptive intelligent interface (or a re-design of the system!).

The terms "Adaptive Interface" and "Intelligent Interface" are used interchangeably in the literature, as for an interface to adapt, it must in some way be intelligent (and conversely the intelligence in an interface is demonstrated by its ability to adapt; see Schneider-Hufschmidt et al., 1993). From this point the umbrella term Intelligent Adaptive User Interface (IAUI) will be used to include both intelligent and adaptive user interfaces. The use of a particular term usually depends on the domain in which the interface is being implemented. Taxonomies of adaptive and intelligent user interfaces also show this partitioning (see Section 2.4).

2.2 Factors Triggering Adaptation

Essentially an IAUI allows a system to be used by a number of users, regardless of individual differences. The objective is to generally improve combined human-system performance, aiding users who experience difficulty and allowing users to make the most of their domain knowledge (Egan, 1988). It follows that the characteristics of users that lead to individual differences are those factors which trigger adaptation in adaptive intelligent interfaces, whether or not this expertise is related to preferences, skills, etc. (Desmarais and Liu 1993). Browne, Totterdell and Norman (1990), Edmonds (1987) and Lee (1993) provide a series of elements influencing individual differences including:

a) Individual differences;
b) Psycho-motor skills;
c) Procedural capability;
d) Learning ability;
e) Understanding;
f) Expectation/motives;
g) Cognitive strategies;
h) Cognitive abilities (memory and attention);
i) Preferences;
j) Temporal changes;
k) Situation specificity;
l) Growth and transition of user skills;
m) Multitasking;
n) Error-propensity of users.

Not all of these elements are adapted to in every IAUI, but the elements that are adapted to depend upon the domain in which the overall system is operating. Polson et al., (1989) view the human characteristics which should be adapted to as those which enable the maximum amount of information to be obtained from multiple sources, maximizing the capabilities of the user given the behaviour of the system. The interface may be thought of as mediating these user capabilities with the characteristics of the domain, and in Weir's (1991) nomenclature (see Section 1.2) an IAUI may be thought of as managing interaction complexity by mapping user abilities to domain and control complexity.

2.3 Mechanisms for Adaptation

An IAUI requires both representation and process in the classic cognitive sense in order to operate. The representation is in the form of data or knowledge that enables the system to decide when the process of adaptation should take place and what form the adaptation should take (Carter, 1990). The process encompasses the models and implementation that represent and animate the knowledge.

2.3.1 Knowledge Required by an Adaptive Intelligent Interface

Card (1989) highlights Young's Third Law of expert systems which says that every expert system must contain knowledge about two domains: (a) the area of expertise and (b) how to communicate with the user. Essentially an IAUI is an expert system for communicating with the user in a particular domain, requiring the system to know when and how to adapt (Waterman 1986). Norico and Stanley (1989) provide an overview of the early literature and summarize Rissland's (1984) description of the types of knowledge which are necessary for an interface to adapt intelligently as:

1. Knowledge of the user (expertise, capability, preferences, etc.);
2. Knowledge of the interaction (modalities, dialogue styles, etc.);
3. Knowledge of the task and domain;
4. Knowledge of the overall system.

Thus in order to adapt, the IAUI must contain knowledge bases to both detect and trigger adaptation, and also knowledge of the appropriate form of adaptation in a given situation (e.g., how the interface or dialogue or system behaviour changes). For example, in CHORIS (a system for emergency crisis management, Tyler et al. 1991) the knowledge bases include *Vocabulary* used by the presentation manager, a *Domain Model*, *Command/Task Model* and *User Model*.

2.3.2 Components of an Adaptive Intelligent Interface

In order to adapt to humans, a system needs to know what the user wants to accomplish – it needs a model of the tasks that user may perform and possibly also of the domain in which the tasks will be carried out (Norico and Stanley, 1989). This enables the system to infer user plans and goals from their sequence of actions. The system also needs to know the range of dialogues that can be used and the characteristics of each dialogue in terms of structure and modalities (interface styles), so that it may change the dialogue in order to appropriately adapt. Finally the system needs some model of the user to assess or predict user characteristics and trigger adaptation. Chignell et al., (1989) view an intelligent interface as being comprised of 3 main components;

1. A Task model;
2. A User model;
3. A Translator (between user intentions and machine actions - a dialogue manager).

Browne, Norman and Adhami (1990) also specify a user model (to model the variability on which adaptation takes place), a dialogue model (to permit changes in the user interface) and a task model (as variability is often dependent on the context of interaction). Essentially the dialogue model provides the ability to translate between user intentions and machine actions (in Chignell's terms). The user model may capture expertise in a variety of forms, depending on the tasks being performed (Desmarais and Liu, 1993). A more finely-grained model is given by Rouse (1991), comprising a user model, error monitor, adaptive aiding and interface manager. The adaptive aiding component provides the mechanisms for changing the dialogue. The user model includes intent, resources, performance, plans and goals. The world, system and user states are monitored in order to detect factors that trigger adaptation according to these models. User models generally involve cognitive models of the user that can address user performance. The capability of the user model in real-time system such as supervisory control is one of both evaluating and predicting user performance (Eberts and Eberts, 1989). This is necessary as the dynamic capabilities of the user need to be predicted in order to perform dynamic task allocation, taking into account resource limitations (Rouse 1991).

2.3.3 Dialogue Management

The effect of an IAUI is to manage the way in which human and computer communicate; a process termed dialogue management (see definitions of dialogue styles, and the concept of dialogue instigation in Section 1.4.4). The way in which adaptation is visible is by a change in the human-computer dialogue. This change may be in the form the dialogue takes, or in the content of the dialogue, or both. In terms of the form the dialogue takes, adaptation can be broadly defined to take place when the system responds autonomously in some way (i.e., it adapts autonomously). Thus the classification of systems according to the way feedback takes place does not continue into IAUIs because the feedback can be changed as part of the intelligent adaptation. This might be evident in the way dialogue is presented; e.g., the computer suggests something, rather than waiting for the user to ask before responding (Whalster, 1989). Alternatively the computer may instigate a change in the way information is presented, by presenting graphical information, or using sound to get the user's attention. Initially the intelligent interface was dominated by the natural-language metaphor, because interaction was centered on the use of a Teletype for interaction (Miller et al.,1991). However, adaptive and intelligent interfaces now perform complex management of multiple modalities extending into graphical user interfaces and those making use of speech, sound, gestures, touch, etc. (see Elkerton and Williges 1989 for a list of possible modalities). In process control the PROMISE system (Alty and Bergan, 1995) allows multiple modalities to be used for displaying system state. CUBRICON (Neal and Shapiro, 1991) allows many modalities of communication, controlled as an intelligent and adaptive dialogue. One possible way to view such interfaces is as intelligent multi-modal interface control systems, systems that share many of the characteristics of IAUIs as their *raison d'être* is to adapt the presentation of information to the user.

2.4 System Classifications of IAUIs

Whilst IAUIs can be seen to have a similar set of components and range of factors concerning how and when they adapt, in practice these systems are often classified according to the domains in which they are implemented. At a high level of abstraction Rouse (1991) subsumes the area of intelligent and adaptive interfaces into the global concept of *aiding*. Aiding is defined as functionality that is added to a baseline design concept specifically for the purpose of enhancing human decision-making, problem solving and performance in general. Thus aiding itself may be classified according to the domain in which it is implemented. Starting with a more historical analysis, Woods et al. (1991) begin a classification of IAUIs by noting that many initial expert systems produced the impetus to improve dialogue. Their classification begins with a survey of human-human advisory interaction. This is extended as a basis for examining human-intelligent computer interaction and advisory interactions. Woods' survey in the most general sense seeks to address how it is possible to integrate computer power and human practitioners. Woods and Rouse's surveys serve as defining the domains in which IAUIs are likely to be found, described in the specific terms of each domain. Totterdell and Rautenbach (1990) provide a taxonomy of adaptive systems classified by level of adaptivity and relating these levels to the associated biological systems. They note that such classifications are based on how much control the system itself has in instigating change; from none (a designed system, incapable of further change) to self-modifying. Their complete taxonomy is given in Table 2.1.

System Type	Characteristic
Designed	Static
Adaptable/tailorable	Deferred selection
Adaptive	Apparent learning
Self-regulating	Learning with evaluation by trial and error
Self-mediating	Planning with internal evaluation
Self-modifying	Generalization and meta-knowledge

Table 2.1. Totterdell and Rautenbach's taxonomy of adaptive systems

More detailed taxonomies seek to partition the global concept of adaptive and intelligent interfaces to identify common components and functionalities between diverse domains of implementation.

Malinowski et al., (1992) and Dieterich et al., (1993) approach the classification of adaptive interfaces from a more user-centric perspective. Their analysis is based on the sequence of events that are considered when adapting the interface. The process of adapting the interface is divided into 4 stages.

The first stage is based on whether the user or the system instigates the adaptation. The second stage concerns who proposes the form that adaptation should take. The third stage is the decision of what proposal to accept, and the fourth is the execution of the proposal. These events can be instigated by the user (U) or the system (S), or both, or neither. This gives 4 by 2 by 2 matrix, of which 6 combinations are identified as of particular interest, User Initiated Self Adaptation, Self Adaptation, Computer Aided Adaptation, User-Controlled Self Adaptation, Adaptation and System-Initiated Adaptation (Figure 2.2).

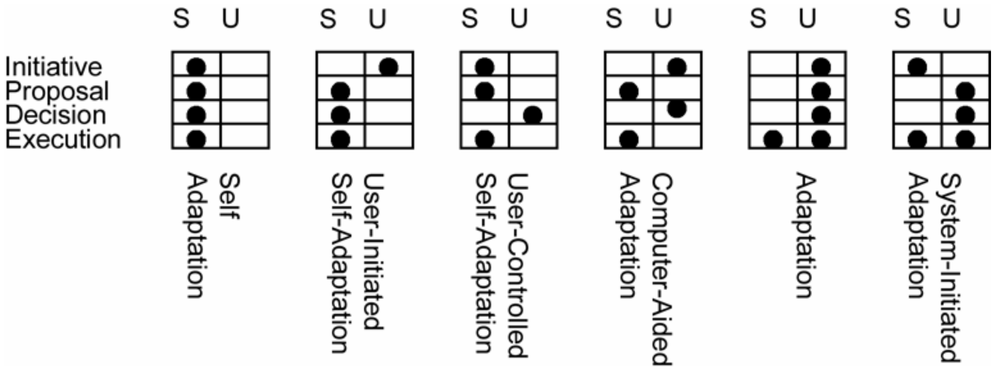


Figure 2.2 Models of Adaptation

Malinowski et al. use this set of adaptivity definitions to classify systems by identifying which of the six classes a particular adaptive system supports in a way which is very similar to defining the level of automation (Section 1.4.4). The possible space of system behaviour encompasses systems exhibiting more than one classification (e.g., in *adaptation* both user and system can execute adaptation). It is therefore possible for a system to be classified as several categories (or in the case of human-human, all categories). The way in which the system behaves in adapting defines the level of automation exhibited by the system, but not the triggering factors, the levels of possible adaptation (cf. Totterdell and Rautenbach 1990), or the form the adaptation takes. Obviously, these are important factors which contribute the behaviour of an adaptive system.

Balint (1995) approaches the classification of adaptive human-computer interfaces from the perspective of the domain in which the system is deployed (e.g., co-operation-based user teaching) and the type of adaptivity shown by the interface. The type of adaptivity is divided into 4 categories; computer-based human entertainment, computer-aided human creative activities, computerised office administration, computer integrated systems.

Neither Malinowski et al.'s nor Balint's taxonomies examine adaptivity from the viewpoint of the user in terms of identifying interfaces by *when* the user might see the interface adapt, and the *form* which this adaptation may take. A taxonomy should therefore encompass the dialogue factors of Malinowski et al. as part of the form, and Balint's concept of adaptivity type in terms of triggering factors, but also include issues that can effect these factors. Adaptive interfaces implicitly adapt to change the factors and facets of their adaptation (see Totterdell and Rautenbach, 1990) based on whether they can learn (or be taught). Thus the classification of IAUIs may be made according to the following 3 factors:

a) What is adapted to;
b) How adaptation takes place;
c) Whether or not the factors in (a) or (b) can be modified by the system.

(a) Concerns whether the system adapts to the user or to both the user and the system – distinguishing issues of time-specific interaction, etc. (b) concerns the factors from user or system which are taken into account, and the process through which user-computer dialogue allows adaptation. This addresses the ways in which non-autonomous adaptation implicitly involves user-choice factors. (c) addresses the system's ability to self-adapt, and learn from the factors in (a), changing the factors in (b). Adaptation to user and system also includes factors concerning the situation in which the interaction takes place as in context-sensitive help systems. Adaptation to the user takes the form of changes in dialogue and interface, and includes a range of applications such as tutoring systems, critiquing systems, information filtering and on-line help.

2.4.1 Adaptation to User and System

Adaptation can initially be divided into adaptation based on user characteristics, and adaptation based on user and system characteristics (adaptation based on system characteristics alone is automatic control). In the case of real-time systems, *Adaptive Aiding* is aiding which changes as task demands change, providing adaptive user support (Opperman, 1994). *Intelligent Decision Support Systems* (Hollnagel 1987) are IAUIs applied to complex dynamic environments such as process control, also known as *Intelligent Assistant Systems* (Boy, 1991). Thurman and Mitchell (1995) call these systems *Multi-System Management aids* when used in hybrid environments such as satellite control. These real-time systems introduce the temporal nature of not just what and how to display information but when the information should be displayed. In this way the IAUI manages both the quality and quantity of information; e.g., in order to reduce operator overload (or offering extra information and support). IAUIs in real-time systems and supervisory control are often concerned with monitoring user performance in terms of errors, stress and situational awareness, and using these factors to change the presentation of information to the user, including aspects of information filtering for data overload. In addition these systems provide advice-giving capabilities for support in problem-solving, reasoning and decision-making. These systems have been deployed in aerospace applications (Higgins et al., 1989), navigation assistance in fighter aircraft (Amalberti and Deblon, 1992) and manufacturing simulation (Trumbly, 1994).

2.4.1.1 Context-Sensitive Help

Context sensitive help uses the state of a system to infer what information is relevant to the user and to consequently constrain the information supplied. Mittal and Moore (1995) present help systems with follow-up contextual levels of detail. Danielsen et al., (1993) compare the effectiveness of help with context sensitive help, concluding that context sensitive help is faster, but may reduce comprehension. Hiyoshi et al., (1993) extend this concept to include context sensitive product user manuals which provide operating assistance tailored to how the product is used.

2.4.2 Adaptation to User

User Support capabilities (Lee, 1993) also encompass intelligent and adaptive user interfaces. Many of the first applications of adaptive and intelligent interfaces may be classified as "user support" as they were developed to enhance general user interaction with a complex system. Several systems were built for supporting particular applications such as electronic mail (Hockley, 1986; Mallen 1996), statistical tools and spreadsheets (Maskery 1984, De Rosis et al., 1993, Thomas 1993, Opperman 1994), Directory and Database Searching (Croft, 1984; Greenberg and Witten, 1985, Catarci, et al.; 1993; Brazier and Ruttkay, 1993) and Design Tools (Roth et al., 1994). Other systems were designed to provide general support by automating routine tasks. For example, Eager (Cypher 1991) is a programming-by-example system for a hypercard environment. It monitors the user's activities and when it detects an iterative pattern, it writes a program to complete the iteration (i.e., it recognizes and automates repetitive tasks).

2.4.2.1 Dialogue and Interface Preferences

Other user support is provided by systems that use user preferences to adapt the display of information. Some of these systems were aimed at the disabled; Pal (Pickering et al., 1984) is a communication aid for the disabled which adapts by predicting the stems of words being typed by the user, reducing the length of keying sequences. Fels and Hinton (1995) implemented a system called Glove-Talk II that translates hand gestures to speech through an adaptive interface. Savidis and Stephanidis (1995) designed a system intended for blind users that provides an adaptive method for converting visual to non-visual dialogues. More generally there have been other systems implemented for Window Management (Bellik, and Teil, 1993; Funke et al., 1993), Prompting

Style (Malinowski et al., 1993) and to integrate desktop applications and network services in a task-based fashion (Wood, Dey and Abowd, 1997). These systems either monitor user behaviour and adapt on a frequency basis, or identify user preferences and use this to adapt. For example, Adaptive Indexing (Furnas, 1985) monitors index terms and changes associated connections, whereas Lokuge and Ishizaki (1995) adapt the visualisation of complex information spaces based on particular user preferences.

2.4.2.2 Intelligent Tutoring Systems

Intelligent Tutoring Systems try to impart knowledge to the user in the areas of computer-based learning and training. Kedar et al., (1993) start with the premise that the best way to learn about something is to ask an expert, and thus these kinds of system often adopt the role of an expert tutor. Other systems adapt to the level of expertise of the user. Monitor (Benyon, 1984) selects dialogue scripts based on information about the user collected in a user model in the domain of computer-assisted learning. AIM (Fowler et al., 1987) adapts based on expertise and cognitive style (learning strategies). Novices seem to prefer system-guided dialogues whereas more experienced users are happy with a range of dialogue styles. Other systems have been implemented in Language tutoring (Schwind, 1990). Some systems take a goal-orientated approach and tailor advice accordingly (almost like a learning critic). For example, EdCoach (Desmarais et al., 1993) is a system that infers user goals and tailors advice accordingly. More ambitiously Crews (1995) attempts to provide a more generic Intelligent Learning Environment which provides general adaptivity to a range of domains.

2.4.2.3 Critiquing Systems

Critiquing Systems provide a system-instigated suggestion in areas of reasoning, decision-making and problem-solving. In Decision Making for example, Kossakowski (1989) compared adaptive (advice giving) vs. flexible (presents options) adaptation. The adaptive interface resulted in shorter learning time, decreased use of help and increased the use of more effective strategies. There has been some research on advice-giving systems that criticise the user in lieu of giving advice (e.g., Silverman 1992; Fischer et al, 1991; Stevens and Edwards, 1996). These systems overlap with Help and Advisory systems, but do not generally take the form of adaptive or intelligent user interfaces, instead they present a fixed knowledge-base of advice.

2.4.2.4 Information Filtering

Systems providing information filtering generally support the user in searching and browsing information, particularly in hypertext documents. Mathe and Chen's (1996) system on Hypertext Document Indexing suggest this gives better retrieval salience, but this is not statistically verified. Brusilovsky (1996) addresses adaptive hypermedia as a sub-class of IAUIs and lists 29 systems that personalise hypermedia in various ways. This includes filtering, and assisting navigation based on user models.

2.4.2.5 On-Line Help

On-Line help systems provide information intended to assist users in operating a complex system. In the Adaptable Help Manual (Mason and Thomas, 1984) the adaptive part of the interface models the user by quantifying the 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. In the Help System (Brooks and Thorburn, 1988) the user could trigger adaptation by pressing a button. This changed the detail of help provided. Users in the control group (no adaptation) requested help more often. It is not clear how effective such help systems are. Neerincx and de Greef's work (1993) indicates that on-line help worsens task performance and learning of novices substantially. More generally help systems seek to provide frameworks for deploying help. Sukaviriya (1993) uses the UIDE (user interface design environment) to provide a structure for adaptive help systems. Other work seeks to automate the generation of Help using formal methods that can be used to provide adaptive interfaces (Thimbleby and Addison, 1996).

2.5 Evaluating Adaptive Intelligent user interfaces

Regardless of the domain of application, mechanisms, factors triggering adaptation and so on, there is little comprehensive evaluation of the *effectiveness* of adaptive interfaces. "Dialogue" (Maskery, 1984) provided an adaptive interface to a package of statistical tool. The interface had 3 levels of dialogue, 1) forced choice system led, 2) free choice system led and 3) free choice user led. Users experienced difficulty when they were transferred to the user led interface. Although this system is often cited as the reason for providing adaptive assistance, adaptation could not take place within a session. "Monitor" (Benyon 1984) selects dialogue scripts based on information about the user collected in a user model in the domain of computer-assisted learning. There were inconclusive results on performance. Hockley's (1986) front-end for electronic mail detected 40% of user difficulties at the expense of a false positive rate of 35%. In addition, subjective user comments indicated that there were some problems with the inconsistency generated by the adaptation. Copeland (1992) provided a different dialogue style for each level of expertise. Adaptation was to level of user skill and task complexity. There were only 4 subjects and inconclusive results.

On the positive side Kossakowski (1989) compared adaptive (advice giving) versus flexible (presents options) adaptation. The adaptive interface resulted in shorter learning time, decreased use of help and increased the use of more effective strategies. In Manufacturing Simulation (Trumbly, 1994) the system adapts to skill level and improves user performance in both task learning and interface learning.

In assistance with spreadsheets (Thomas, 1993; Opperman, 1994) the "Flexel" system provides adaptation and presents suggestions for adaptation based on the user's interaction style (using menus and key shortcuts). In trials, a "critique" button allowed the users to instigate suggested adaptation. In subjective evaluation users liked the system. The Adaptive Toolbar (Debevc et al., 1996; Debevc, 1993) provided a menu that adds items based on frequency and probability of use. There was a significant ($p<0.01$) decrease in time taken for both novices and experts to complete the given tasks. CUBRICON (Funke et al., 1993) implemented an automated window managing system. Although only 2 subjects were used in subjective expert evaluations, both subjects gave positive evaluations.

From the surveyed literature, it is clear that there has been little comprehensive evaluation of IAUIs, and the evaluations that have taken place do not convincingly demonstrate the utility of IAUI as an interface mechanism that improves performance. The important aspects underlying the use of IAUIs stem from the desire to help manage user interaction with the computer through changing dialogues. However, it is by no means certain that such an improvement takes place. The central concept of an IAUI is to mediate between the user and the computer; to help the user map their goals into tasks that the computer can perform. One factor in this mapping is the management of human-computer dialogue by the computer. In the trials of IAUIs which were inconclusive or showed degradations in performance, the common factor is the unpredictability of the dialogue itself. When systems adapt without providing the user with an indication of why they are adapting, the user understandably becomes confused. This seems to be particularly true when dialogue and interface preferences are changed dynamically (e.g., Hockley, 1986; Copeland, 1992). In situations where the dialogue is more consistent, or the adaptation is clearly signaled to the user, performance improves (e.g., Thomas, 1993; Opperman, 1994). This unpredictability of dialogue instigation is reflected by the lack of a comprehensive taxonomy of IAUIs discussed in section 2.4. The 3-factor model developed in section 2.4 explicates this by explicitly including both the triggering factors of adaptation and the process by which dialogue is consequently adapted.

Although the effectiveness of IAUIs has not been conclusively proved (indeed, perhaps *because* of this), they serve as a precursor to an interface metaphor that is currently a topic of vigorous research – that of *agent based interfaces*. Although the two fields are rarely connected in terms of theory and implementation, they share many of the rationales and implementational structures. In fact, as the following discussion will show, effectively the field of Interface Agents has subsumed that of intelligent and adaptive interfaces. For this new paradigm to be successful, however, it must encompass the problems encountered in the design and implementation of IAUIs and provide a more robust framework for the functionality and implementation of such interfaces. This is particularly true because the agent-interface paradigm extends the richness of elements such as dialogue control and user and task modelling to attempt (in some cases) to provide almost human-like, collaborative interfaces.

2.6 Adaptive Intelligent User Interfaces as Agents

In many instances in the description of dialogue styles, and the behaviour of intelligent and adaptive interfaces, comparisons have been made between the communication of user and computer as "communicating agents". For example:

> "Human-computer interaction with a system having a developed user discourse machine is less like the use of a tool by a human than it is like the conversation between intelligent agents." Card (1989, p33).

> "[An intelligent interface is] an intelligent entity mediating between two or more interacting agents who possess an incomplete understanding of each other's knowledge and/or understanding." Chignell and Hancock (1989)

> "[An interface is] a device or representation that allows two agents to perform a task cooperatively." Chignell, Hancock and Loewenthal (1989, p.3).

Revisiting earlier examinations of interactivity (Section 1.4.4), Storrs' (1989) description closely links agents and interaction. Although the term "agent" has been applied to adaptive and intelligent interface technology, the embodiment of such interfaces as agents is a relatively recent event. Bird and Kasper (1993) identify the change in terminology as indicating a change in paradigm. The move from IAUI to agent-based support is seen as a move towards active, intelligent, decision support. Bird's work is located in problem-solving and decision-making, and his grounding is one of moving towards a partnership paradigm (from a prosthesis/re-design model). Maes' (1994) description of agent functionality essentially encompasses those functions realised by IAUI - an agent is seen as an interface intermediary that can reduce work and information overload by filtering, retrieving information, managing mail and meeting scheduling, selecting books and movies based on personal preferences, etc. The use of agents has also been proffered as a means of managing complexity in control systems (e.g., Zachary et al, 1995). Chin (1991) discusses whether intelligent interfaces should be structured as agents, or whether it is better to think of them as tools that intelligently organise direct manipulation options available for the user. There remains considerable debate over the utility of each approach (e.g. see Shneiderman and Maes, 1997**).** Proponents of the interface-as-tool point out that the agent view requires the interface to have a well defined dialogue model. Interface-as-agent proponents argue that the dialogue metaphor is a natural one, and when taken in the generalised case, dialogue includes "direct manipulation". In addition Chin argues that intelligent interfaces need to behave as agents for at least some of the time, as the interface may need to take the initiative to correct the user, volunteer information or suggest alternative courses of action. This is because a consultant system will generally have greater knowledge in their field of expertise than their users. From this discussion, it is clear that the interface-as-agent model encompasses IAUIs. The agent-model emphasises the issues of autonomy and to also highlights the capability of the agent to take the initiative in interacting with a user.

Jennings, Sycara and Wooldridge (1998) also emphasise that an agent interface can take the initiative and co-operate with the user in order to achieve a task. This emphasises the dialogue control aspect of IAUIs. Indeed the central characteristic of agent-based interfaces is the ability to instigate dialogue – essential for systems that break the computer-as-tool metaphor and allow multiple kinds of feedback. Thus from this perspective, agent based interfaces and intelligent or adaptive interfaces are one and the same. The agent model emphasises the presence of an adaptive component within an overall system. It is therefore possible to view a complete system as adapting, even if it is a single component of the system (the agent) which provides this capability.

2.6.1 The Agent Perspective

Interface Agents are a specialised form of the more general concept of *software agent.* Although there is no real consensus over what a software agent *is* (Franklin and Graesser, 1996; Bird, 1993), Nwana (1996) offers a survey of software agents, including a bottom-up classification of agents based on function and domain. Wooldridge and Jennings (1994) approach the issue from the top-down, addressing agents as intentional systems. Generally definitions of agent include the following capabilities (Milewski and Lewis, 1997):

1. An ability to work asynchronously and autonomously;
2. An ability to change behavior according to accumulated knowledge;
3. An ability to take initiative;
4. Inferential capability;
5. Prior knowledge of goals and methods;
6. Natural Language (use and/or understanding);
7. Personality.

Norman (1994) highlights this view of intelligent agents as "human-like automatons, working without supervision on tasks thought to be for our benefit, but not necessarily to our liking" (p68). The factors that Norman suggests should be considered when designing intelligent agents include:

a) Ensuring people feel in control;
b) The nature of human – agent communication;
c) Built-in safeguards to prevent runaway computation;
d) Providing accurate explanations;
e) Privacy concerns;
f) Hiding complexity whilst simultaneously revealing the underlying operations.

This use of an agent is seen as indirect management, in which the agent takes the role of as personal assistant who is collaborating with the user. Maes (1994) introduces two criteria that summarize the issues raised by Norman:

Competence – how does an agent acquire the knowledge it needs ?
Trust – how can we ensure a user feels comfortable using an agent ?

These factors relate to the paradigm of an agent as providing indirect management, in which the agent behaves as a personal assistant that is collaborating with the user in the same work environment. The assistant may learn, or may be taught, or may be created with the necessary knowledge to help users (e.g., in the form of user preferences). However, in this mode of operation, where the agent acts as a personal assistant in much the same way as a travel agent or real-estate agent, the user must have confidence in the agent and trust them to act in their best interests (or at least attempt to).

2.6.2 Agent-based personal assistants

Nwana (1996) classifies software agents into 7 groups, however the classification is contentious as it combines elements of functionality (mobility, learning) with what the agent does or how it behaves. Nwana's definition of an "Interface Agent" emphasises learning and autonomy in order to perform tasks for their 'owners' in the role of a personal assistant. This includes Assistants, Guides, Memory Aids, Filters and Critics, Matchmaking and Referrals, Agency (buying and selling on someone's behalf) and Entertainment (p219). Agents that manage user access to information (Information Agents) are not identified as interface agents. These agents are seen to include search, filtering and other user-related tasks. Other classifications (Smart Agents, Collaborative Learning Agents) also include elements of intelligent and adaptive user interfaces. Lashkari et al., (1994) view collaborative interface agents as agents which 'watch over the shoulder' of the user to observe and detect patterns which can then be automated – obviously an adaptation of the user interface. Kushiro et al., (1996) adopt the agent-based paradigm for the design of agent-interfaces to consumer products (although this is implemented from the user interface management software viewpoint, rather than as an intelligent dialogue partner).

It is therefore necessary to adopt a more inclusive definition of an interface agent which captures the role that an agent plays in manipulating both the information supplied to, and the interaction with a user. An Interface Agent may therefore be defined as *a program that can affect the interface without explicit* *instruction from the user* (c.f. Lieberman, 1997). This definition also captures the important capability of an interface to adapt autonomously by instigating or changing a dialogue. Lieberman (1997) distinguishes between interface agents as agents that actively assist a user in operating an interactive interface, and autonomous agents as software that takes action without user intervention – but highlights the need for agents which fulfil both of these definitions.

Beale and Wood (1994) present a more detailed classification that is more closely related to the role that the agent adopts when working with the user. This serves as a useful taxonomy to briefly survey the literature of interface agents.

2.6.2.1 User Agents

User Agents are adaptive, learning, self-customising pieces of software. This includes systems that intelligently filter and file information, systems that learn and automate based on user behaviour and preferences. For example, Kullberg (1995) presents an intelligent agent that learns to sketch annotations on an electronic calendar. Maulsby et al., (1993) present a simulated prototype of an instructable agent. Users instruct the agent, and the agent prompts the user. Sen et al. (1997) examine agents that can facilitate and streamline group problem solving in organisations. This includes managing multiple preferences for things such as meeting times. Hoyle and Lueg (1997) examine a commercial agent (Open Sesame!) for managing the Macintosh desktop, and conclude that "situatedness is fundamentally neglected in personal assistant design" (p55). For example, Hoyle and Lueg note that to although Open Sesame! correctly observed that the user has emptied the trashcan several times immediately after dragging a document into it, it cannot infer that this action also depends on the type and content of the document. The agent needs contextual information to situate the task if it is to correctly infer user actions.

Lashkari et al., (1994) emphasise that agents that learn by observing user behaviour take time to become useful and cannot extrapolate to actions the agent has not seen the user perform. Their solution to this is to allow interface agents to collaborate and share expertise. This appears to work, although it assumes a homogeneity of users which might not be the case in the real world.

2.6.2.2 Agent Guides

Agent Guides are agents that act as tutors or guides, supplementing user knowledge and skill. This includes "wizards", "coaches", Guides (Laurel et al., 1990), tutoring systems, etc. These agents assist the user in carrying out a specific task or set of tasks (Dryer, 1997). Wizards – software that guides the user in tasks such as the installation of software - are not necessarily "intelligent" but may be perceived to be because of their task-specificity. Guides (Laurel et al., 1990) provide task assistance by monitoring a persons' interaction and providing assistance accordingly. Lester et al., (1997) address the effect of animated personal agents in learning environments, concluding that they are a powerful tool to support learning (although data was subjective and did not include a "no agent" case).

2.6.2.3 Autonomous Agents

Autonomous Agents are agents that work on behalf of the user without any interaction or input from the user. For example, Kautz et al., (1994) use autonomous agents in conjunction with agents encapsulating user preferences to enable visitor scheduling in a commercial laboratory environment.

2.6.2.4 Symbiotic and Co-operative Agents

Symbiotic and Co-operative Agents are "Agents that are in there with you" (Clark and Smyth, 1993); they assist the current task by providing alternate views and additional relevant information. For example, Letizia (Lieberman 1997) is a co-operative interface agent that treats world-wide-web search as a co-operative venture between user and computer. It monitors current content and compares it with “nearby” pages using user preferences.

2.6.2.5 Anthropomorphic Agents

Anthropomorphic Agents are agents that imitate humans, or certain elements of human behaviour such as personality, etc. This concept of an agent which can instigate a dialogue with the user may be seen as a convergence between the intelligent and adaptive interface perspective of agent, and a vision of what it might be like to interact with a computer drawn from science-fiction (e.g., see Etzioni and Weld, 1995, Nardi et al., 1998). Anthropomorphizing computers and the conception of agents which fulfil human-like roles has prompted the development of agents having a range of human-like characteristics that are intended to enhance their communicative abilities (Negroponte, 1990; Kay, 1990) i.e., the agent displays some form of human-like intelligence in its ability to carry out a conversation with the human user. For example, “Guides” (Laurel et al., 1990) are introduced in the context of a multimedia database. They are representations of prototypical characters from American history that function as interface agents. A Guide provides navigational assistance by suggesting related material. The Guides are generic characters from the period, with attributes (gender, occupation, costume, etc.) that go some way to defining their likely character. Bates (1994) highlights the importance of such believable characters in providing the illusion of life, particularly in the arts, such as animation of cartoon characters. However, it is not clear how much explicit character needs to be ascribed to an object in order for the object to be perceived as having character, nor whether this is necessary for interface agents. For example Friedman (1995) found that 83% of subjects attributed agent qualities to computers in general, and 21% held computers morally responsible for error, without any attempt being made to explicitly give the computer human-like characteristics. Reeves and Nass (1996) examined the ability of users to distinguish between a variety of media (including computers) and “real-life” and found that people ascribe personality to various sorts of media without the media exhibiting any sophisticated behaviour. In particular, some media provide contextual cues that affect the perception of intelligence and ability of a system. Funk and Lind (1997) tacitly imbue agents with personality by raising the question of what makes an agent friendly. Friendliness can in general be seen as a social attribute indicative of co-operation. Castelfranchi et al., (1997) advance several reasons for giving agents personalities, based around the importance of increasing believability and providing narrative capabilities (i.e., to be able to conduct compelling dialogues). This also plays a part in social dialogue control (Thorisson, 1993). An important aspect of personality concerns the human ascription of personality to entities that possess motivations, particularly in predicting behaviour. This may be seen as a way of managing complexity, although Nardi et al., (1998) note that anthropomorphism in an agent interface is incongruent with the goal of unobtrusiveness, and this may make the system more subjectively complex for the user.

2.7 Combining Taxonomies

Comparing the taxonomies developed for Intelligent and Adaptive User Interfaces with Agent taxonomies, various similarities can be observed. In a general sense an interface agent can include the ability to maintain user dialogue preferences, and mediate between the user and the domain. There is obviously an overlap in the ability of IAUIs to provide interventionary assistance (Help and Critiquing systems) and co-operative agents. Personal Assistants and User Agents generally both provide the capability to adapt to the user based on user preferences or performance.

The classification system presented in section 2.4 may therefore be used to categorise both IAUIs and Interface agents as shown in Table 2.2. The ability of the system to self-adapt (in IAUI terms) becomes the ability of the agent or system to learn (from the behaviour of the system and the user).

Factor	Example categories
What is adapted to	User preferences, User behaviour, System behaviour.
How Adaptation takes place	Automation/Initiative (changes in interaction), Changes to the interface.
Ability to learn	User behaviour, System behaviour.

Table 2.2 Classification Criteria for IAUIs and Agents

The crucial insight from the proceeding discussion is that the paradigm of an agent essentially subsumes that of an IAUI by providing a conceptual wrapper which encloses the ability of the software to intelligently adapt the interface. In addition to the functions performed by an IAUI, the agent paradigm also introduces the possibility of visible intelligence and personality, moving even closer to the model of human-computer interaction as conversation with an intelligent entity embodied in software. From this perspective it can be seen that Interface Agents are, in fact, IAUIs. They perform the same function of active, intelligent decision support (Bird and Kasper, 1993), and they do so using a dialogue which is that of an *assistant*. This results in an agent dialogue which is asymmetric in that because the agent is assisting the user it may instigate a dialogue autonomously but it will never take the initiative in acting unless it has been told to do so by the user (e.g., see section 1.4.4).

The paradigm of interface as agent has made explicit conceptual elements that are also present in some IAUIs. These elements are highlighted because of the agent-based paradigm itself and centre around the concept of the agent adapting to the user, rather than the interface or the computer performing the adaptation (although both computer and interface do, of course, adapt as a result, and consequently the entire system may be viewed as adaptive). Also important to note is that the use of a dialogue based on delegation within a specific domain, and the collaborative nature of the resultant interaction, enable the user to anticipate behavioural change within the interface in a far more predictable fashion than was possible with some IAUIs. Because the agent is related to the task set in a more concrete fashion, and its role is made more specific, the user can place the agents' behaviour in a context that improves interaction.

2.8 Performance Predictions

The literature on both IAUIs and Agents predicts improved interaction when compared with static user interfaces. This advantage is emphasised when the systems to which they act as the user interface are complex. However, the literature surveyed by no means consists solely of examples where interaction has been enhanced.

The common ingredient of successful implementations seems to be the consideration of context. When the IAUI or agent has access to contextual components of the interaction, the system appears to be effective. Because an agent-based framework makes more contextual elements visible to the user (delegation-style dialogue, domain constraints) the user's expectations of how the system will behave can be managed. Returning to the examination of the knowledge required by such a system, an important component in both IAUI and Interface Agents is knowledge of the *user*, and knowledge of the *task and domain* (the *purpose* of the system). Systems that have a clear definition of these elements consequently make context more explicit. This can be seen in the results of Kossakowski (1989), Trumbly (1994), and other systems where the range of responses, tasks, domain and user characteristics are well defined. Conversely when there is no clear model of context, interaction becomes more problematic, as Hoyle and Lueg (1997) allude to when they point out that situatedness (i.e., context) is often ignored in the design of personal agents. This lack of context consequently reduces the effectiveness of the system. Some failures of IAUIs can be identified as failures to adequately model the task, domain and user context (e.g., Hockley 1986; Copeland 1993). Brown and

Duguid (1994) highlight the importance of context in relation to designed artefacts:

> "…context is an essential component of communication and a major source of simplicity and efficiency, yet many approaches to interface design aim for or proclaim "self-evidence", which implicitly or explicitly assumes that context-independence can be achieved."

A retrospective evaluation of Intelligent and Adaptive user interfaces and Agent-based interfaces provides the critical insight that the common factor contributing to their effectiveness is *a model of context*.

The extension of agent technology to a widespread user population therefore brings with it some challenges (even though catering to heterogeneous user populations is one of the goals of IAUIs and agent systems). Many consumer-based systems are complex (according to the concept of complexity defined in this thesis), and therefore appear to be prime candidates for agent-based interfaces. However, these systems have ill-defined usage contexts, which consequently means that the implementation of good agent interfaces may be problematical.

2.9 Summary

This chapter surveyed the literature on intelligent and adaptive user interfaces as techniques for managing subjective system complexity by managing the interaction of the system and the user. The rationale for this is that by tailoring the appearance and behaviour of the system to the user, subjective system complexity can be reduced. The literature shows that IAUIs do not seem to provide a convincing increase in performance, and that a new research paradigm has consequently arisen supplanting IAUIs – that of Software Agents. By surveying the literature on software agents, and relating their capabilities to those of IAUIs by developing a classification system, Interface Agents have been shown to encompass and extend the capabilities of IAUIs, with potentially enhanced capabilities for managing interaction and complexity.

However, the literature evaluating Interface Agents indicates that there is no consistent improvement in performance. The paradigm of Interface Agent allows some insight into why this improvement is missing, because the conception of an agent makes the components required in their construction, (and consequently their behaviour) more explicit then those of IAUIs. A key element in successful agent systems can therefore be pinpointed as *context*. Agents with components that allow interaction context to be modelled and maintained to seem to provide an increase in performance and greater usability.

Chapter 3 examines the importance of context in more general terms, and develops a model of context that is appropriate for complex systems with diverse user populations. This model will subsequently be empirically evaluated in Chapter 4.

Chapter 3

The Importance of Context

3.0 Static and Dynamic Context

The previous chapter examined the use of intelligent interfaces and agent-based interfaces and identified that appropriate use of context is a critical factor in their successful deployment. This chapter addresses what context is when considered as a factor in interacting with computer systems. This chapter develops a model of context as applied to the implementation of intelligent interface agents in information systems, using an active model of agent operation called *contextualization*[1]. The conceptualisation of contextualization as a process (rather than as a state) provides important insight into the design of interactive systems, as interaction is a dynamic process and consequently causes context to change. The resulting contextualization model will be used to identify several heuristic resource characteristics that can be made use of in agent interface design, with the objective of improving the user-system interaction.

3.1 Models of Context

Context may typically be viewed as the environment of communication that enables the intended meaning to be ascribed by the recipient of some data. For example, in uttering the phrase "please shut the door," the context includes referred objects, world knowledge, social rules, etc. The way in which a construct or artefact is used to convey meaning therefore implies the use of elements of context.

Context may be defined as:

---

[1] Some of the ideas in this chapter were first discussed in Edmondson and Meech (1994), and Meech (1995)

> The circumstances that form the setting for an event, statement or idea, and in terms of which it can be fully understood and assessed.
> (The New Oxford Dictionary of English, 1998)

Interaction may therefore be viewed as an event (or set of events), the circumstances of which constitute context. In order for the event and its consequences to be understood, the appropriate context must be provided. Consequently if the required interaction is to be conveyed to a user, the elements of context must also be clearly signalled.

3.1.1 The Dynamics of Context

Implicitly the definition considers context to be static - it is defined for a single moment in time, or in relation to a constant configuration (of words, objects, etc.). In a dynamic world, context is created when required, using the available cues and information. In language use, meaning is ascribed by the recipient using the environment (or *context*) in which the communication takes place, resulting in *information*. Thus there are a number of contextual factors that are dynamically integrated by a process to produce meaningful data. This process of *contextualization*[2] involves both the immediate data, the history of the data (e.g., what has preceded a word in a text or utterance, or visual information in the case of a sign), and the knowledge already possessed by the recipient (i.e., mental models, general knowledge, etc). The recipient's knowledge will include a wide range of information that will also provide contextual information, including social factors and cultural issues. From this perspective it can be seen that context (in the static sense) is often used obliquely in user interface design, where the concept of metaphor is used to recruit contextual understanding

---

[2] The Latin origin of context is *contextus* - to weave together - indicating its active origins.

from the user, e.g., using “desktop” icons such as folders (to store things), trashcans (to remove things), etc.

3.1.2 The Scope of Context

In dealing with some systems it is difficult to identify the scope of contexts that play a part in the interaction, and boundary factors which lie outside the current contextual focus. Brown and Duguid (1994) see the use of systems and artefacts as being supported by what they term *border resources* – resources that are outside the scope which is normally recognised as providing context. These resources encompass a range of social, personal and learnt elements that are made use of by users, and they develop over time into more stable conventions or *genres*. For example, calling a movie a “thriller” places it into an established genre and consequently establishes a recognisable context. Importantly these elements are seen as not being inherent or self-evident and cannot be predefined – they are socially constructed, and consequently dynamic. In terms of interactive systems, genres place them in a similar context, but often there is no established genre that completely fits – the context is ill-defined. In order to extend these insights into a more concrete form that can be made use of in an interface, it is therefore necessary to determine what constitutes context and associated resources for these systems.

3.2 Context in Human-Computer Interaction

The field of information systems design has recognised the importance of identifying the context in which the user is interacting, and this has led to the use of several methods which are used at the requirements stage of design to capture elements surrounding the use of a system in it’s environment. These methods include *ethnography* (Simonsen and Kensing, 1997) and a range of methodologies which attempt to explain and predict user behaviour within a particular environment, such as *situated action*, *activity theory* and *distributed cognition* (Kushmerik, 1996; Holm and Karlgren, 1996; Nardi, 1992; 1996). Less structured approaches use context sensitivity as a means of producing task coherence in a wide range of organisational contexts (Hellman, 1990). All these approaches to understanding human behaviour arose from a realisation that behaviour cannot be entirely predicted from the users' knowledge of the task, and the device which can aid in performing the task. There is, however, some discussion on how these theories feed into the design process and how different they are from more traditional models from cognitive psychology. The situated-action argument is that behaviour is determined by context (social and organisational) and that analysing the context in which technology is used is more valuable than trying to understand and model the underlying cognitive processes. Green et al. (1996) see the two approaches as being linked, as a user will bring standard cognitive processes to any interaction with an artefact, but the way in which the processes are used will depend upon the context. The overall consensus is that contextual factors play an important part in determining behaviour, and should be included in the design phase of a project.

In terms of cognitive models of behaviour related to context, Hollnagel (1993) distinguishes between models that emphasise the sequential nature of cognition and models that view cognition as being determined by the context in which it takes place. Hollnagel terms these *prototype models* and *contextual control models*. The contextual control model is based on the concept that the selection of one action from the many that are possible in a given situation is determined by current needs and constraints and not by the normative characteristics of the component action. This can be viewed as a separation of control actions and competence, although it could be argued that the control modes that Hollnagel sees as comprising competence simply serve to partition the action space as *types of performance*. These in effect constitute a set of possible contexts and ways of moving between them.

In human-computer interaction (i.e., the use phase, rather than the design phase) context is recognised as playing an important part in the use of complex systems such as commercial aircraft (Chappell and Mitchell, 1996) and a variety of decision-making tasks (Kirlik and Markert, 1990). In computer systems, Schweighardt (1990) sees context as being made up of the objects referred to by the users and the users themselves. This may take the form of a dialogue history, or a set of representational (e.g., pictorial) cues or indication of task status, etc. and perhaps some model of the user (e.g., their preferences).

Once again, these approaches see context as a static, well-defined concept based on the domain and tasks under consideration. Cooper (1991) (echoing Brown and Duguid, 1994) advocates the necessity to look beyond the immediate in defining context, and emphasises the ability of people to construct an underlying reality in order to resolve contradiction and inconsistency. In this way context may be viewed as the result of the active process of *contextualization*. In social interaction people orient themselves towards a sense of context and use it as an interpretative resource. The primary means for creating and sustaining a sense of context is the use of language – context is an *interactional* achievement. In dialogue management, Young (1991, p161) sees context has having both static (persistent) and dynamic (transient) components. Because of this, context should not be viewed as a static entity, and there is no certainty that consensus (shared context) will be reached – context may be viewed as contested rather than agreed. Maskery and Meads (1992) develop the concept of shared context with a *focus* (Figure 3.1). In this model interaction is used to provide a common context, within which there is in turn a focal point. The common context and focal point can change in size and overlap dynamically, ideally maintaining the appropriate overlap. Schweighardt (1990) also sees the distinction of levels of context as a focal point which moves as the context changes, (i.e., as contextualization takes place) echoing Brown and

Duguid's (1994) concept of focus and border. The movement of this focal point may therefore be understood as the process of contextualization.

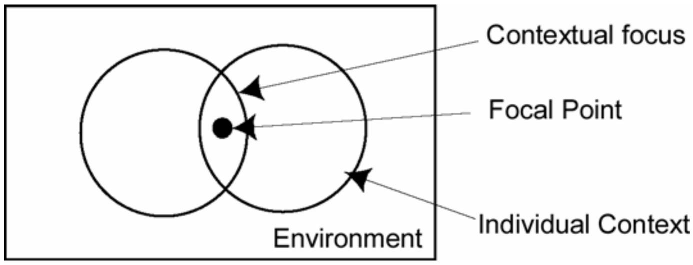


Figure 3.1 Focal Context

There appear to be no clear cognitive theories relating context and interaction (e.g., see Maskery et al., 1992a,b; Cooper, 1991 for discussions). It is clear that context is dynamic and changes with interaction, but it is unclear how each style of interaction changes context, and what effect this has on the user. The significance of context for the development of representation and interaction models arises from the fact that contextualization is important as an active process - it is not a passive constraint on activity.

3.3 Context and Interaction - the Dynamics of Contextualization

Context is constructed dynamically, and this process may be thought of as contextualization. Human-computer interaction is inherently dynamic, and consequently it is obvious that context is constructed during the interaction. This may be viewed as contextualization taking place in both the user and the computer with which the user is interacting (Figure 3.2), which leads to the production of a shared context that facilitates interaction (in effect this is constitutes the focal context for the interaction).

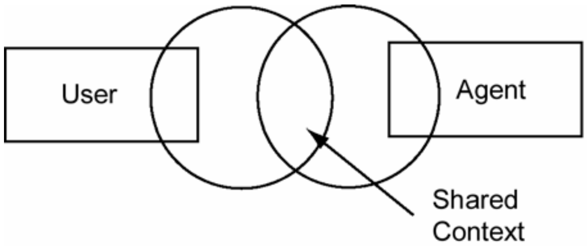


Figure 3.2 Shared Contextual Focus

The important question now concerns how interaction affects the process of contextualization. From previously surveying the field of human-computer interaction (Section 1.4.4) it is possible to identify three conceptual interaction modes and compare the contextualization that takes place in each case. These interaction modes are *Tool Usage*, *Agency Mediation* and *Communication*. Each of these modes of interaction captures user-system interaction relating to the way feedback is controlled by the system due to the degree of autonomy possessed by the system. This implements a form of dialogue control in which the expected feedback provides contextual information to the user.

3.3.1 Tool Usage

In this mode of interaction there is no automation, and contextualization exists as a process *only within the human.* There is no active contextualization by the tool. The tool and its circumstances of use must provide sufficient data to enable the appropriate contextualization to take place. The tool usage metaphor obliges the designer to ensure that the system provides the necessary contextualization cues for the user to understand the context. In this case the computer functions as a tool and provides the user with contextualization data in the form of clearly represented functional potential and feedback concerning performance. These may take the form of icons, buttons, labels, etc.

3.3.2 Agency Mediation

This mode of interaction exploits contextualization in a similar way to human-human communication with the computer agent adopting a subordinate role (e.g., a nurse in an operating theatre, or a soldier or butler being given orders). As such the contextualization is asymmetric because the role taken by the agent reduces the contextualization scope, and this must be understood by the user issuing requests. For contextualization to play a role in agency mediated interaction it is necessary for the asymmetric dialogue patterns to be modelled within an appropriate context. The viability of this approach rests on the extent to which sensible constraints can be placed on the domain of interaction in order to force the dialogue to depend only on limited exploitation of context. The key to agency mediation lies in the correct identification of the reduced contextualization scope.

3.3.3 Communication

Contextualization in human-human communication requires each communicator to adapt dynamically to the response of the other in order to ensure the intended contextualization process takes place. The metaphor of human-human communication for HCI requires the design of a dialogue system that can contextualize as capably as the human user in a range of domains. This appears to be unrealistic for the near future because linguistic theory does not yet encompass the complexities of pragmatic organisation of human discourse, and therefore a model of discourse is not available for instantiation in a computational form.

From this discussion it can be seen that the way in which contextualization takes place depends upon the style of interaction. The aim of the designer of an interface is to ensure that the interaction is supported in its requirements for contextualization. Therefore, in addition to any other functionality, the computer must serve as a device for enabling or facilitating contextualization by the human user in order to promote appropriate interaction.

3.4 Contextualization Requirements In Human Computer Interaction

The tool usage metaphor requires that the relationship between the task(s) and the tool(s) available should be clearly defined and unambiguously available to the user who is solely responsible for correct management of contextualization. The tool designer, however, does have some responsibility for ensuring that the functionality of the tool is readily appreciated (and indicated to the user). The criterion for choosing a tool metaphor for the interface is that the management of contextualization is both naturally and readily undertaken by the user.

In the case of tool usage, complexity is managed through careful attention to the needs of the user, in order to facilitate contextualization. In the case of agency mediation the same holds true, but additionally the complexity revealed to the user may be reduced because the agent is managing some of the contextualization on behalf of the user.

To illustrate the above points consider as an example the services provided by a travel agent. In principle, these could be provided in large measure by a complex menu-driven system for issuing tickets, bookings etc. However, the agent-based implementation offers the user the freedom to be imprecise in initial specification, and to vary simultaneously several parameters of the booking. The agent, in this case, is performing complexity reduction by contextualizing the traveller's requirements in the special domain in which the agent is skilled. Note, however, that the agent will have to selectively supply the user with some aspects of the contextualization domain (restrictions on seat, weekend rates, etc.,) and the user must be able to understand these. The expectations of the user must be reconciled with the capabilities of the system. Users - even as travellers - may require training.

The agency mediation metaphor requires that the user understand the overall limitations of the agent and the specific capabilities of the agent in relation to the task(s). This means that the specification of the agent's capabilities must be clearly available to the user, and here also the designer has an important role to play. Additionally, however, the agent's performance must be fed back to the user, and this places a further demand on the designer. The criterion for selecting agency mediation as the metaphor is that the user naturally and readily accepts the use of a recognisably appropriate subordinate to which components of the task can be assigned, because such recognition implies awareness of the contextualization limitations.

In designing an intelligent agent-based interface it is therefore possible to speculate that an agent should assume a subordinate role and maintain an appropriately asymmetric dialogue with the user. In order to manage this process the agent must maintain a model of context and be able to dynamically manage the process of contextualization.

3.5 Contextualization in Intelligent and Adaptive User Interfaces

Several authors have attempted to make use of context in the design of adaptive interfaces, with the motivation to improve the task-to-tool mapping and subsequently improve interaction (Croft, 1984). Here it is important to recognise that *adaptation* refers to *the ability of the system to act appropriately in a particular context*. However, the conception of context is static; its scope is already constrained by an implicit context, usually represented by a task (Tyler and Treu, 1989). For example, in Croft's work the ability of the interface to adapt successfully is because the behaviour of the system (the task context) is already anticipated by the user as "edit", "form filling", "email" or "calendaring". This is because the user has recognised the next (higher) level of context is "*office tasks*" (the *genre*), and they know what it is possible to do in this context. The aim of adaptation should therefore be to enable the user to recognise this context in the first instance.

The distinction between the form of dialogue control and the level of "automation" or dialogue adaptation in a contextualizing interface accounts for related work on initiative in collaborative planning (Cohen et al., 1998). Work on initiative as a dialogue control mechanism (i.e., which dialogue partner can take the initiative in triggering a dialogue) can be related to the context in which an utterance is made, and how this triggers contextual understanding in the recipient. e.g., Utterance classifications can be made which signal the intended role to the recipient:

Assertions: retain speaker control;

Commands: retain speaker control and instigate action;

Questions: act as commands, except when in response to a question or command;

Prompts: give hearer control.

Theories of initiative can therefore be viewed as methods of indicating contextual elements of role in a dialogue process.

In terms of cognitive science, context can therefore be thought of in classical terms as being made up of both *representation* and *process*. So far, these components have been discussed in a general sense in order that their properties could be discussed in relation to general models of human-computer interaction, emphasising interaction as a triggering process. In order to relate these elements to agent-based interfaces, it is first necessary to identify what the representation and process consist of in an agent interface that provides contextualization.

3.5.1 Elements of Context and Contextualization

The preceding discussion has outlined the importance of contextualization as an active process in human-computer interaction, and by considering contextualization when interacting with tools, agents and people has identified several factors that must be taken into account. These factors relate to whether the computer can contextualize or not - that is, actively try to construct context both for the user and on it's own behalf. These factors relate to dialogue instigation and the awareness of role adopted (tool, agent, person) in an interaction. Examining the literature on context (Mittal and Paris, 1993) the following are also identified as components of context:

1. The problem solving situation (the Tasks);
2. The participants involved (expertise, beliefs, goals, etc.);
3. The mode of interaction in which communication is occurring;
4. The discourse taking place;
5. The external world.

Here the elements of dialogue and role are encompassed by *mode of interaction discourse*. If we consider the interaction of a single user and we constrain ourselves to the process of contextualization within the agent, the agent correspondingly requires the following:

1. An identification of the user (expertise, beliefs, goals, preferences, etc.);
2. A description of tasks that might be carried out;
3. The available interaction styles and the discourse possible in each case;
4. A representation of the requirements of the domain in which the agent is situated (the "external world").

As an aside it is worth noting that these features encompass the elements that constitute an IAUI (Section 2.3), confirming the earlier claim that agents extend the IAUI model. The domain is seen as providing a framework in which the tasks are situated and given structure.

In terms of tasks, this structure can be viewed as a graph hierarchy which decomposes the domain into a tree structure of tasks and rules which indicate when a particular task is to be performed (e.g., see Carey et al., 1989). Moving through the tree from root to leaves gives progressively more task detail, to the point at which an action can be taken to complete the task. Hierarchical Task Analysis (e.g., Carey, 1989) may therefore be viewed as a process which analyses context and provides a representation for it. The graph itself provides a representation of static context, and the rules for traversing the graph go some way to capturing dynamic context. However, these rules are fixed and reliant only on the domain, and consequently are not enough on their own to provide contextualization (for this would require "perfect" user behaviour, without negotiation thorough interaction).

For the user, the task being undertaken is necessarily part of the context for interpretation of feedback data from the system (regardless of the interaction metaphor used). A task has a set of contextual requirements associated with it and these requirements must be unambiguously represented in order for the task to be recognised and understood. However, the representation of the task is not enough to provide context. The style of interaction that relates to that representation is also important, as is the context in which the task is being performed. Edmondson (1991) has argued that the style of interaction adopted should be based on the *underlying* cognitive behaviour of the user in relation to the task. A task analysis should be concerned with identifying the underlying behaviour (for example making selections) and with supporting that behaviour in the interaction (for example through a menu-based interface). This concern with supporting underlying user cognitive behaviour requires that the support for contextualization has to be understood in relation to the behaviour, and the interactional process must take this into account. The task analysis, however, does not automatically identify the most appropriate representation for the contextual data (see, for example, Hutchins et al., 1986). The task context is an indication of where in the task hierarchy an action takes place. For example, pressing the "1" key on the computer keyboard may be thought of as a task, but it can take place in many contexts, depending on the goals of the user.

When designing an interface to support contextualization the interface designer is concerned with representing task relevant data in a way that is well matched to the user's cognitive needs and task expectations. A task analysis must also contribute to the identification of the appropriate style of interaction and its unambiguous presentation to the user. The implication for task analysis is that explicit attention must be paid to the contextualization needs of the user.

The implications for an agent-based interface concern providing the agent with the correct task space, an appropriate model of the user, the scope of the agent and the interaction styles that may be used.

Implicitly the agency mediation paradigm reduces the contextualization scope in two ways. The first of these results from the asymmetric dialogue (regardless of interaction style). The issue of an order or instruction to an agent - whether a synthetic character or a system daemon - will be matched to the capabilities of the agent. The person issuing the order needs to understand these limits and this constitutes a constraint on contextualization. The second constraint arises from limits on the potential behaviour of the agent. The agent will be autonomous - but only within defined limits (i.e., the scope of the agent's competence). For example, a travel agent does not provide services of a medical nature. The agent's behaviour amounts to surrogate contextualization, and this is of limited scope.

The management of complexity is intimately related to the management of contextualization. The purpose of an intelligent interface therefore becomes one of managing the process of contextualization. In this sense a good interface, intelligent or otherwise, is most profitably seen as one which *promotes* contextualization.

3.6 Contextualization in Agent Mediated Interfaces

The result of the preceding discussion leads to the conclusion that interface agents are most profitably modelled as systems that actively provide context to the user (i.e., promote contextualization). An Intelligent Agent needs to manage context because in order to successfully interact with the user, it must understand the context in which it is interacting and provide a suitable dialogue.

The communicating agent metaphor implies something about the way the system behaves - it behaves more intelligently (i.e., like interacting with a *person*). The onus on recognising what the system can and cannot do, when and how to act, etc. is shared between the computer and the user. This is apparent in the way dialogue takes place between user and system – the system can also instigate and control dialogue (or take the *initiative*).

Context may be thought of as having both explicit and implicit components. From the perspective of intelligent, adaptive and agent-based interfaces, context is explicitly constrained by task-based design. Therefore the more constrained the tasks and the task environment, the better the assistance works; i.e., the more certain the agent is of what the user is trying to accomplish, the more targeted the assistance can be. As we move towards the border of context, the task environment (implicit context) is a separate factor. There must be sufficient context to allow contextualization to move outwards to understand more about the domain itself. For example, dealing with email as a task cannot capture the behaviour based on content, even though the number of command options may be limited (send, reply, compose, etc.). But this does not take into account the content of the mail itself (i.e., the meaning to the recipient), nor does it consider the reasons for sending mail. An agent interface must address both implicit and explicit context in order to fully manage the contextualization process.

To provide contextualization, all elements must be made as explicit as possible to the user, and the agent must have some representation of the elements of contextualization in order to make use of them. From the preceding discussion, Table 3.1 shows the factors that should be considered.

Contextual Factor	**IAUI Factor**
The external world (the situated context)	Knowledge of overall system
The set of tasks (the focal context)	Knowledge of tasks
A model of the user (preferences, etc. – the user context)	Knowledge of the user
The form of dialogue (elements of agent role context)	Knowledge of the interaction
The representation of the dialogue (physical representations of context).	Knowledge of the interface

Table 3.1 Components of Contextualization and IAUIs

It is interesting to note that these elements, composed in a top-down fashion from the theoretical perspective of contextualization, mirror those identified in a bottom-up analysis of Intelligent Adaptive User Interfaces (Section 2.3). Comparing the Contextual Factors with the knowledge identified as being necessary in an IAUI (Table 3.1) we can see a close match between the categories. This strengthens the view that IAUIs are actually systems that are intended to contextualize, but that this may not always be recognised by their designers. Consequently, many of the IAUIs neglect aspects of context which reduces their effectiveness.

There are some important aspects missing from the list above, and these concern elements of implicit context (and are related to the user). This insight comes from the realisation that agent communication is closer to the space of human-human communication than it is to that of tool-usage. Traditional conceptions of context make little use of contextualization factors that are not explicitly part of the interactive scenario, and that are obviously border resources in terms of contextualization. Recent work, such as Salker et al., (1999), continues to view context as a phenomenon which can be accounted for by "toolkits" of sensor data. This work does, however, recognise the dynamic nature of environmental context, and that context must be acquired from multiple, distributed and heterogeneous sources. But the concept of *environmental* in this sense does not include the *user*. Thus the important insight is that there are a range of contextualization factors which people make use of to understand communicative dialogues that are not direct components of the dialogue itself, nor are they directly part of the task or the domain. These elements include personality, emotion, attitude, appearance, body language and a range of other social cues. It is therefore possible that in order for the user to correctly contextualize the system, some of these background, *ambient* (Lachman, 1997) or social factors should be taken into account and used to help the user contextualize.

To address such social contextual factors, it is profitable to re-visit issues in agent technology in a top-down analysis of suggested agent uses, and connect this with the concepts of contextualization. This concerns the development of *believable agents*.

3.7 Extending the Scope of Agent Contextualization: Believable Agents

So far, an agent has been viewed as a relatively anonymous program that interacts with the user through various interface styles. However, one aspect of contextualization is only partly considered through this viewpoint, and this concerns the process of contextualization within the user based on their expectations of what the agent can do for them. In surveying the proposed deployment of interface agents, previous discussions (Section 2.6) highlighted the progression of agent incarnations from configurable rule-based systems to human-like anthropomorphized agents.

The development of agents has so far been presented from the perspective of aids to the user in performing a set of tasks. However, some researchers approach the design of agents from the perspective of entertainment – agents that are *fun* to interact with (as an end in itself). Much of this work has been driven from the use of agents as characters in interactive narratives and game systems. The objective of such systems is to provide an experience that is believable enough for a user to *suspend disbelief* in a way which allows users to interact with agents as if they were “real” or “living entities” (e.g., see Foster and Meech 1995; Kessler and Kilgore, 1997). Perlin and Goldbers’s (1998) rationale for this believableness stems from the desire to convey mood and personality in order to express emotional messages or to portray specific characters rather than to solve a particular problem. They note that in certain applications the *way* in which the agent provides information can be as important as the information itself.

Several types of agents, particularly Agent Guides (Section 2.6.2.2) and Personal Assistants (Section 2.6.2) could conceivably be implemented in a way in which the appearance and behaviour of the agent influence the way in which people (users) react and interact with them. It is therefore necessary for a conceptual framework of contextualization to encompass and model the effect of qualities such as personality, friendliness, etc. on the user.

Agents have been designed to act as guides in virtual environments (Doyle and Hayes-Roth, 1997). In addition to the capabilities of the guides described by Laurel et al., (1990) these agents have a form of personality and variable moods, and consequently are intended to be more of a companion than a resource for the user. Elliott et al., (1998) describe agent behaviour in relation to the goals that a pedagogical agent has in regard to its "pupils." These goals include "I will engage the student", "The student should attend to me when I am talking to them," "The student should be cautious when it is appropriate," etc. These goals are used to prompt the appropriate behaviour in the user, and feedback from the user can be used to maintain satisfaction of these goals. Noma and Badler (1997) describe a *virtual human presenter*, an agent intended to act as an interface mediating communication between user and computer. These agents are capable of introducing products to customers – they are virtual salespeople, and need similar social skills.

An application directed more at the entertainment end of the interaction spectrum is *Creatures* (Grand et al., 1997), artificial creatures that inhabit a user's computer for entertainment; they have no other goal than to amuse and entertain. Work in theatre-based agents generally falls somewhere between pure "entertainment" and "guidance," in that the agents must be entertaining, but also guide the user through some form of narrative. For example, Wavish and Connah (1997) describe a project that integrates human-figure animation software with agent technology in order to produce virtual actors for films, games and virtual reality worlds. The rationale behind their approach is not to make the agents more intelligent, but to make them more convincing at portraying characters. Believable characters (or believable agents) gain their believability from exhibiting character in their behaviour. All these applications of agents extend their capability to act in a social setting, or adopt characteristics that are usually found only in people. They are generally viewed as being more *believable* than mechanistic agent systems.

Mateas (1997) reviews believable agents in the context of a virtual world inhabited by characters with which the user can interact (the *Oz* project). The following factors are seen to contribute to believability:

1. Personality;
2. Emotion;
3. Self-Motivation;
4. Change (personality change with time);
5. Social relationships;
6. The illusion of life.

In addition to believability, it is also necessary for the user to have trust in the competence of an agent and for the agent to exhibit social and ethical traits which enable the user to relate to the agent coherently. Friedman and Nissenbaum (1997) emphasize the complex social relationships that are possible (but not desirable) when interacting with an agent in which such ethical traits are not present. They relate Harold Pinter's screenplay for the film *The Servant* in which the roles of Wealthy Young Man (in control) and Butler (as agent) are gradually reversed until the *agent* is controlling the *user*. They conclude that the user must both *feel* and *be* in control in such a social relationship.

Virtual Theatre requires agents to portray characters in a script, and consequently a convincing model of mind (or intelligence) is viewed as less important then convincing character portrayal. One important element of character is viewed as the ability to improvise without a script. However, this goal seems somewhat at odds with the necessity of script-following to provide a story. Other studies also claim that interaction can take place without plot, script or narrative (Kessler and Kilgore, 1997). However, in the work described, the agents are always deployed in an application or task-specific context and this necessarily constrains their behaviour in a way that meets user expectations (so that interaction and guidance are appropriate). *Roles* are a concept which can be used to govern the way in which actors interact with one another in social situations, and this role based behaviour can be used to give improvisation based on elements such as personality and emotional state. This provides a high-level behaviour that allows improvisational behaviour (which provides consistent character without an explicit script). Hayes-Roth and van Gent (1997) present improvisation in "animated and smart" puppets that can be given high-level directions by children to construct stories. The high level instructions are used as the basis for improvised behaviour, and can consequently be viewed as providing contextual bounds to the behaviour of the puppet.

Mateas (1997) links *character* with *story*, and defines drama as consisting of characters, story and presentation. A story (or narrative) is an experience with a temporal structure and defined presentation. Interaction, as the ability to influence temporal structure and presentation, raises the question of what is meant by the term "interactive story?" By allowing a communicative dialogue through interaction, the agent must be intelligent enough to guide the story in a way that provides the required narrative as interaction takes place. Oz takes an alternative approach of confining the interaction to certain points in the narrative, and allowing branching to take place. This is an approach taken by many computer games and (semi-) interactive fiction, and constrains the number of paths that the agent must be capable of navigating. Kelso, Weyhrauch and Bates (1992) use real actors and a director to try out concepts of character and plot in an experimental setting. Real actors were given characters and a high-level script of a drama, and a participant interacted with them in a stage setting. The progression of the interaction depended on the behaviour of the participant (i.e., the character they adopted) and consequently the exact ending of the scenario depended on their personality and that of the actors they engaged with. Even within the limitations of a simple scenario (buying a ticket at a bus station), the two experimental trials yielded very different narratives, depending on the expectations and attitudes of the actors, highlighting the importance of the personality of the (inter)actors in changing the outcome of the interaction.

3.7.1 Empirical Studies of Believable Agents

Several investigations have been made which address the use of believable agent characteristics and their affect on user interaction with the agent. Koad and Maes (1996) survey a number of implementations that use representations of the human face as part of the interface. They argue that using a face as an agent representation is engaging and makes the user pay more attention, but requires more effort form the user to interact with the system. The benefits can be made to outweigh the disadvantages in certain applications where conveying emotional states of the agent to the user is important. In a similar study, Thorisson and Cassell (1997) examined visual feedback in a tutoring agent. However, in addition to facial expressions they also included a condition using non-verbal (*backchannel*) responses such as nodding, finger-drumming, etc. Facial emotion had no significant effect on ease of interaction, whereas there was a significant increase in the participant's rating of the agent and ease of interaction when the non-verbal cues were introduced.

Foner (1997) presents a sociological case study of people interacting with Julia, an agent designed to appear human in restricted environments while being entertaining and informative. Although interaction is text-only, it often elicits suprisingly intense emotional responses in those who encounter the agent. However, the context in which the agent exists is that of a MUD (Curtis, 1992), a shared space which exists for explorative games. Generally, players do not expect to encounter a non-human player, and when they realise that is what

Julia is, they are naturally surprised. The interesting observation from these studies is that users interact with relatively unsophisticated software systems in a similar way to interacting with other humans (The other well-known example being Eliza (Weizenbaum, 1966)).

In a wide-ranging survey of how people treat various forms of media as though they were "real people", Reeves and Nass (1996) examine the reaction of people to computers and media as a social phenomenon. Their findings suggest that people respond socially and naturally to such media even though they believe it is not reasonable to do so, and even though they do not think that such responses characterise themselves. To briefly summarise their findings:

1. Polite forms of interaction engender polite responses.
2. The perceived distance of a character from a subject affects the intimacy of response (e.g., the closer a representation of a face, the more attentive the user is, and the better the recall).
3. Flattery, even from a computer, gives better subjective evaluation of task performance, and makes the computer more likable (!) The same results hold for computer praise of performance.
4. Using representations of personality along axes of dominance-submissiveness and friendliness-unfriendliness, subjects recognize these personality attributes using a minimal set of cues (see also Nass, Moon, Fogg, Reeves and Dryer, 1995). This holds true when the personality is indicated by very simple cues.
5. In terms of emotion, using measures of arousal (intensity) and valance (good or bad), people respond to media and computers using these same dimensions as they would in reacting to real-life experiences.
6. In perceiving level of expertise, if you present a system as expert (just by giving it an "expert" label), people are more inclined to trust its responses.
7. In examining teamwork, a computer is treated as a team member in a similar way to a person.
8. Gender is also ascribed by users to systems, for example in speech based systems Male-voiced computers are viewed a) more seriously, b) as more knowledgeable, c) as having greater drive, d) more extravert and e) more intelligent than female-voiced computers.
9. The perception of voice is linked with an actor: Users respond to different voices on the same computer as if they were different social actors. Users also respond to the same voice on different computers as if they were the same social actor.

There are two important conclusions to be drawn from this work. Firstly, people treat computers as social actors, and secondly, the same contextual factors that people exploit when dealing with other people can be exploited by computers and media. Note that in these experiments there was no attempt to explicitly introduce the concept of agency: the computer is not even hinted at as being anything other than an inanimate object, however, people relate to the computers in the same way as they do to people. Nass, Moon, Fogg, Reeves and Dryer (1995) summarize this:

> "Human-computer interaction is fundamentally social and not anthropomorphic; that is individuals can be induced to behave as if computers are human, even though users *know* that the machines do not actually possess "selves" or human motivations" p225

The results of Reeves and Nass suggest that we can expect users to react to agents which are placed in social contexts as though they were people, and to consequently ascribe to them attributes such as personality, friendliness, emotion, expertise and even gender. The desire to build believable agents implicitly makes use of these responses to further reinforce the aspects of the agent which provide the social believability.

The motivation for providing believableness in agents can therefore be justified in terms of empirical findings, but it is necessary for a theory of contextualization to encompass this and integrate it with the framework produced so far. This theory (and the models that result from it) must capture believableness as it obviously affects the process of contextualization.

3.8 Contextualization in Agents

The critical insight from the preceding discussion of believable agents (and the reason that the topic is discussed here, rather than in the preceding chapter) is that *human social cues apply to these agent based systems, and they provide a great deal of contextual information to the user*. Observing the use of context from a human-human perspective will obviously raise issues concerned with providing context for the people concerned, and allowing this to be contextualized by the participants. Thus the findings of Reeves and Nass (1996) concerning the inability of people to distinguish between media and real-life can also be understood from the perspective of media providing social context, which affects how the media are perceived. Such social conventions have a major effect on human-human interaction and consequently on human-computer interaction, especially when the computer exhibits some human-like qualities.

In a study examining human perception of agent action, Sengers (1998) examines the transitional behaviour of existing social agents and observes that behavioural changes (between observable states of behaviour) may be viewed as exhibiting patterns which are similar to that of schizophrenics. The change between observable actions lacks coherency, and this manifests itself in a parallel fashion to a human mental disorder. Sengers poses the question "how can an agent [behaviour] appear coherent to the user" and answers "by supporting the user in constructing coherent interpretations of the agent." In terms of the model advocated by this thesis this is the same as *enabling the user to contextualize the behaviour of the agent*. Because the user uses the agents' actions and behaviour to interpret the agents' activity, the cues that the agent provides form the context in which the user places the whole interaction. By allowing the agent to provide appropriate behavioural cues for the user, the agent is *contextualizing* for the user, and this in turn facilitates the *user's* contextualization.

The other issues that concern the designers of believable agents also benefit from a re-analysis from the perspective of context and contextualization. The improvisational qualities sought by researchers such as Hayes-Roth and van Gent (1997) become the ability to make use of context in a way which is perceived by users to *situate agent behaviour within a context*. This is the central tenant of improvisational theatre, the difficulty being the dynamics of improvisation and the ability to contextualize as the context changes.

Other forms of storytelling use the term *narrative* to describe structure. The arguments about the use of narrative by agents (and in interfaces generally) pivot about leveraging the user's inherent knowledge about how these structures provide context and understanding. Narrative (in a structuralist sense) is seen as the communication of a linked set of events (Rimmon-Kenan, 1983). This linking allows both structural and temporal links to be made within a story, and this in turn allows contextualization to take place in the audience (e.g., Mateas, 1997). From a more theoretical point of view, the basis of Narrative Psychology (Bruner, 1986; 1990) is that *intentional* understanding is obtained by structuring events into stories (the alternative view being that *inanimate* understanding - the computer as tool paradigm – is understood by cause and effect and logical reasoning). In the contextual paradigm, tool and narrative are placed in a continuum in which the contextual cues are provided by both human and artefact, and vary only in their type.

The rationale for understanding intentional behaviour in this way means that systems should incorporate narrative to provide a *socially constructable context* (contextualization). Traditional narratives use character to explicate the intentions of the actors; this builds on people's social ability to recognise behavioural character traits and infer intentions - to provide social contextualization. It is less clear how interactivity fits with narrative unless the interactivity is limited in a way that enables structure to be maintained - improvisational behaviour clearly lacks any predefined structure (for it is, by definition, unplanned). Interactive Narratives (Smith and Bates, 1989) generally avoid this problem by only allowing certain interactions to change certain aspects of the route taken through a story (see Mateas, 1997); the participant is constrained within the author's intended context. For example in *Dogmatic*, Galyean (1995) presents a virtual environment that changes the events and appearance of the world to prompt a participant to take action that is meaningful to the narrative. Character is still viewed as an important element of storytelling, and the various factors surrounding compelling characters such as emotion are currently being researched (Elliott et al., 1998). Using these narrative devices, a context is provided as an overall story "shape," but events still change within this framework (and *contextualization takes place within these constraints*). It may be that the design of compelling interactive narratives will only be possible when the need to maintain contextualization is realised. This will necessitate a different approach to authoring interactive narrative in which contextualization in each possible story-thread is explicitly considered.

Sengers (1998) also suggests that a more appropriate way of thinking about an agent's behaviour is as the construction of a narrative. However, this is a reaction against the non-situated deployment of existing agents, which equates to deploying an agent without a context. From a contextual perspective, Sengers' work is not entirely compelling because her experiments are given in the context of story-telling, rather than relating interaction to concrete tasks or goals. However, this is because her main objective is to use this story-telling to address the use of transitions in behaviour to make agent intention clearer. Thus, the agents' goals are to tell a story, and to observe the effectiveness of the story-telling, with and without narrative transitions. None of the exponents of narrative interaction fully address the way that narrative can be used through interaction to accomplish some task, and to explicate agent actions by the use of narrative. This requires sophisticated agent-characters that can steer a user's behaviour in a way that maintains the desired narrative.

Nass and Reeves (1996) work essentially studies social context, and particularly the process of contextualization in people when they interact with computers. Their findings provide an insight into how it is possible to use the human predisposition to contextualize based on social cues in the design of intelligent agent interfaces. In order to do this the agents do not need to possess large amounts of knowledge, but can use heuristics to prompt the required contextualization in the user. This may give a solution to providing some means of prompting user behaviour in a way that is similar to steering a participant through an interactive narrative, although it is by no means as complex a task.

In the preceding discussion, the intention was to give a model of context and contextualization (the dynamic process of obtaining context) which provides a framework in which to situate agent design. This framework accounts for a range of agent behaviour, including social behaviour, and accounts for the findings in the literature regarding elements of successful agent design.

Having identified context and contextualization as a means of conceptually designing successful agent interfaces, the next step is to unpack the definitions of context and contextualization as they apply to interface agent design.

3.9 Providing Contextualization in an Agent Interface

It is possible to re-visit our taxonomy of interface agents based on the preceding discussion, and clarify the classifications presented in Chapter 2. In doing so it is possible to extend the taxonomy forwards (to include believable and social agents). The list becomes something like the following (See also Maes, 1997; Wilson, 1997; Bickmore et al,. 1998):

1. **Gurus** - The agent performs some reasoning (possibly prompted by the user) and supplies the results;
2. **Colleagues** - The user and the agent negotiate about the issue under consideration in a way akin to asking a colleague for an opinion (Wilson, 1997);
3. **Assistants** - "Secretary Agents" (Mase, 1997) or Guides; These agents allow delegation of tasks from the user, necessarily including elements of Guru and Colleague;
4. **Animated Autonomous Personal Representatives** - Agents that "stand in" for the user. They may express a user's point of view, give guided tours, presentations, or represent the user's opinions (including elements of personality). Avatars (virtual physical representations) are included in this category;
5. **Companions** - Agents which exist as believable social entities;
6. **Entertainers** - Agents which entertain (e.g., "Actors" in virtual theatre).

It is interesting to compare the level of agent autonomy in the taxonomy above with the levels of automation discussed in Section 1.4.4. Although the granularity of the two taxonomies is slightly different, the level at which the agent (or computer) can take the initiative in taking action increases from top to bottom. The role of the agent indicates the level at which it is capable of taking the initiative, a contextual cue that is missing from conventional implementations of automation. Both classifications implicitly address interaction complexity and interaction with an artefact that can affect these complexity issues on behalf of the user. The successful management of contextualization (role in the agent case) will consequently provide for ease of use by reducing complexity.

Because this thesis is concerned with reducing complexity, elements of agent implementation associated with entertainment (theatre, improvisation, entertainment) are not directly of interest in the implementation of agent interfaces. However, these issues may peripherally affect the design of agents from an interactive standpoint, as they provide social cues for the user. Focussing on the Assistant role of an agent, it is now possible to identify a series of factors that address contextualization and interaction in such agents.

3.10 A Model of Contextualization

The agent-interaction contextualization paradigm is reflexive – both user and agent are viewed as actively constructing context as a process of contextualization. When designing an interface (agent-based or otherwise) the design implication is that the interface and interaction should provide the user with cues to aid their contextualization, making use of appropriate task and domain factors and relevant border resources. These border resources may be more generic than the factors usually taken into account as "user preferences" and may include generic user factors (such as social context, etc.) that influence contextualization.

In Table 3.1 the components required for an agent to contextualize were identified and compared with the knowledge required to implement an intelligent or adaptive user interface. In discussing contextualization as a process, it is clear that these components interact in order to provide contextualization. These components and the interactions between them are shown in Figure 3.3.

A User Model component encompasses elements of user behaviour and user preferences that influence contextualization. These factors constrain interactions with the task set, as specific user characteristics will influence the traversal of a task hierarchy based on known user preferences within tasks. For example, if the user model contains preferences that relate to specific tasks, no dialogue is required to elicit these from the user. In this way both the User Model and the Task Set influence the way in which dialogue is instigated (by user or system) and the role which the system (as an agent) may adopt.

User preferences from the User Model may directly affect dialogue instigation and discourse structure by specifying a preferred dialogue style or agent role, and the selected Task Set may further influence this (e.g., by pruning the tasks and subsequent dialogues). The dialogue model may, in turn, affect the traversal of the task hierarchy based on the form of dialogue style selected.

The form of dialogue and the tasks possible in the domain finally impact the way in which the dialogue is rendered. Preferences combined within dialogue models may select a particular modality (speech, text, graphics, etc.) or may constrain the rendering in a particular modality in order to conform to particular user expectations (e.g., a particular look and feel, dialogue style, etc.).

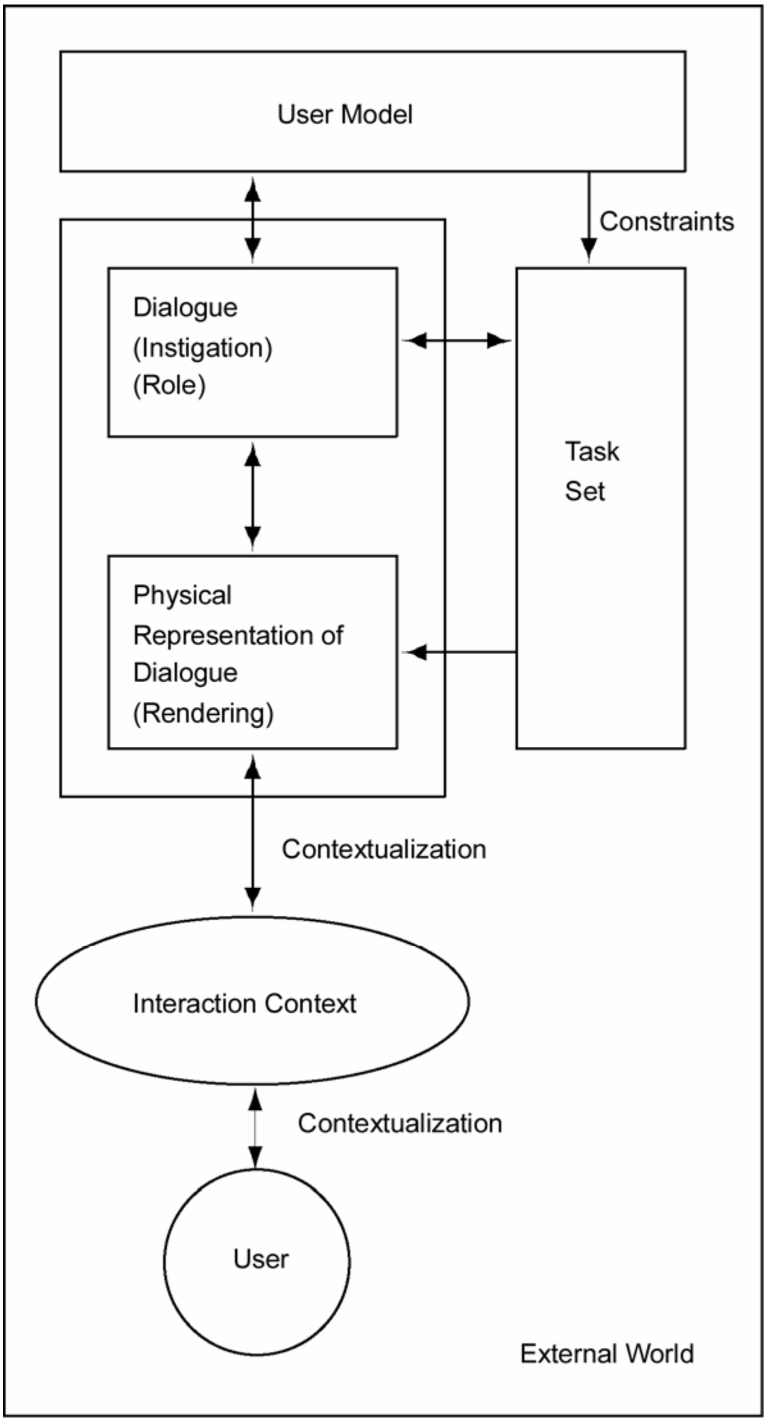


Figure 3.3 A Model of Contextualization

In combination these components interact dynamically to produce an interaction context. The interactions between the components can be viewed as the process of contextualization, as this process of constructing the form of interaction provides a way of managing the cues which allow the user to contextualize the system's behaviour. By monitoring the user's interaction, the system may, in turn, contextualize the response of the user and generate a new context which provides enough common context for the user to contextualize and respond appropriately. This allows the system to maintain a shared context with the user that disambiguates the interaction (Section 3.3).

At first appearances this model has some properties in common with conventional adaptive, intelligent and agent-based interfaces. However, the significant advantage of developing this model from the perspective of contextualization is that an appreciation of the scope of contextual cues (border resources including behavioural and social cues, etc.) allows appropriately rich models to be incorporated in the components and into the contextualization process. When considering preferences it becomes possible to take into account both invariant task factors and also user specific behavioural and social factors. In this way the process of contextualization may be tailored to the user and the task. Appropriate interaction models may be chosen, forms of dialogue instigation and control specified, and rendering of the interface can be designed to be consistent with these choices. For example, the role that the interface plays may be specified (Section 3.9), the external domain factors that influence the interaction may be identified, including those particular to the user in relation to the domain.

The advantage of having such a model is that it compels a designer to consider both the process of contextualization, and the components of context which are important in the domain and tasks under consideration. In addition, it highlights the importance of exploring contextualization factors which lie on the border of what would normally be considered, and incorporates all of these into a single model of interaction using either a conventional or an agent-based design paradigm.

3.11 Border Resources in Contextualization

In previously examining the effectiveness of believable agents (Sections 3.7 and 3.7.1) one of the issues identified as affecting user response in a range of agent implementations is personality. This is reinforced by studies of human response to computers and media from a social perspective (e.g., Reeves and Nass, 1996). In considering contextualization as a process which resides in both the user and the agent system, elements of personality become a resource not only for the user, but also for the agent system. Consequently personality may be viewed as an element of the User Model of contextualization, as elements of the user personality will influence the way in which the user responds to the agent (and the apparent personality of the agent). This in turn propagates through the model and will influence elements of dialogue instigation and dialogue rendering.

There is an obvious link between personality traits and user preferences - both being indications of default tendencies in behaviour. Preferences are in important factor in tailoring system behaviours to individual users or groups of users, so it would seem likely that personality traits will also provide for a level of system tailorability through the model of contextualization. In effect, Personality becomes a border resource in considering preferences within the User Model – it is a social cue in interactivity.

3.11.1 Personality Models

It is generally understood that individual differences influence interaction. Personality is the branch of psychology which is concerned with providing a systematic account of the ways in which individuals differ from one-another. Personality studies intend to provide a systematic account of individual tendencies (proclivities, propensities, dispositions, inclinations, etc.) to act or not in certain ways on certain occasions. Allport (1961) describes personality in terms of traits, an evolution of Jung's approach to personality (e.g., see Ewen, 1998). These personal traits (or personal dispositions) determine each person's unique style of behaviour. Allport estimates there may be some 4,000-5,000 traits (and 18,00 trait names) and makes the distinction between dynamic traits (which determine why people do things), temperament traits (how people do things – their style) and ability traits (how well people do things).

The models of personality which evolved from this research consist of sets of traits which catalogue behavioural responses. Most research reduces the number of characteristics to a limited set of dimensions using cluster and factor analysis techniques (e.g., Cattell and Kline (1976) subjected Allport's list of traits to factor analysis and identified 16 personality factors). The Big Five Model (e.g., see Moon and Nass, 1996) proposes the following factors:

1. Extraversion (dominance or submissiveness);
2. Agreeableness;
3. Consistency (dependability);
4. Emotional stability;
5. Culture (intellect/openness to experience).

Many models of personality address interpersonal interaction. The principle of similarity attraction suggests that people prefer to interact with other people who have similar personalities. Thus for each factor in a taxonomy, there is a bipolar scale with opposing characteristics at each end. The Myers-Briggs Type Indicator (e.g., see Murray, 1990) is intended to be an inventory of basic preferences (rather than a measure of traits) and is based on a four-factor model of two-dimensional scales:

1. Extraversion-Introversion;
2. Sensation-Intuition;
3. Thinking-Feeling;
4. Judgement-Perception

Other researches (e.g., Wiggins, 1979; Kiesler, 1983) propose taxonomies with eight or more bipolar axes. These scales cover a wide range of aspects of personality, all of which relate to behavioural traits.

It is not clear what effect some of these traits will have on interaction, only that there may be a connection between some personality factors and the way in which system or agent behaviour is perceived. However, in the case of Extraversion versus Introversion (or dominant-submissive) we may expect that the form of dialogue instigation may interact with this trait. For example we may expect extraverted/dominant people to prefer to instigate dialogue themselves, whilst introverted/submissive people may respond more favourably when the agent or system takes the initiative in dialogue instigation.

3.11.2 Other Border Resources

The model of contextualization presented here, and the concept of border resources that are made use of in contextualization, allow other aspects of interface design to be placed within this framework. The notorious difficulties of situating issues such as aesthetics (and other art and design based elements of user interface design) within a design framework can be addressed by treating these as border resources for contextualization. Although it is often said that "interface design is as much as an art as a science", viewing these issues as an interaction between user preferences and the physical representation of dialogue allows a designer to address these issues as border resources using the model of contextualization. Although not a prescriptive recipe for design, this framework makes explicit the issues that are raised in this part of the design process and their impact on interaction.

3.12 Agent Characteristics for Contextualization

Addressing agents that primarily perform an assistant role with the objective of reducing the user-perceived complexity of a task, it is now possible to identify which elements can be drawn from contextualization theory and implemented in agent interfaces. From the preceding analysis it is possible to identify several factors which promote contextualization in agent-user exchanges. These factors are directly related to the elements of context and contextualization described in section 3.6.

1. The Set of Tasks: Task Models

Tasks structures conceptually provide context to the user and facilitate the mapping of user goals to appropriate outcomes. The user needs to know what tasks are possible in a given context. The more coherent the tasks, the more constrained the context (and vice-versa). Consequently, the agent should utilise task models to provide coherence and the appropriate cues in interacting with the user.

2. A Model of the User: User Preferences

The ability of a system to act appropriately in a given context (*adaptation*) requires that the agent have both a model of the tasks, and also knowledge about how the user is likely to behave in carrying out those tasks. Therefore the agent should make use of *user preferences* in order to better contextualize.

3. The Form of Dialogue: Delegation

The style of task allocation from user to agent should be one of delegation. This delegation should take place using an asymmetric dialogue. An asymmetric dialogue using delegation maintains a coherent level of autonomy and allows allocation of tasks in a clearly contextualized manner.

4. The Representation of Dialogue: Interface Adaptation

The agent should be able to adapt the interface in order to prompt the appropriate contextualization in the user. This will enable the adaptation of the interface for a given usage context.

5. The External World: Social Cues

Social cues affect a range of agent behaviour. For example, the level of autonomy shown by an agent is, to varying extents, a factor of social context, as this relates a dominance factor in terms of personality traits. This and other findings show that the use of social cues will enable the agent to provide better contextualization.

3.13 Summary

This chapter has examined the concept of context, particularly in relation to theories of human computer interaction. *Contextualization* has been developed as a theory of the influence that context has on interaction, with the rationale that *enabling* contextualization *reduces* complexity. The process of contextualization captures the active process of interpreting a dynamic environment and deciding what action to take. The elements of context and contextualization are very similar to those identified in adaptive and intelligent interface technologies. A good way of viewing intelligent and adaptive interfaces is as interfaces that adapt for a given context, or provide contextualization. These commonalties can be viewed as evidence that intelligent and adaptive user interfaces are actually user interfaces that are intended to contextualize, but this is not realised by the designers. Indeed, the lack of such a coherent interface model, and the absence of some of the functionality required for contextualization can be used to explain the failure of some intelligent and adaptive user interfaces.

In examining elements of context and contextualization, particularly in regard to interface agents, it becomes possible to identify a taxonomy of interface agents (Section 3.6) and from this draw up a list of desirable functionalities that interface agent should possess in order to provide good contextualization

(Section 3.12). In the next chapter these characteristics will be evaluated empirically to verify their utility.

Chapter 4

Experimental Investigation

4.0 Effects of Improved Contextualization

The previous chapter developed a model of contextualization, and from this identified a number of contextualization components that contribute to improved agent-user interaction. In this chapter, the effectiveness of these components will be investigated empirically.

Each factor has been identified as facilitating contextualization, consequently improving user interaction with the agent. In evaluating these factors it is necessary to specify what the observable characteristics of this improved interaction will be in relation to contextualization.

The form that improved interaction may take is defined by the interaction that is observable between user and agent. *Context* is the environment in which interaction takes place. *Contextualization* is the process that allows these contextual factors to be made use of in interacting with a user. Consequently, the better the contextualization process, the more contextual elements that can be made use of, resulting in fewer elements that must be made explicit in the interaction. As contextualization improves it will therefore be indicated by:

1. Fewer dialogue exchanges to accomplish a given task;
2. Fewer broken dialogues (dialogue errors) in accomplishing a given task;
3. Less time to complete a task;
4. Greater user satisfaction.

4.1 Contextualization Components

From the previous examination of components that positively contribute to contextualization, several components were identified as important in the design of an agent interface which will have the capability to contextualize:

1. **Task Models**: A clearly represented and shared task model will improve interaction.
2. **User Preferences**: An agent that maintains a model of the user (and uses it to adapt in context) will be more effective than one that does not.
3. **Social Cues**: The provision of social cues will improve user-agent interaction.
4. **Delegation**: The communication between user and agent will be more effective when the agent maintains an asymmetric dialogue with the user, in which the user takes the initiative in interactions.
5. **Interface Adaptation**: An interface which adapts in context will be better than one which does not.

In order to evaluate these predictions empirically, experiments were designed to compare user-system interaction with various conditions. Not all components are explicitly taken into account because of design requirements and interdependencies between components. Implicit in the design of the system is that there must be a consistent task model for all interactions (note this is a recognised component of all IAUIs). This is achieved by choosing an experimental domain having a task structure with which users are familiar, and which is further reinforced by instruction. In terms of Interface Adaptation and User Preferences, the experimental trial uses adaptation triggered by specified user preferences. These components are therefore combined into one condition. In a similar way, Interface Adaptation and User Preferences are closely bound together because user preferences are one of the components that can instigate interface adaptation (together with a task model). The three experiments become:

Experiment 1: Agent Dialogue Autonomy,

Experiment 2: Interface Adaptation with User Preferences,

Experiment 3: Social Cues (Personality) in Agent Interaction.

4.2 Social Cues in Agent Interaction.

Although only Experiment 3 explicitly relates to social cues, the overall design of the experimental series requires that assessment methods for social cues be taken into account at the outset. The prediction that contextualization components may be influenced by social factors related to user characteristics allows the hypothesis that the user's personality may influence the way in which they respond to dialogue instigation. To investigate any effect that this may have, all participants were given a questionnaire to assess personality factors that may influence performance and preference relating to dialogue styles. Questionnaires were administered at the start of each experimental trial in order to balance personality types within conditions.

4.3 Experiment 1: Agent Dialogue Autonomy

The factors of Delegation and Interface Adaptation may be used to describe the degree of autonomy an agent has in controlling dialogue with the user, or the degree of initiative that the agent may take in instigating dialogue. Delegation implies an asymmetric dialogue with the user acting as the delegator. This style of dialogue will provide consistent contextualization for the user due to the reinforcement of role and the predictability of agent behaviour (partly because this form of dialogue has an attendant set of social cues in terms of representing the respective authority and responsibility between the dialogue partners).

It is important to note that dialogue style (in these terms) is one of the interface characteristics that can be manipulated by an Intelligent or Adaptive User Interface (IAUI). An IAUI may have the capability to change dialogue styles dynamically during interaction, allowing delegative dialogues in both directions. An IAUI may also support mixed initiative dialogues in which either dialogue partner can take the initiative (i.e., instigate dialogue). This obviously conflicts with maintaining a delegation-style of dialogue, but it can be compared to a human-human environment in which dialogue control may be based on social factors. For example in some circumstances the dialogue is purely asymmetric (e.g., A Commanding Officer giving orders to a Soldier), in others there may be some degree of adaptation by the "agent" (e.g., a Nurse in an operating theatre suggesting something to the Surgeon). In other environments there is little asymmetry and the two participants have a collaborative dialogue (e.g., two people co-operating in a repair task).

The model of contextualization suggests that a delegatory dialogue style from user to agent will provide better performance than other models because of the consistency and control which this gives the user. Several instances of IAUIs produce deterioration in performance because they provide inconsistent dialogue patterns (Hockley, 1986), and there is little empirical evidence that models of mixed initiative dialogue can follow human communicative behaviour closely enough not to confuse the user (Rich and Sidner 1998).

There are two levels of dialogue control which are consequently of interest:

1. User-Driven. This case concerns a system in which the interaction is driven solely by the user – the user instigates command actions by always taking the initiative and delegating to the software.

2. Agent-Led. This case implements an interrogative dialogue by always instigating dialogue with the user.

4.3.1 Methodology

In order to investigate the effects of dialogue style on interaction an experiment was carried out in which the dialogue could be controlled in a representative domain. Travel Planning was chosen in order to provide a familiar task domain which has obvious parallels with agent interaction (much travel planning is carried out with the assistance of a human travel agent). This provides a familiar environment for experimental participants, whilst at the same time allowing an appropriate range of conditions to be evaluated. In addition the style of interaction of a person and a travel agent can be observed to encompass all the possible dialogue styles. The user is presented with the flight planning scenario described below and must use the provided experimental software to schedule a journey.

4.3.2 Scenario Description

The aircraft flight planning scenario given to subjects is as follows (See Appendix A5):

> You are an Ottawa-based sales representative planning a round-trip from Ottawa to New York, then to Boston, Toronto and back to Ottawa. Today is Monday, October 25. You are due in New York on Tuesday (October 26) to visit a client who is available between 1pm and 3pm. On Wednesday (October 27) you are visiting a client in Boston who is available from 10am to 2pm. On Thursday (October 28) you are due in Toronto to visit a client at 3pm. You fly back to Ottawa on Friday. You would like to avoid having more than one flight each day. Your preferred airline is Air Canada.

The objective is therefore to schedule the required journey using the software tools provided.

4.3.3 Experimental Conditions

There is 1 independent variable (dialogue style instigation) that will be manipulated, with 2 conditions. These are:

1. User-Driven,
2. Agent-Led,

The experiment therefore consists of 1 factor having 2 conditions giving a 1 x 2 factorial design.

4.3.4 Hypotheses

H1 A user-driven (delagative) dialogue will be more effective than agent-led dialogues.

More specifically, a User-driven dialogue will

1. have fewer dialogue exchanges;
2. have fewer broken dialogues;
3. be faster;
4. provide greater user satisfaction.

The null hypothesis is therefore:

H0 There is no performance advantage provided by any particular dialogue style.

4.3.5 Software Design

Software to implement the interface was written in Microsoft Visual Basic 6. The dialogues take the form of a series of dialogue boxes that are produced from the task model shown in Figure 4.1. Note that the task model is constant, but the way in which the dialogue relating to the tasks is instigated varies whilst the appearance of the interface remains consistent throughout all conditions. A series of example dialogues from Condition 1 are given in Appendix A6. The software allows flight details to be specified and then searches a database of matching flights for possible matches. A matching flight may then be selected and placed into the overall itinerary. The software uses real flight data taken from an on-line reservation system[3].

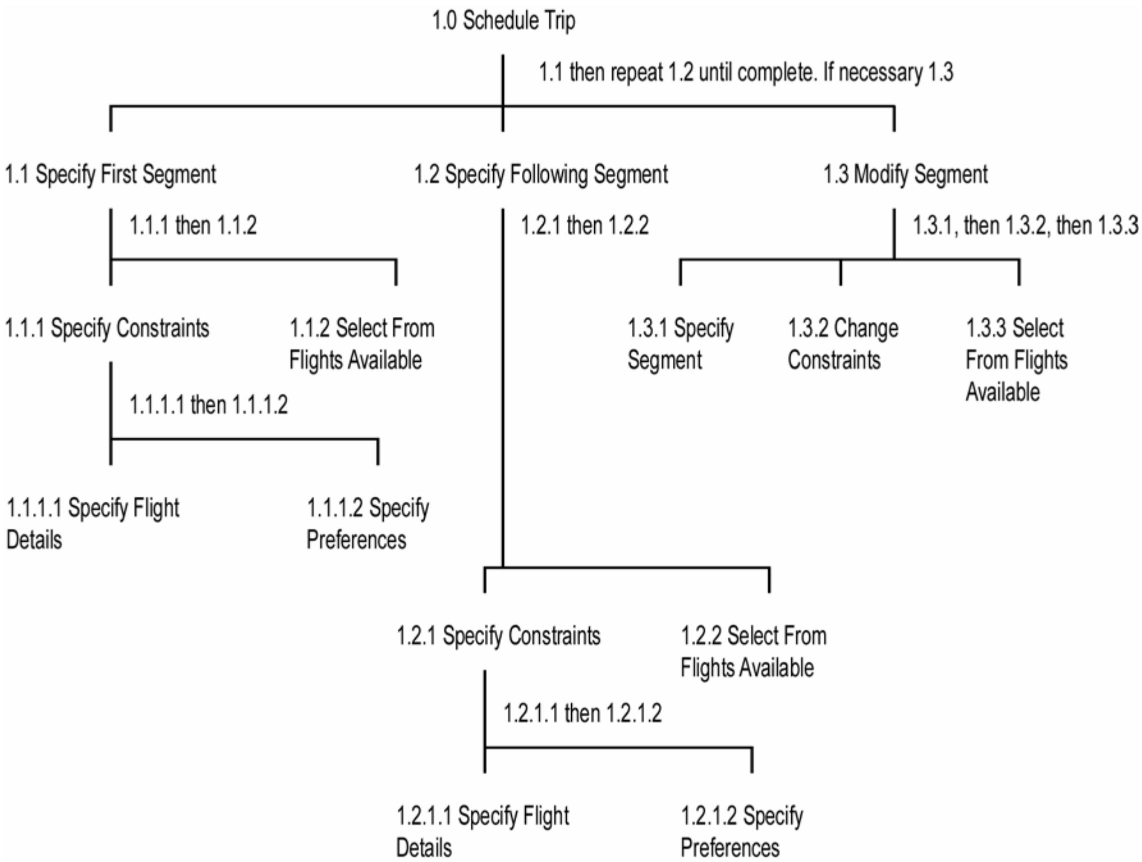


Figure 4.1. Task Analysis of the Experimental Domain

Each condition affects the presentation order of the possible dialogues. In the User-Driven condition (Condition 1) the software behaves as a traditional piece of desktop software – nothing happens at the interface unless the user takes some action (presses a button, clicks on a list, etc). This case operates as the equivalent of a delegatory dialogue, as the user configures a set of choices which are then acted upon by a flight-search agent. In the Agent-Led condition (Condition 2), the system automatically generates dialogue boxes according to the task model. This interface also has the ability to aid the user in constraining the number of possible solutions. Note that the same forms are used in each

[3] http://www.travelocity.com/

version of the software, but it is the way in which they are presented to the user that varies. The Agent-Led condition is implemented by hard-coding "intelligent" responses into the software, using known contextual constraints (i.e., the task is specified, so the kinds of behaviour the system can exhibit are limited). In this way, for the agent-led condition, the system will automatically generate a form to instigate a dialogue when specific data must be entered, prompting the user to take action.

4.3.6 Data Collection

20 subjects participated in the experiment, 10 per condition. Subjects were drawn from Canadian Governmental Research Institutes and Universities. Subjects were selected to have a general knowledge of computer operation but no other filtering process was used. Subjects were randomly assigned to one of the two conditions, balanced by primary personality trait (see Experiment 3).

The following data were collected:

Objective:	Number of Dialogue Exchanges	(Frequency)
	Number of broken dialogues	(Frequency)
	Time Taken	(Quantitative)

These data were logged by software, see Appendix B1 for example data file

Subjective:	User Satisfaction Questionnaire	(Categorical)

This data was obtained after each software trial. The questionnaire used is given in Appendix A5. The complete set of results are given in Appendix B2.

4.3.6.1 Performance Results

The performance results are given in Table 4.1.

Condition	**n**	**No. Dialogues**	**Broken Dialogues**	**Time**	**Log Time**

		Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	**10**	**43.6**	**16.249**	**2.6**	**2.412**	**509.80**	**146.23**	**6.1935**	**0.3082**
2	**10**	**44.5**	**17.551**	**4.1**	**6.871**	**704.90**	**238.52**	**6.5069**	**0.3380**

Table 4.1. Summary of performance results

The results were analysed using the *Statistica* statistical analysis program, and the means of each variable were compared using analysis of variance. The results are shown in Table 4.2.

Variable	Mean Square Effect	Mean Square Error	F(df1,2) 1,18	p-level
Dialogues	**4.1**	**286.05**	**0.014158**	**0.9066020**
Broken Dialogues	**11.3**	**26.52**	**0.424261**	**0.5230480**
Time	**190320.0**	**39138.25**	**4.862763**	**0.0406873**
Log Time	**0.5**	**0.10**	**4.694015**	**0.0439326**

Table 4.2. Analysis of performance data

As can be seen, there was no significant difference between conditions for either the number of dialogues or broken dialogues. However, there was a significant difference ($p<0.05$) in the overall time taken for the interaction. Overall the data are slightly positively skewed (skewness=0.861, see Figure 4.2) and to ensure this does not interfere with significance, the natural logarithm of time was also taken. Again logs of time show a significant difference ($p<0.05$) in time taken for interaction between conditions.

Although both the mean number of dialogues and the mean number of broken dialogues are higher in Condition 1 than in Condition 2, the difference in mean values was not significant.

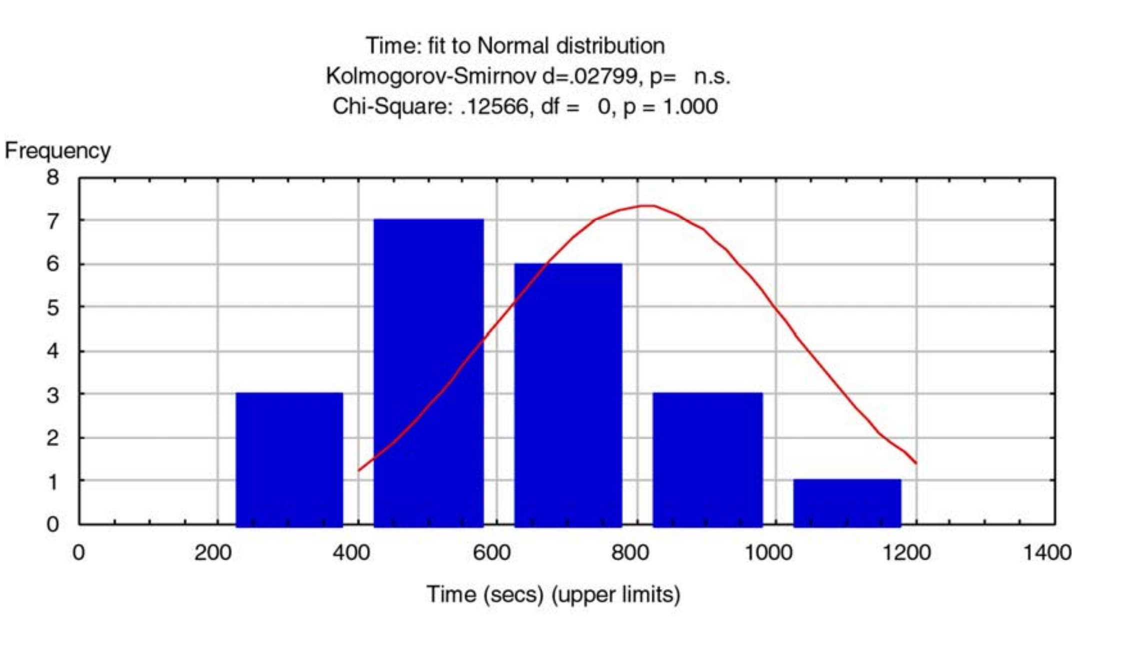


Figure 4.2. Distribution of total time data for both trials

The lack of significant difference between the number of dialogues and dialogue errors may be due to the use of a common task model and interface implementation. In terms of contextualization, this suggests that to some extent, the use of a clear task model can reduce the effects of dialogue implementation.

4.3.6.2 User Satisfaction Questionnaire Results

The user satisfaction questionnaire is based on standard usability questionnaires for obtaining subjective user feedback on the quality of the user interface and the interaction that takes place (e.g., Shneiderman, 1987; Chin et al., 1988; Lewis, 1995; Lin et al., 1997). The questionnaire consists of 20 statements to which the user can indicate their level of agreement on a 7-point Likert scale (See Appendix A5). The results from the questionnaire data are shown in Table 4.3 and graphically in Figures 4.3-4.22. The data from the questionnaires was reliable (Condition1: Cronbach's *alpha*=0.88; Condition 2: Cronbach's *alpha*=0.82).

	Condition 1				Condition 2			
	Mean	**Median**	**Mode**	**Freq. Mode**	**Mean**	**Median**	**Mode**	**Freq. Mode**
Q1	**3.2**	**3**	**m**	**-**	**3.8**	**4**	**m**	**-**
Q2	**3.3**	**3**	**5**	**3**	**4.2**	**4**	**4**	**5**
Q3	**3.6**	**3**	**2**	**3**	**4.7**	**5**	**5**	**6**
Q4	**4.9**	**5**	**5**	**3**	**5.6**	**6**	**6**	**5**
Q5	**2.9**	**3**	**3**	**4**	**3.4**	**3**	**3**	**4**
Q6	**3.9**	**4**	**6**	**3**	**3.9**	**3.5**	**m**	**-**
Q7	**4.1**	**4**	**m**	**-**	**4.6**	**5**	**6**	**4**
Q8	**4.7**	**5**	**6**	**3**	**5**	**5**	**m**	**-**
Q9	**3.8**	**3.5**	**3**	**4**	**3.8**	**4**	**m**	**-**
Q10	**3.7**	**4**	**4**	**3**	**3.7**	**3**	**3**	**4**
Q11	**3.3**	**3**	**3**	**5**	**3**	**3**	**3**	**4**
Q12	**2.3**	**2**	**2**	**5**	**2.5**	**2.5**	**m**	**-**
Q13	**4.3**	**5**	**5**	**5**	**4.9**	**5**	**6**	**3**
Q14	**2.9**	**2.5**	**2**	**3**	**3.8**	**4**	**4**	**4**
Q15	**1.8**	**2**	**m**	**-**	**1.9**	**2**	**2**	**5**
Q16	**5**	**5.5**	**7**	**3**	**3.3**	**3**	**m**	**-**
Q17	**3.2**	**2**	**2**	**5**	**2.7**	**2.5**	**m**	**-**
Q18	**4.4**	**4.5**	**6**	**5**	**3.8**	**3**	**3**	**4**
Q19	**3.8**	**3.5**	**m**	**-**	**3.9**	**4**	**4**	**3**
Q20	**3.6**	**3.5**	**2**	**3**	**4.9**	**5.5**	**6**	**4**

Table 4.3. Results form User Satisfaction Questionnaire (m=multiple)

When using a Likert scale, although the data are ordinal, the scale may not be linear (e.g., the difference between a score of 6 and a score of 7 may not be the same as the difference between a score of 2 and a score of 3). Consequently the data are best described by median, mode and interquartile range. Box and whisker plots are used to show the median point (circle), the 25% - 75% range, and the maximum and minimum responses. (including 50% of the data about the median).

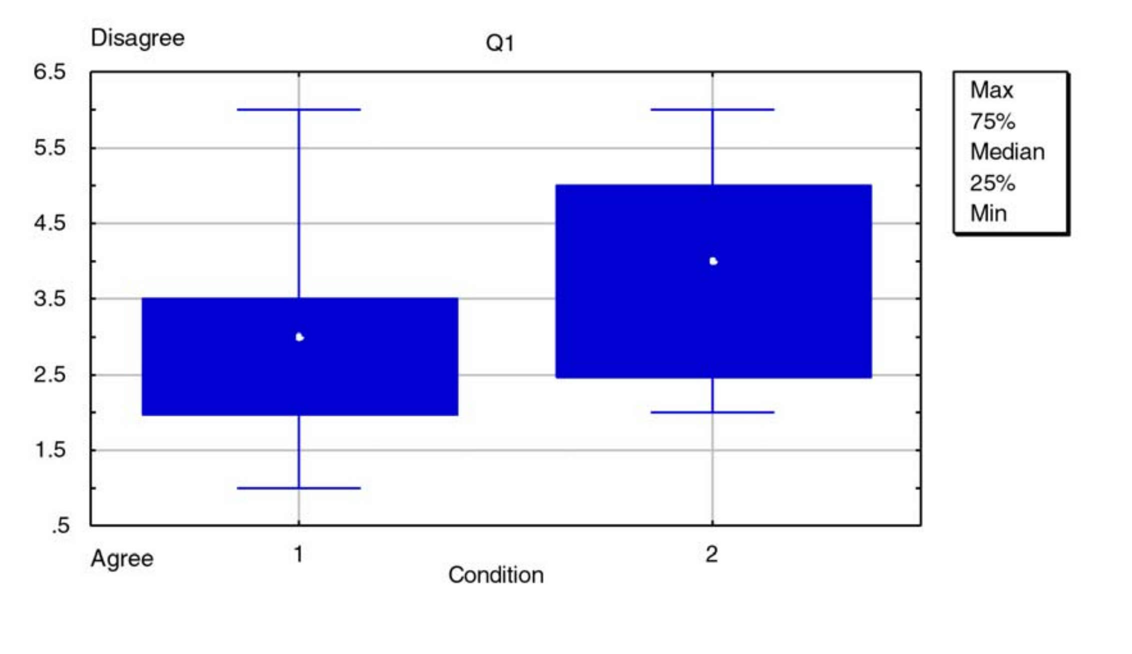


Figure 4.3. Results for Q1

Q1. *It was simple to use this software*

Condition 1 simpler than Condition 2.

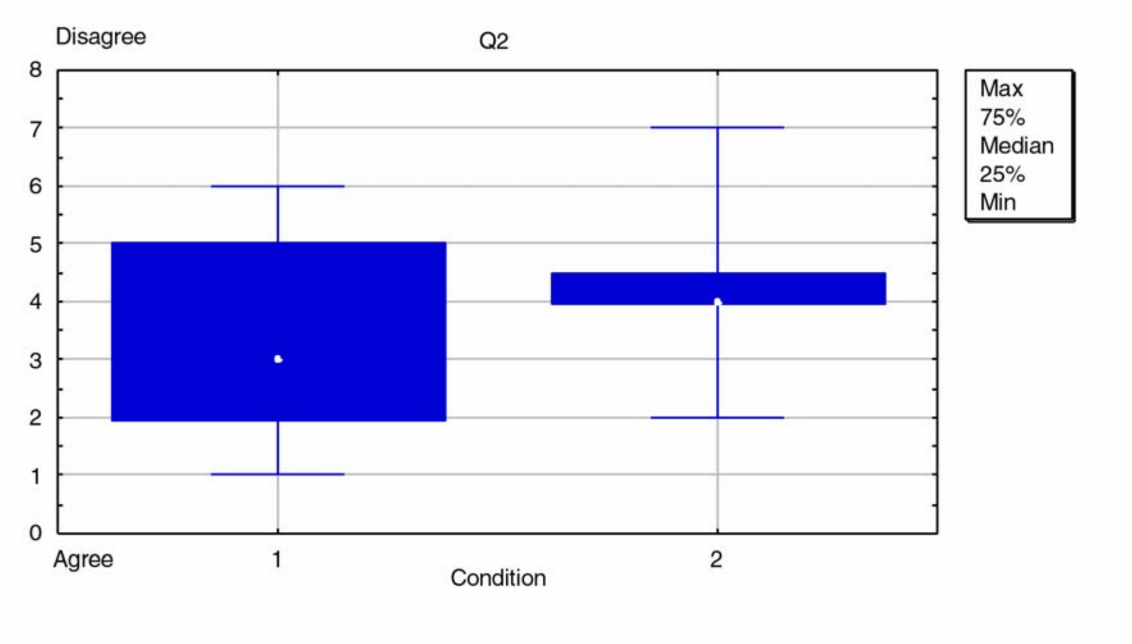


Figure 4.4. Results for Q2

Q2. *I felt comfortable using the software*

Less comfortable in Condition 2 (Agent led)

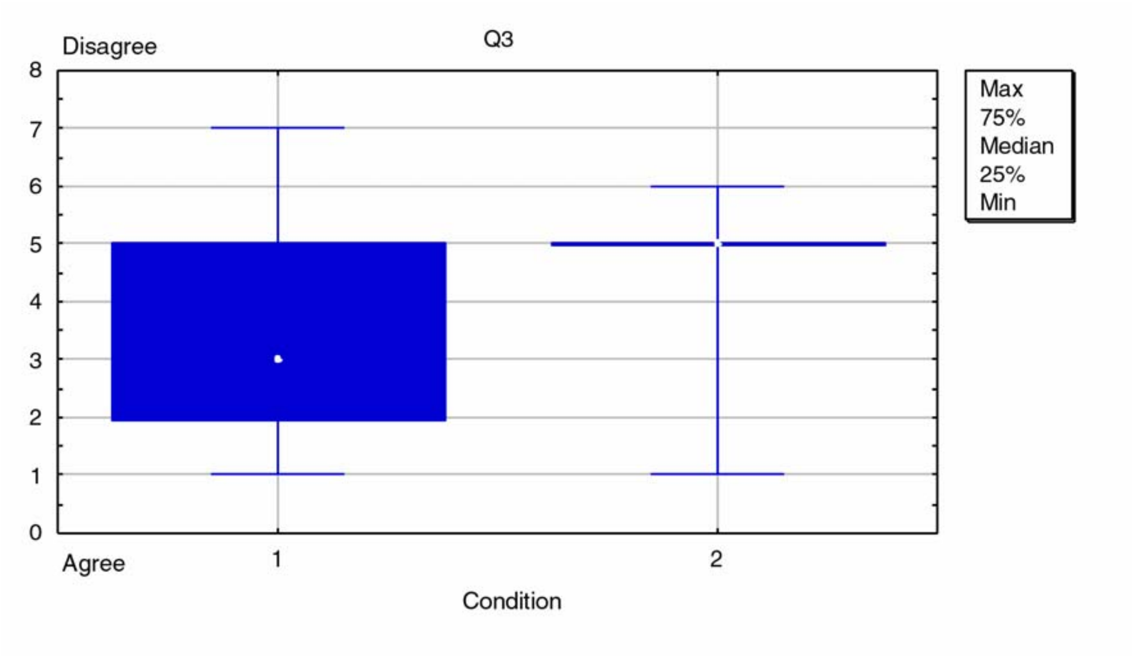


Figure 4.5. Results for Q3

Q3. *Information provided by the software was easy to understand*

Harder to understand in Condition 2 (Agent led)

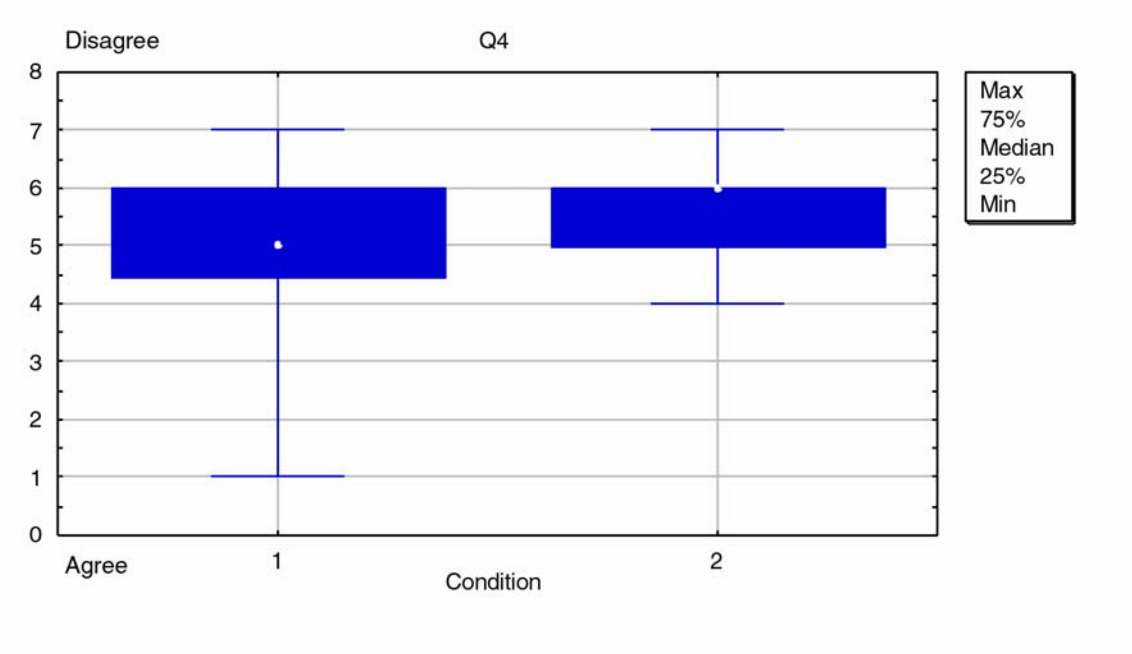


Figure 4.6. Results for Q4

Q4. *The interface to the software was pleasant*

Slightly less pleasant in Condition 2 (Median and Mode values)

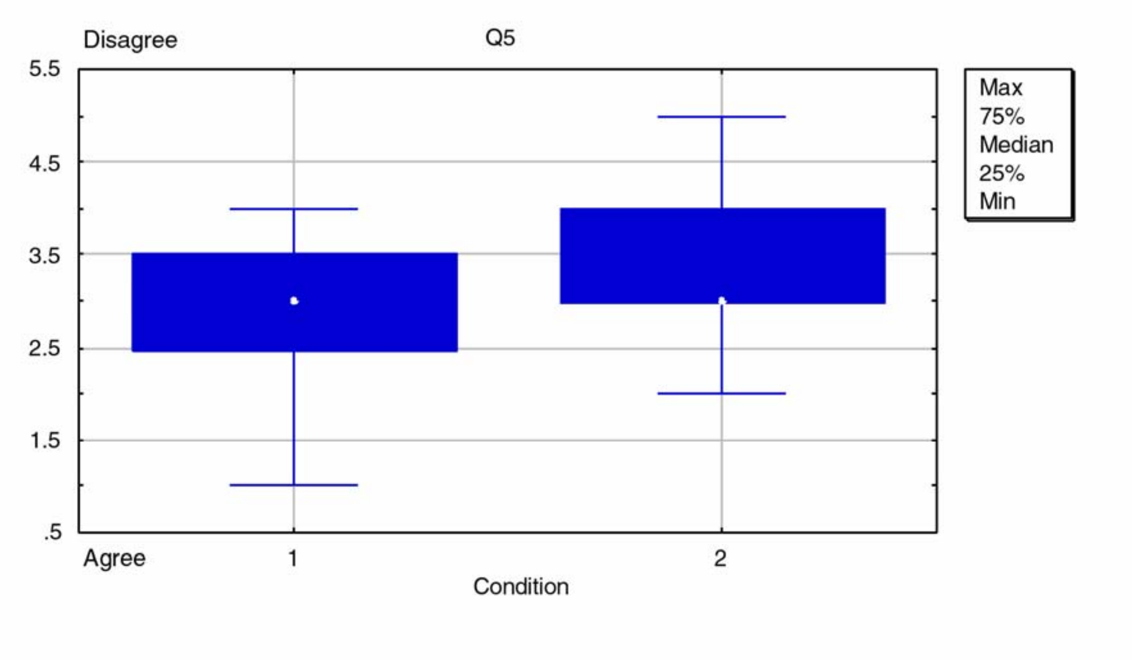


Figure 4.7. Results for Q5

Q5. *I could anticipate the behaviour of the software*

Little difference between conditions (Median, Mode the same)

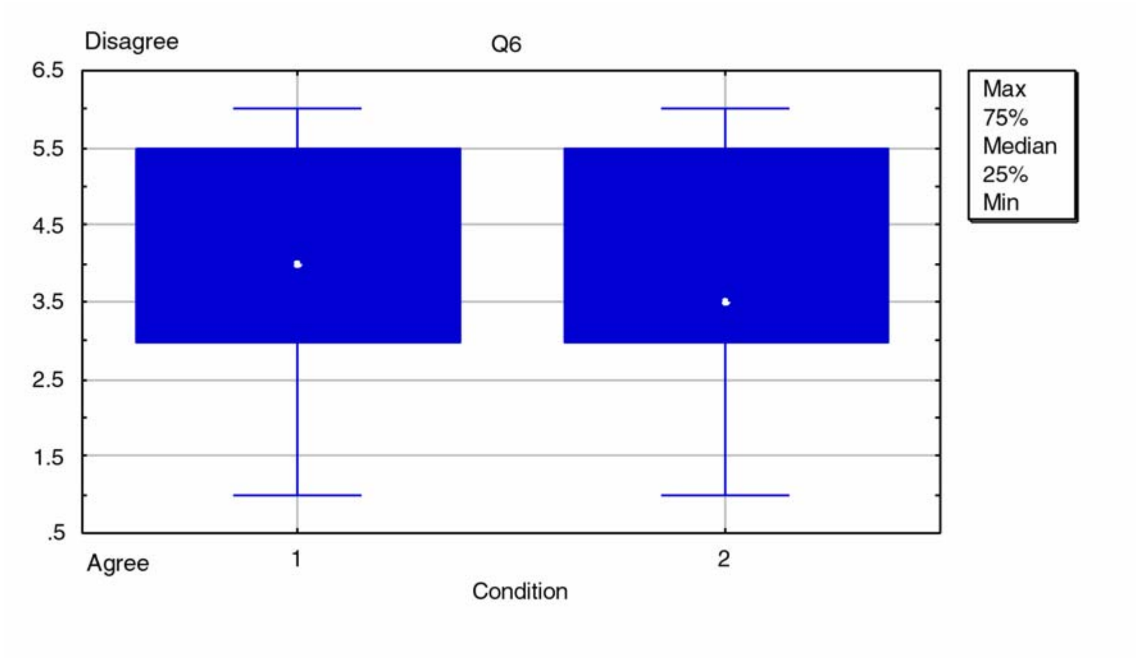


Figure 4.8. Results for Q6

Q6. *Prompts for me to take action were clear*

No difference between conditions.

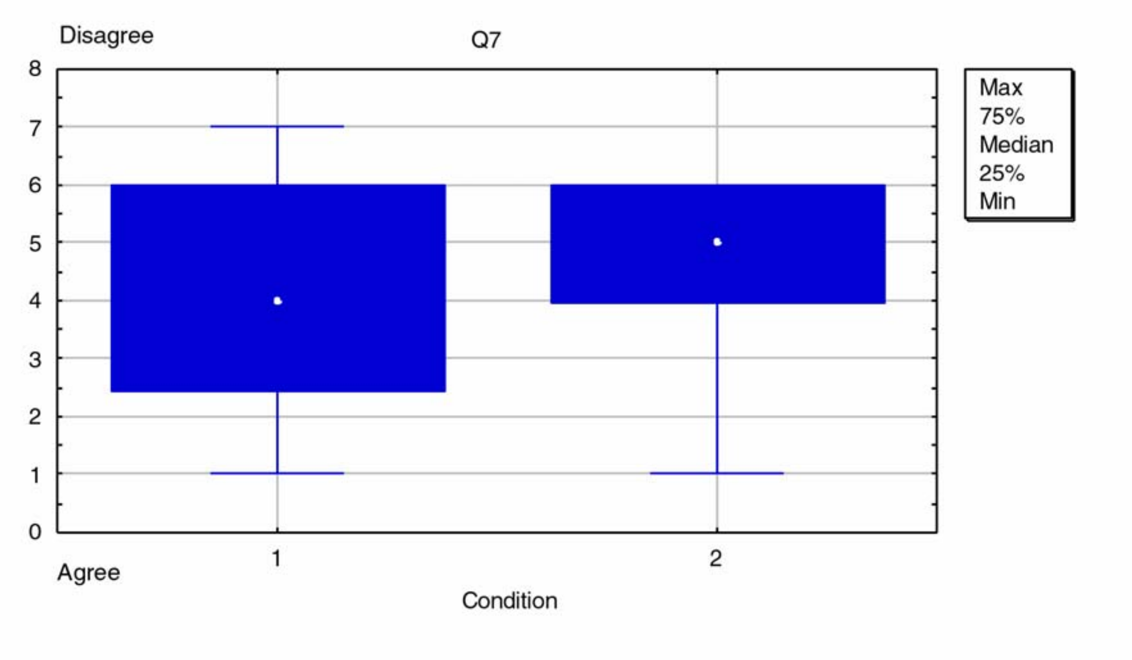


Figure 4.9. Results for Q7

Q7. *The software informed me about its progress*

Less informed in condition 2 (Agent led)

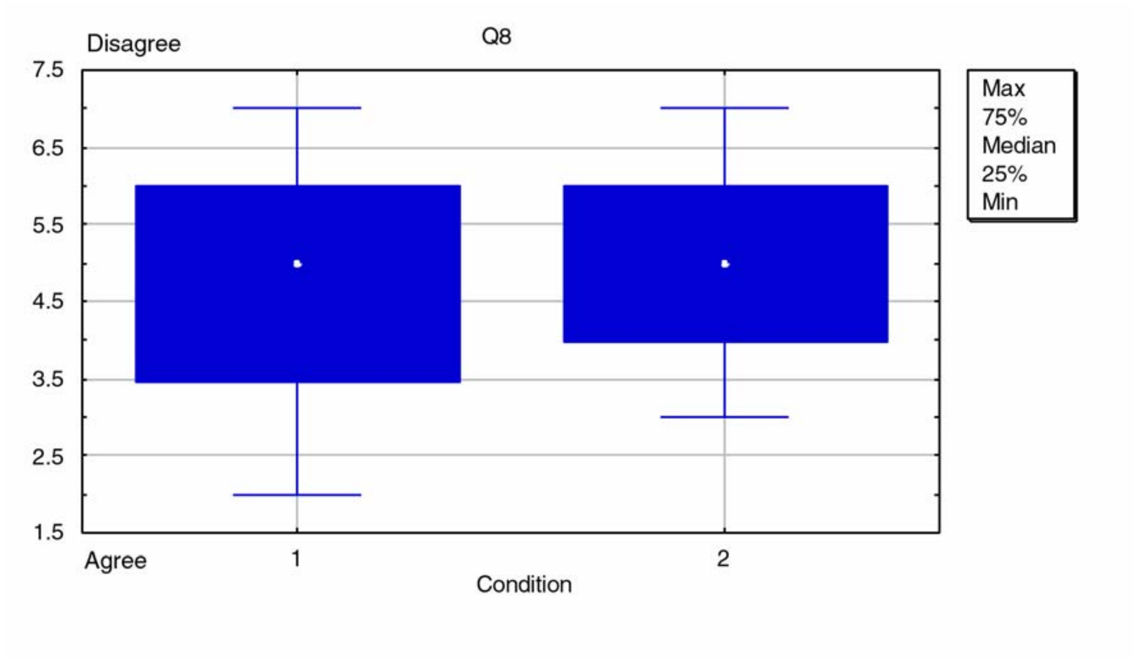


Figure 4.10. Results for Q8

Q8. *I enjoyed using the software*

Little difference between conditions

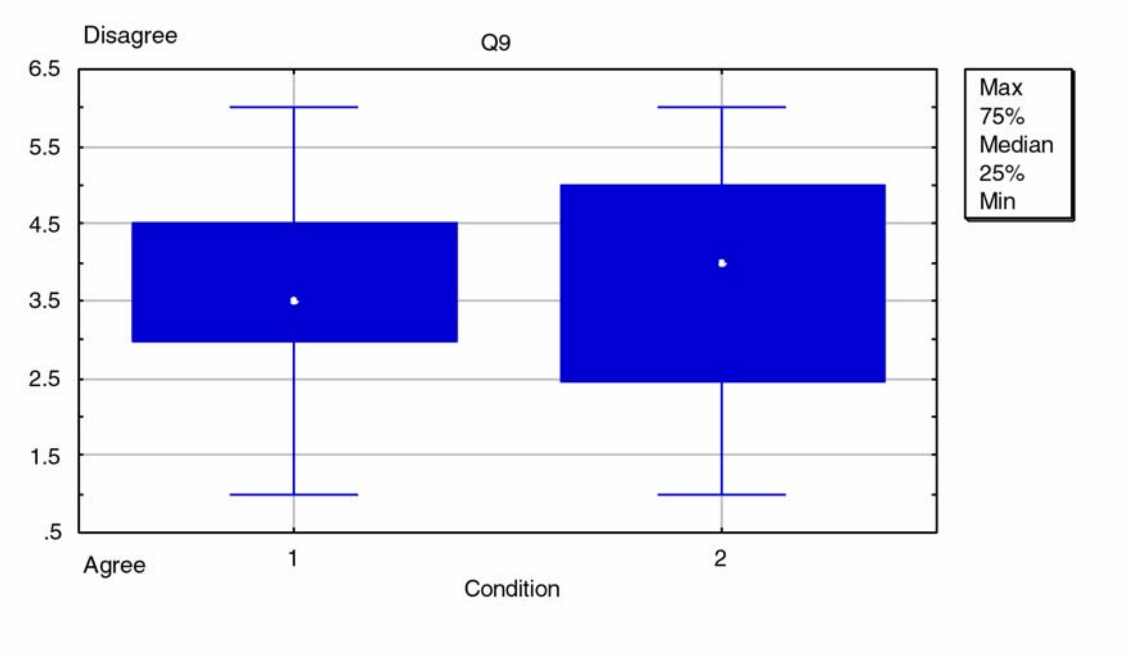


Figure 4.11. Results for Q9

Q9. *I knew what to do next at each point*

Little difference between conditions.

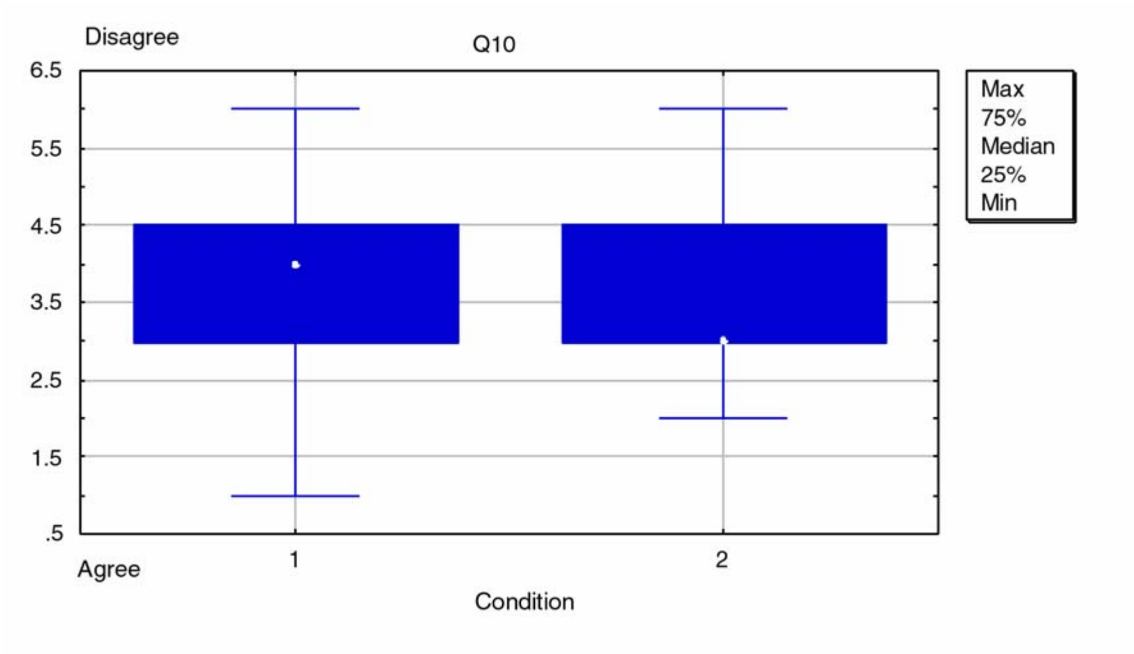


Figure 4.12. Results for Q10

Q10. *It was easy to recover from mistakes*

Little difference between conditions.

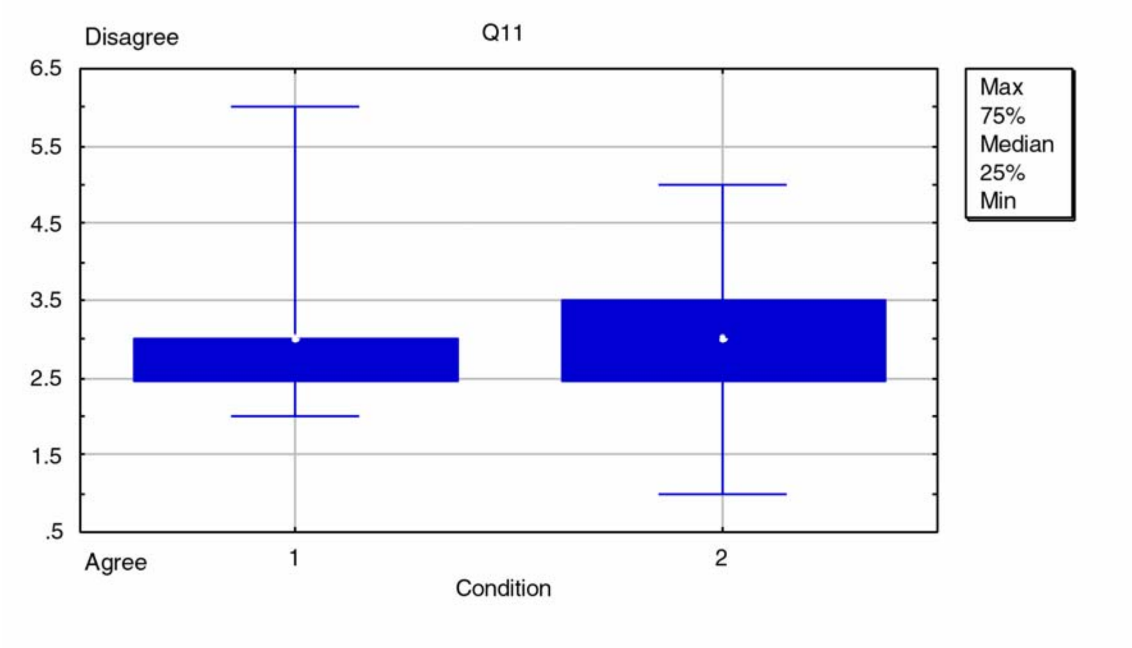


Figure 4.13. Results for Q11

Q11. *The software behaved predictably*

Little difference between conditions.

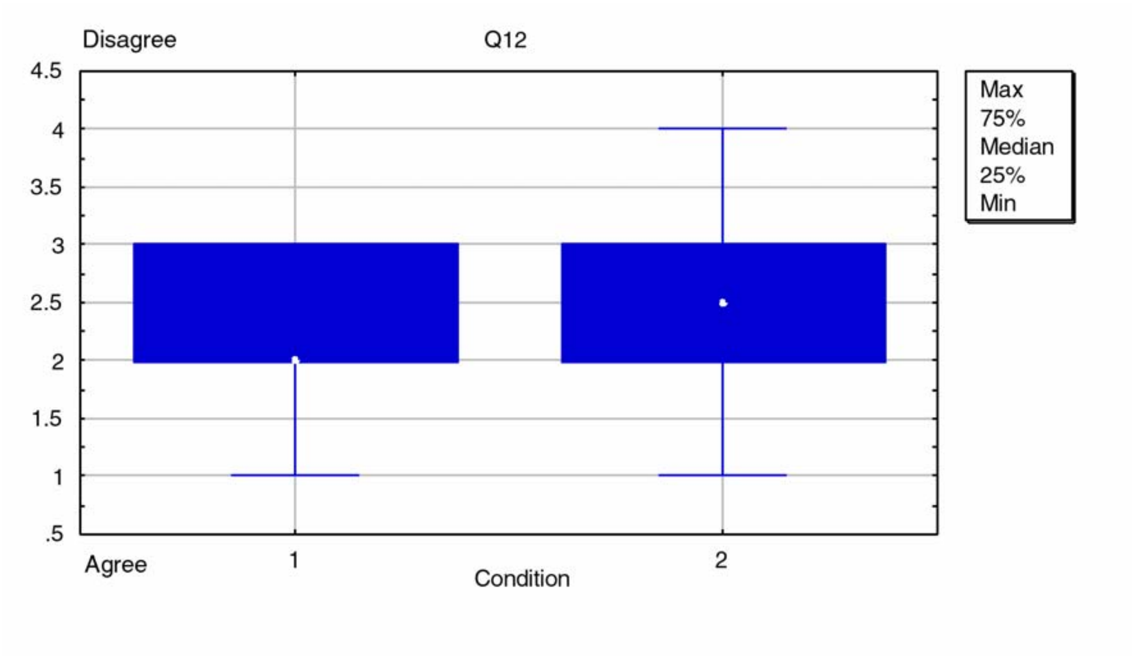


Figure 4.14. Results for Q12

Q12. *It was easy to learn how to use the software*

Little difference between conditions.

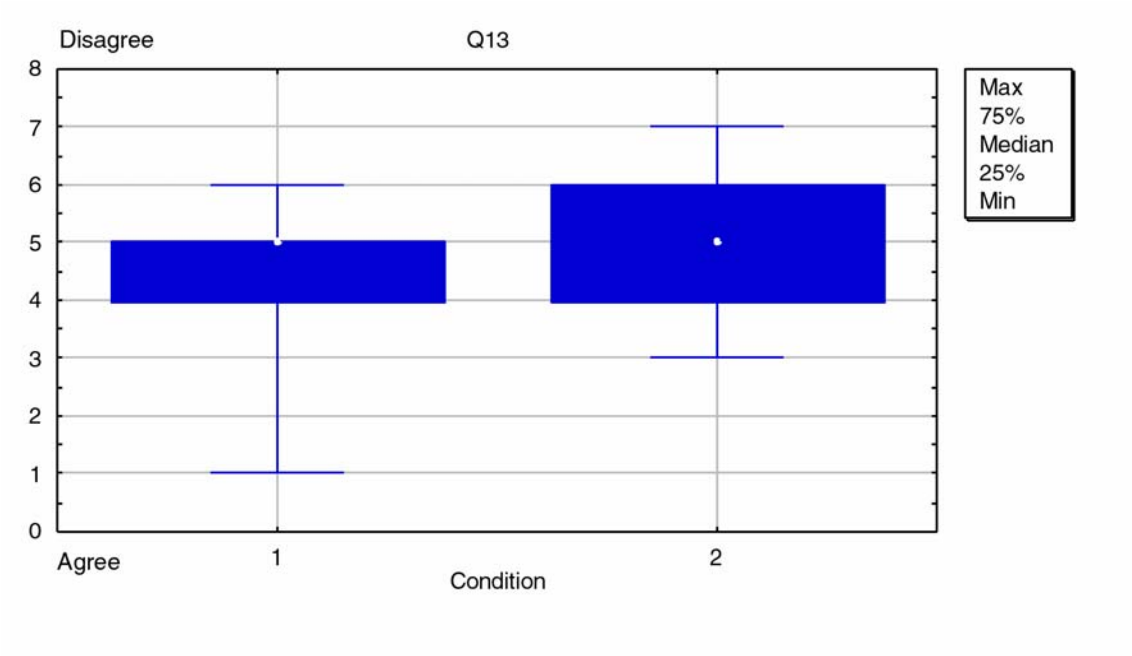


Figure 4.15. Results for Q13

Q13. *I found the software friendly to use*

Little difference between conditions.

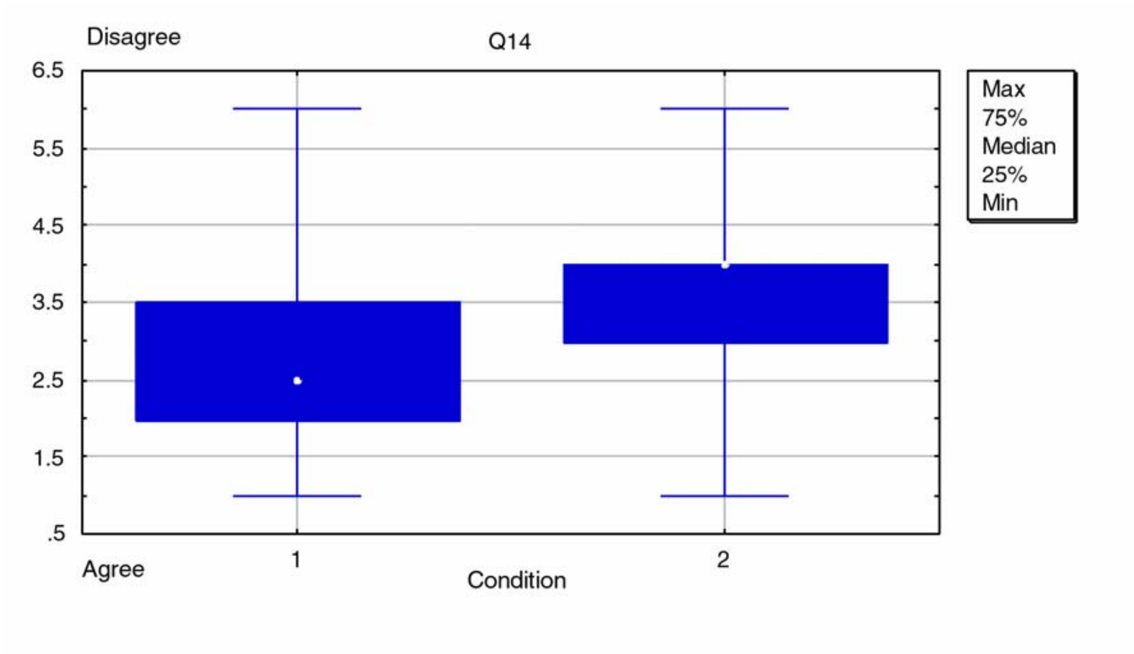


Figure 4.16. Results for Q14

Q14. *The software let me decide what to do next*

Less in condition 2 (Agent led)

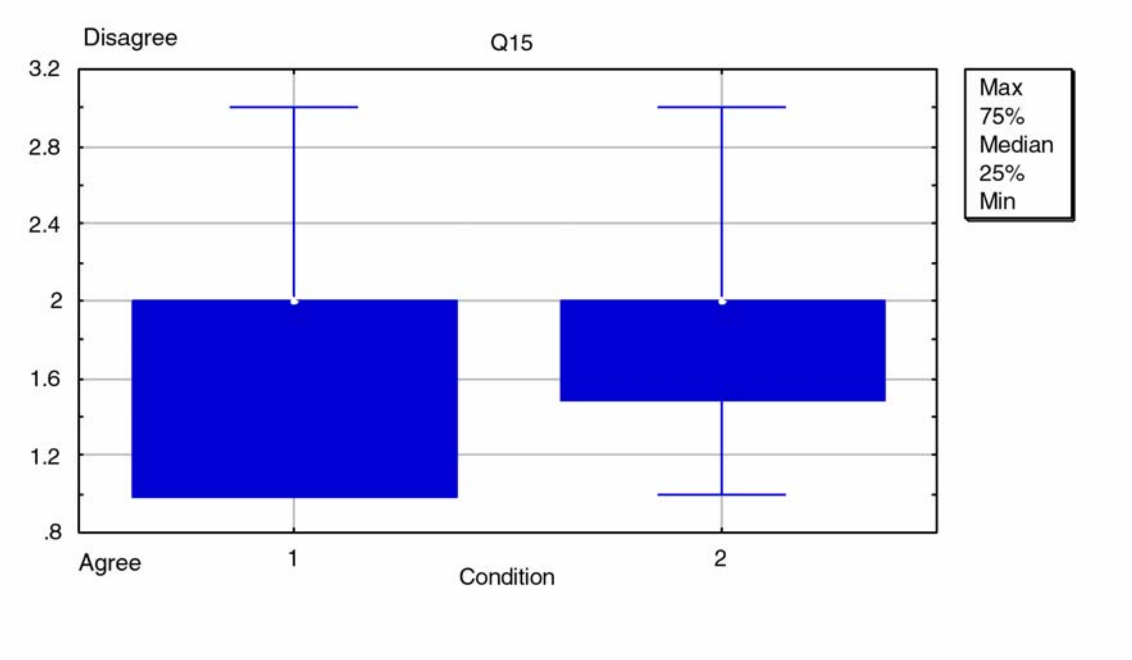


Figure 4.17. Results for Q15

Q15. *The software let me work at my own pace*

Little difference between conditions.

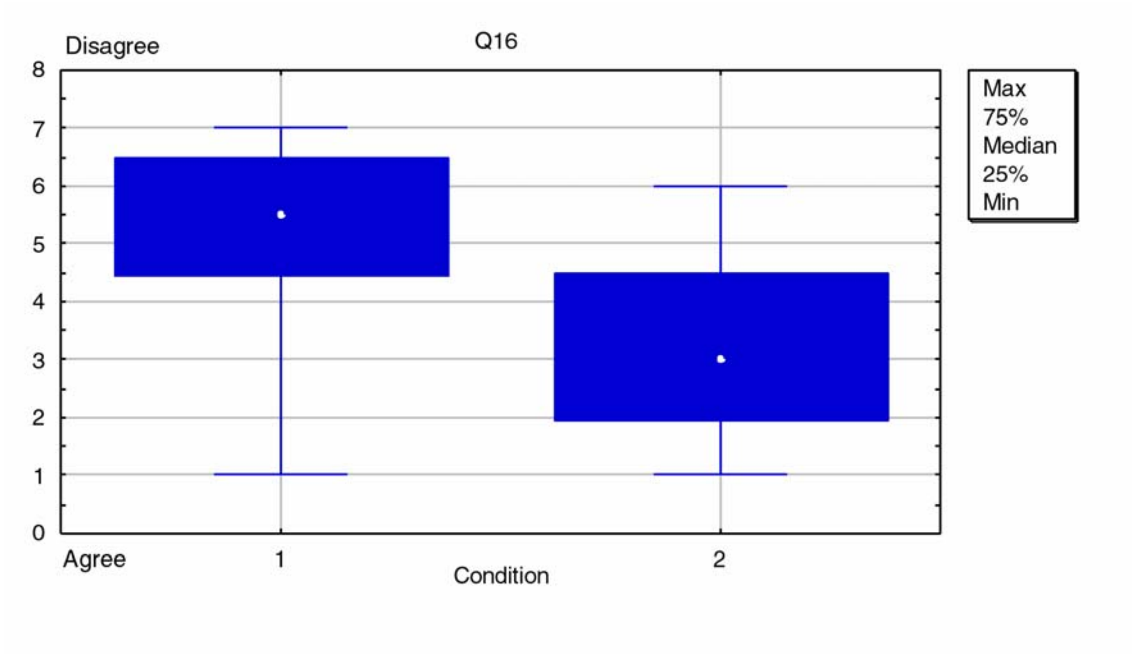


Figure 4.18. Results for Q16

Q16. *The software required me to input too much information*

Less in condition 2 (Agent led)

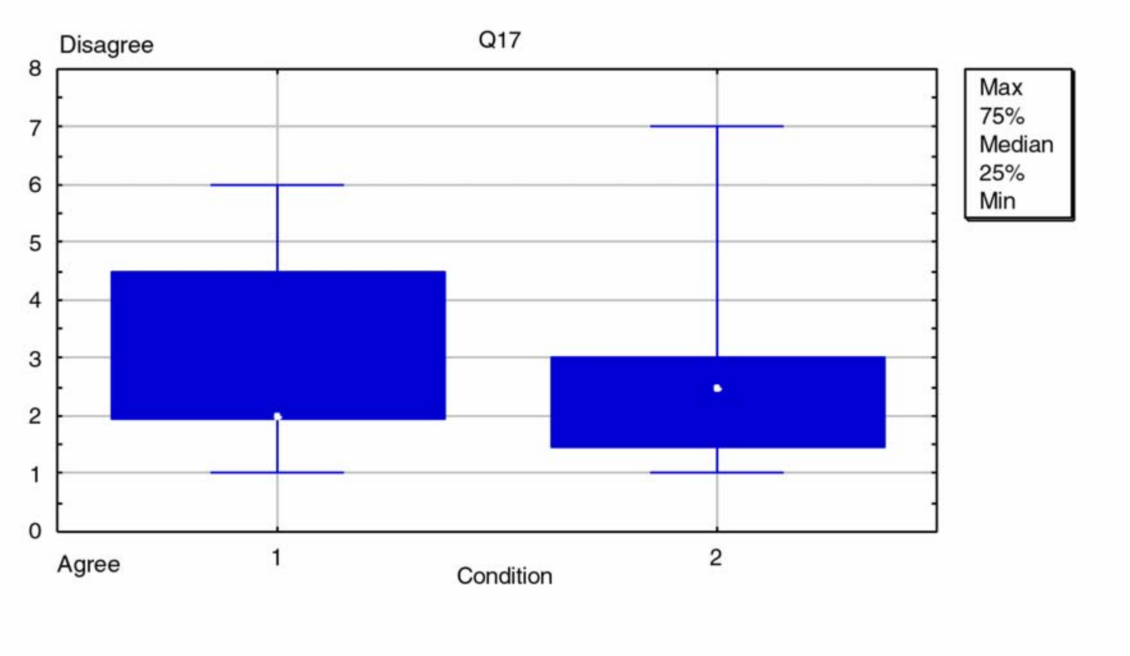


Figure 4.19. Results for Q17

Q17. *I understood the terminology used by the software*

Little difference between conditions.

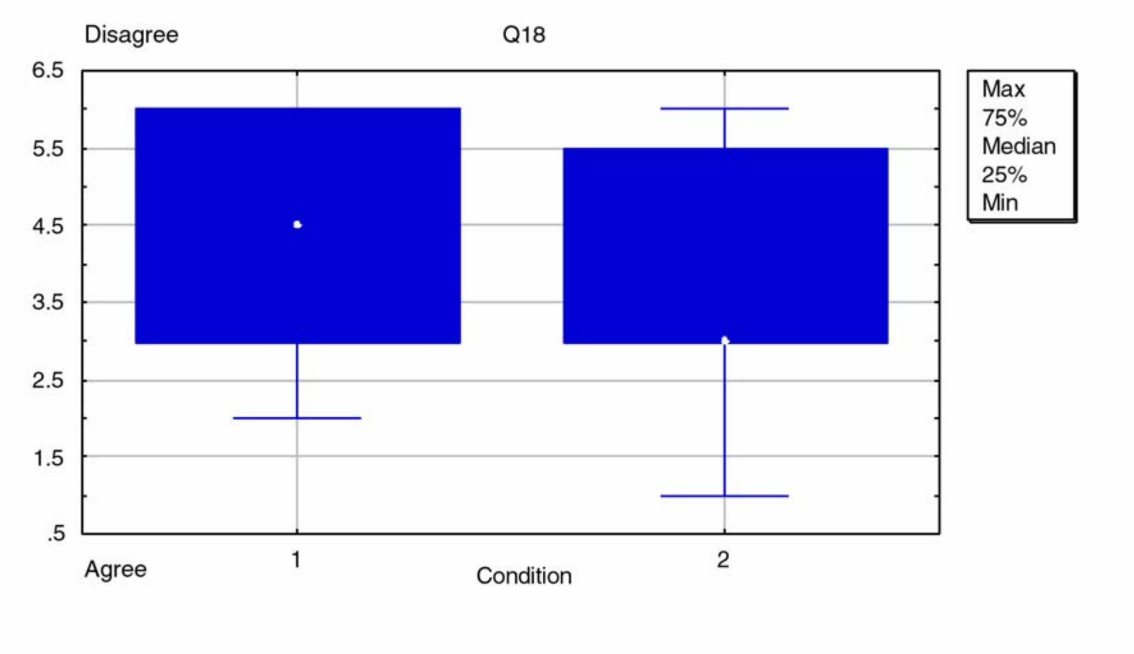


Figure 4.20. Results for Q18

Q18. *The software guided me about what action to take*

Less in condition 1

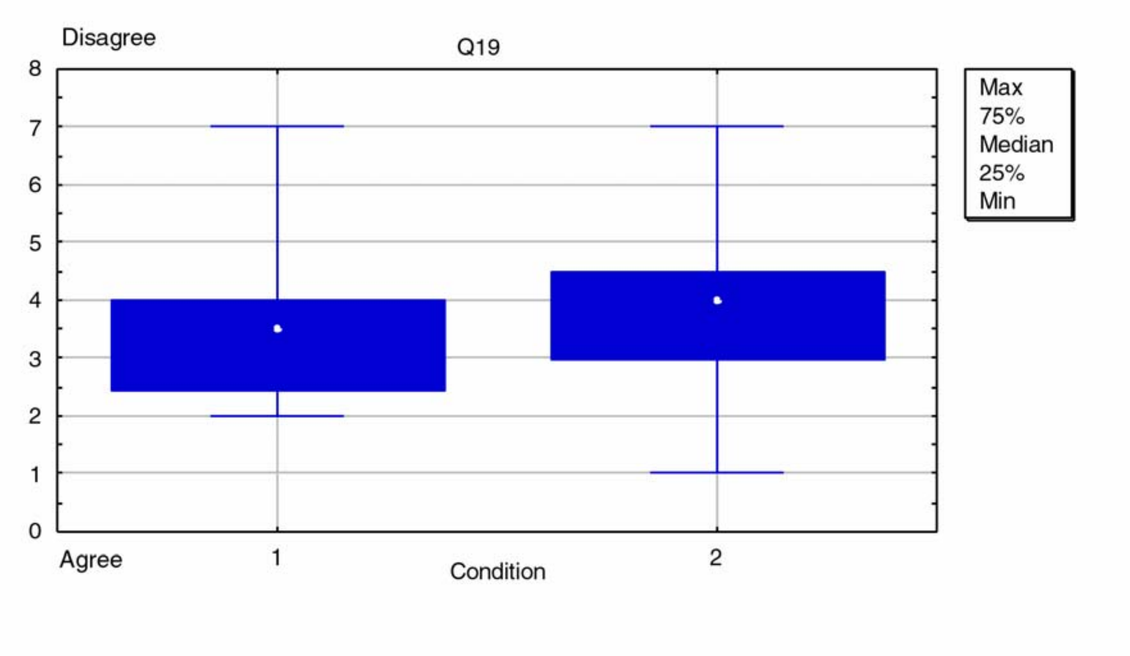


Figure 4.21. Results for Q19

Q19. *I found the software frustrating to use*

Little difference between conditions.

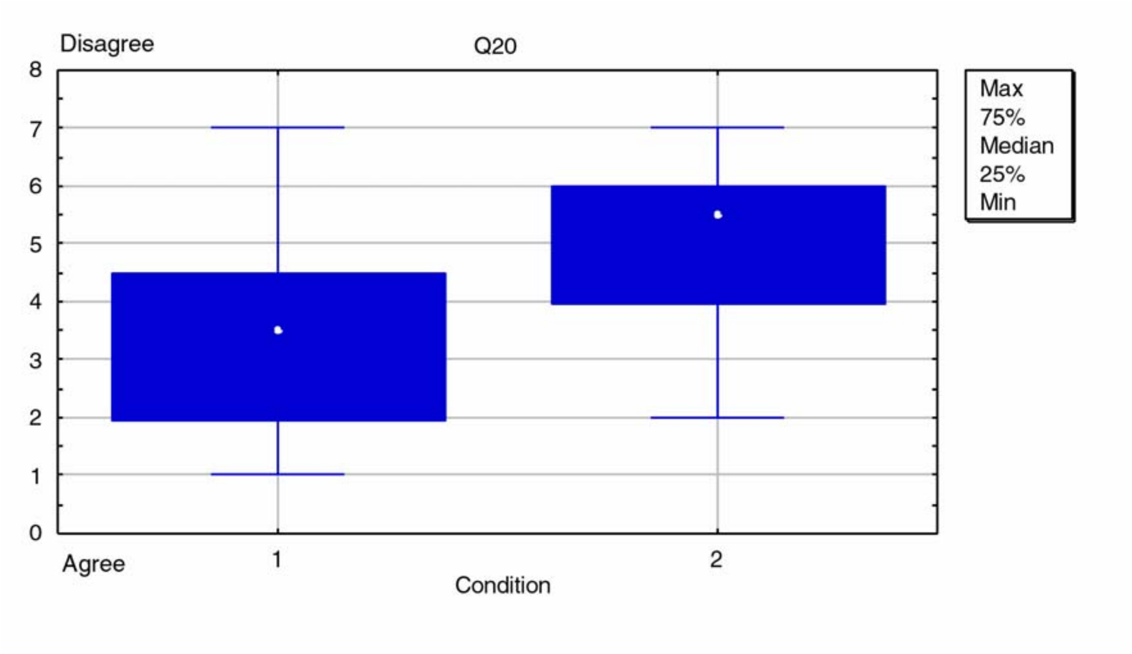


Figure 4.22. Results for Q20

Q20. *Overall, I was happy with the software*

Happier in condition 1 (User Driven)

The Kruskal-Wallis test was used to investigate the differences between the medians of conditions, but was not significant.

4.3.7 Analysis of Results

The travel scenario is loosely based on work by Rich and Sidner (1997, 1998) who investigated mixed initiative collaboration in travel planning systems. The objectives of their work were to address the inflexibility in the order in which actions must be performed by the user and the system. Their assumption is that a human-computer interface based on familiar human discourse rules and conventions will be easier to learn than one which is not (however, in their work this assumption is not validated). Their work is based on adding a collaboration manager (a software agent) to an existing graphical user interface. Their work focuses on a mixed-initiative interaction paradigm, and addresses this from a model of shared planning. Agent and User have separate windows (in addition to the main travel planning application) by which dialogues are exchanged. Communication between the agent and the user is achieved by entering text in the agent's home window. The user communicates with the agent by selecting from a menu. The agent monitors the users actions and suggests alternatives that the user can accept or reject. The system does not understand natural language, but maintains a segmented interaction history and a shared-plan model of the task domain. The agent monitors the progress of the shared plan and uses this to infer when it should take the initiative. The system therefore maintains a formal model (specified by the designer) of the collaborative tasks being undertaken. This can be seen to support contextualization by having a clearly represented task model. Unfortunately the system is not compared to no-agent or different-dialogue style conditions and it is consequently difficult to evaluate the effectiveness of the system in actual use.

In Experiment 1 the manifestation of agent interaction is subtler than that used by Reeves and Nass (1996), Nass et al. (1995) and Rich and Sidner (1997,1998) and it is also more realistic from the point of view of interface behaviour, as it exhibits agent behaviour without any additional behavioural cues (such as visual representation, labelling, etc.). Only the instigation of dialogue is varied between agent and non-agent conditions. The contextualization cues therefore wholly reside in the dialogue patterns, rather than in the structure of the interface or other attributes ascribed to the system.

4.3.7.1 Performance Results

In terms of objective data, the performance results (total task time) support Hypothesis H1. Although there is no significant difference between conditions for the number of dialogue exchanges and dialogue errors, there is a significant difference between the overall time for interaction between conditions. Taking the conventional view of an agent-based dialogue as a way of guiding the user through a task, this would appear to be a contradictory result (one would expect the agent condition to yield faster task completion as it guides the user through the task). However, using contextualization as the basis for predicting performance, the delagatory or command style of dialogue suggests a consistent view of interaction which provides the necessary cues for the user to contextualize the behaviour of the system and the role that the system adopts in relation to the user. The use of a preferred social role from the user's point of view (as delegator) and consistent dialogue behaviour for this role, support these results.

The lack of significant difference between the number of dialogues and dialogue errors was not as originally hypothesized. The explicit use of a common task model and interface design, factors determined to contribute to contextualization (Section 3.5.1), may provide sufficient cues for contextualization to mitigate the change in interaction style and discourse between conditions. The need for the user to spend more time contextualizing these cues (without the interaction and dialogue cues) may, in turn, contribute to the increased time taken in Condition 2, without a corresponding increase in number of dialogues.

4.3.7.2 Questionnaire Results

The subjective results between conditions (questionnaire) also validate the predictions made using the contextualization paradigm.

Using Contextualization as a predictive model for user preferences, we would expect a subjective evaluation to favour Condition 1 (User Driven) over Condition 2 (Agent Led). From the analysis of questionnaire results, subjects found Condition 1 *simpler* (Q1), more *comfortable* (Q2), *easier to understand* (Q3), and *more pleasant* (Q4), than Condition 2. They also felt more *informed* (Q7) and *happier* (Q20) with the software in Condition 1.

Promoting contextualization using a delagative dialogue therefore also has a positive subjective effect on the user. Although only one component of the model of contextualization was used, it has an overall positive effect on the interaction. This effect is a result of promoting contextualization within the user by appropriately managing the dynamic cues in the dialogue.

Subjects also felt that Condition 1 required them to input less information than Condition 2 (Q16). The result for question 16 is interesting, as although subjects felt that Condition 1 required them to input less information than Condition 2, an identical amount of information was required in each condition. In terms of contextualization this may be a symptom of Condition 2 requiring greater contextualization effort from the user, and hence appearing to require more input as overhead.

As we would also expect, users also found that Condition 1 allowed them to *decide what to do next* (Q14) more than Condition 2, as would be expected considering the design objectives between conditions (and thus validating the design of the software).

There was little difference between the conditions for *anticipation of behaviour* and *predictability* (Q5, Q11), *prompts* (Q6), *enjoyment* (Q8*), knowing what to do next* (Q9), *learnability* (Q12), *friendliness* (Q13), and *frustration* (Q19). The results for anticipation and behaviour indicate that although the behaviour was different between conditions, within each condition predictability was consistent.

Traditional views of agent design would suggest that *knowing what to next* (Q9) and *recovery from errors* (Q10), should have been easier in the agent-led condition. However, there was little or no difference between conditions, suggesting that an agent-led dialogue does not necessarily provide these benefits over a well-designed user-driven interface that promotes contextualization using delagative dialogues.

In summary of the 20 questions, 9 showed a difference between conditions (Questions 1, 2, 3, 4, 7, 14, 16, 17, 18, 20). These results are consistent with the predictions of contextualization theory.

To conclude, these results support Hypothesis H1 – A user-driven dialogue is more effective than an agent-led (or active) dialogue, using contextualization as a design paradigm. Specifically it is *faster* and provides *greater subjective user satisfaction.*

4.4 Experiment 2: Adapting With User Preferences

In this experiment the ability of the system to adapt the dialogue to the user (given known user preferences) is examined. In previous analyses of intelligent and adaptive user interfaces, *user preferences* were identified as factors that can be used to trigger adaptation (Section 2.2). Effectively user preferences are treated as user models when used in this way. Dialogue adaptation may therefore take place based on user preferences as a mechanism that triggers the adaptation. An identification of the user is also a component of contextualization (Section 3.5.1). Consequently, the theory of contextualization suggests that the use of user preferences in promoting contextualization will improve performance.

4.4.1 Methodology

The interface used in Experiment 1 was extended to include a preferences dialogue. The preferences form and dialogue allow the user to specify several elements relating to travel preferences which are then used to reduce the quantity of information that need be entered in subsequent dialogues (i.e., it serves as a default set of user choices).

In each trial the preferences dialogue allows the user to specify a preferred *airline*, a preferred *flight time* (morning, afternoon, evening) and the option to sort matching flights by *price*. In the User-driven condition this dialogue is triggered by the user pressing the "preferences" button on the main form. In the Agent-led condition, the preferences form is opened automatically the first time the main form is used (see Appendix A6, Figures A6.4 and A6.5). The experimental scenario description instructs the user that their preferred airline is Air Canada, and this is therefore the choice we would expect subjects to make in the preferences dialogue. The preferences are then used by the software to filter and sort the data retrieved, so that flight data with the preferred attributes appear at the top of results lists.

4.4.2 Experimental Conditions

The User Driven and Agent Led conditions in Experiment 1 were administered again, this time with a set of user preferences being specified and used in the adaptation of the interface. The resulting data can consequently be compared with data for the Experiment 1 conditions to ascertain if there is any variation in performance. There are therefore 2 independent variables (Dialogue Style and Aiding), giving a 2 x 2 factorial design.

1.	User-Driven	No Preferences
2.	Agent Led	No Preferences
3.	User-Driven	Preferences
4.	Agent-Led	Preferences

4.4.3 Hypotheses

H1 A dialogue which adapts to user preferences will be more effective than one which does not.

More specifically, a dialogue adaptation based on user preferences will:

1. have fewer dialogue exchanges;
2. have fewer broken dialogues;
3. be faster;
4. provide greater user satisfaction.

The null hypothesis is therefore:

H0 There is no performance gain using dialogue adaptation with user preferences, regardless of the dialogue style used.

4.4.4 Data Collection

20 subjects participated in the experiment, drawn from the same user population as Experiment 1. In total there are therefore a total of 40 participants, 10 per condition. Subjects were randomly assigned to one of Conditions 3 and 4 balanced by primary personality trait (see Experiment 3).

The following data were collected:

Objective –	Number of Dialogue Exchanges	(Frequency)
	Number of broken dialogues	(Frequency)
	Time Taken	(Quantitative)

These data were logged by the software.

Subjective -	User Satisfaction Questionnaire	(Categorical)

This data was obtained after each software trial.

The complete data sets for all experiments are listed in Appendix B.

4.4.4.1 Performance Results

The objective (performance) results are given in Table 4.4. Once again the logarithm of time was taken to ensure that possible skew in the distribution of time did not play a factor (Figure 4.23)

Condition	n	No. Dialogues		Broken Dialogues		Time		Log (Time)	
		Mean	SD	Mean	SD	Mean	SD	Mean	SD
1	10	43.6	16.249	2.6	2.412	509.80	146.23	6.1935	0.3082
2	10	44.5	17.551	4.1	6.871	704.90	238.52	6.5069	0.3380
3	10	45.2	15.991	1.4	1.8974	576.2	209.52	6.2946	0.3747
4	10	47.4	19.681	8.2	10.507	627.7	231.25	6.3825	0.3634

Table 4.4. Summary of performance results for all conditions

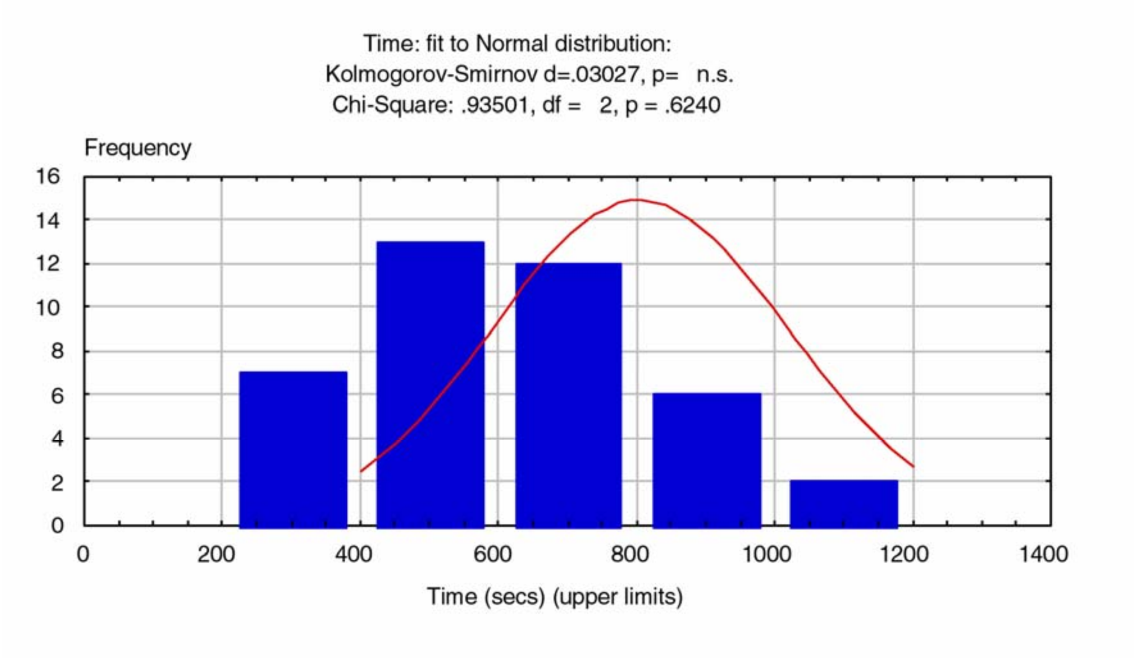


Figure 4.23. Overall distribution of Time for all conditions

The results were analysed by the *Statistica* statistical analysis program, and the means of each condition were compared using analysis of variance. The results are shown in Table 4.5

The results show that there is no significant difference between the means of any of the variables measured, causing the rejection of Hypothesis H1 and accepting Hypothesis H0 – there is no performance gain using dialogue adaptation with user preferences, regardless of the dialogue styles used.

Variable	**Mean Square Effect**	**Mean Square Error**	**F(df 1,2) 3,36**	**p-level**
Dialogues	**96.89**	**282.2**	**0.343341**	**0.794121**
Broken Dialogues	**87.82**	**41.76**	**2.103173**	**0.1169595**
Time	**67957.63**	**43914.84**	**1.547487**	**0.2189921**
Log Time	**0.18**	**0.12**	**1.469934**	**0.2390165**

Table 4.5. Analysis of performance data between all conditions

Based on the original hypothesis and using the model of contextualization, an optimal result would have shown Condition 3 (User-Driven with Preferences) to have significant performance advantages over other conditions. However, from these results it must be concluded that no condition offers a significant performance advantage over the others. However, this does not clarify performance differences between conditions. It may be that dialogue style and adaptation based on user preference have effects of different sizes on the results. It is therefore important to compare results between conditions for significant differences in performance. The results of this analysis are shown in Tables 4.6 – 4.9.

Variable	**Mean Square Effect**	**Mean Square Error**	**F(df 1,2) 1,18**	**p-level**
Dialogues	**211.25**	**317.52**	**0.665318**	**0.425350**
Broken Dialogues	**156.80**	**58.11**	**2.69827**	**0.117811**
Time	**69502.05**	**37432.09**	**1.85675**	**0.189807**
Log Time	**0.1800**	**0.10999**	**1.57421**	**0.225644**

Table 4.6. Condition 1 (User-driven) with Condition 4 (Agent-led with preferences)

Variable	**Mean Square Effect**	**Mean Square Error**	**F(df 1,2) 1,18**	**p-level**
Dialogues	**92.45**	**251.69**	**0.36731**	**0.552041**
Broken Dialogues	**7.2**	**4.71**	**1.528302**	**0.232257**
Time	**22044.8**	**32642.62**	**0.675338**	**0.421949**
Log Time	**0.05**	**0.12**	**0.434812**	**0.517985**

Table 4.7. Condition 1 (User-driven) with Condition 3 (User-driven with preferences)

Variable	Mean Square Effect	Mean Square Error	F(df 1,2) 1,18	p-level
Dialogues	**72.2**	**246.88**	**0.292439**	**0.595291**
Broken Dialogues	**36.45**	**25.40**	**1.434726**	**0.24653**
Time	**82818.45**	**50397.578**	**1.643302**	**0.216139**
Log Time	**0.22999**	**0.12999**	**1.768493**	**0.200176**

Table 4.8. Condition 2 (Agent-led) with Condition 3 (User-Driven with preferences)

Variable	Mean Square Effect	Mean Square Error	F(df 1,2) 1,18	p-level
Dialogues	**180**	**312.71**	**0.575611**	**0.457858**
Broken Dialogues	**84.05**	**78.81**	**1.066549**	**0.315403**
Time	**29799.2**	**55187.05**	**0.539967**	**0.471912**
Log Time	**0.08**	**0.12**	**0.627941**	**0.438432**

Table 4.9. Condition 2 (Agent-led) with Condition 4 (Agent-led with preferences)

For all these comparisons there is no significant difference between any of the variables. This is somewhat unexpected, as from the results of Experiment 1 it might be predicted that Condition 3 (User-Driven with preferences) would show similar (or greater) time performance advantages over Condition 2 as did Condition 1 (User-Driven). Considering the use of a model of contextualization, and without foreknowledge of the relative size of performance advantage of dialogue style versus preferences, it would be expected that:

1. Condition 3 (User-Driven with Preferences) would have higher performance than Condition 1 (User-Driven).
2. Condition 4 (Agent-led with Preferences) would have a higher performance than Condition 2 (Agent-led)

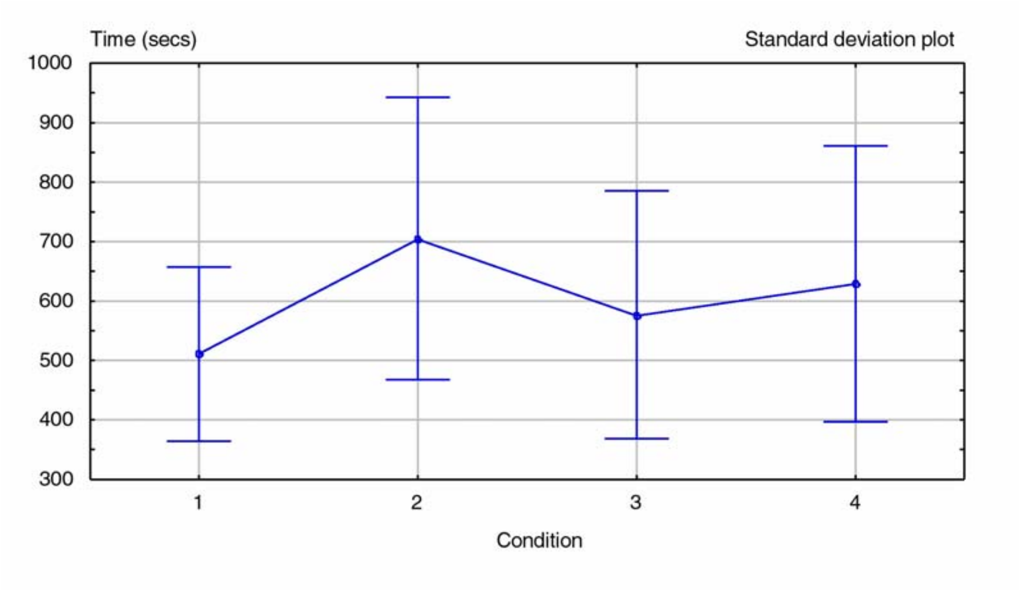


Figure 4.24. Plot of Mean and Standard Deviation of Time by Condition

Comparing the mean and standard deviation of Time and Log(Time) between conditions (Figure 4.24 and Figure 4.25) we can see that the relative performance (time taken for task) does follow this prediction to some extent. Mean task time in Conditions 1 and 3 is less than in Conditions 2 and 4 (but not significantly).

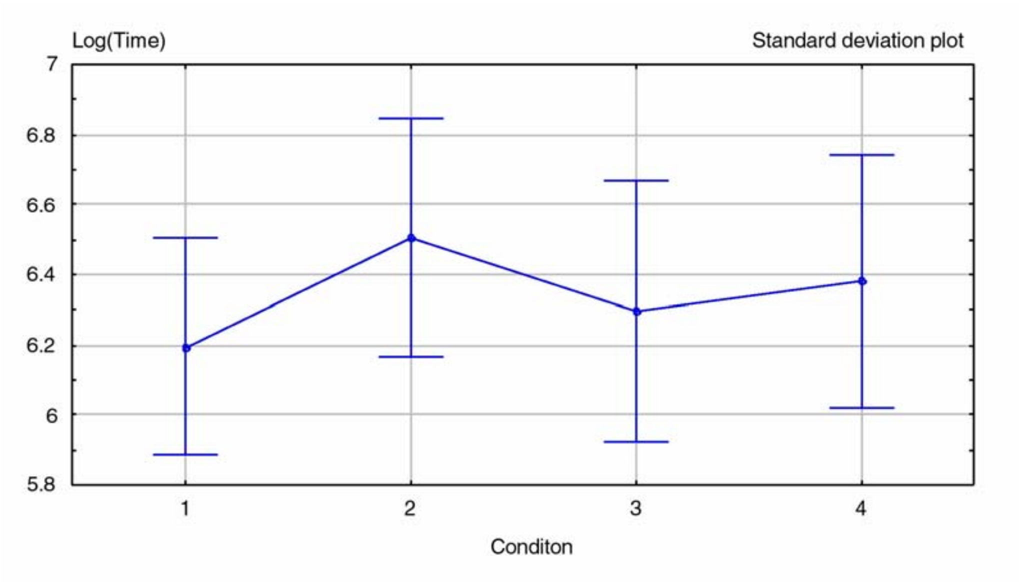


Figure 4.25. Plot of Mean and Standard Deviation of Log(Time) by Condition

Note also that the use of preferences in an Agent-led dialogue (Condition 4) does seem to provide a non-significant speed increase over Agent-led dialogue alone (Condition 2). In addition, Condition 3 shows no speed improvement over Condition 1 – an unexpected result.

Examining the total number of dialogues for each condition also provides results consistent with initial predictions (Figure 4.26). Because a common task model exists for all conditions and only the dynamics of presentation change between conditions, a large variation in total number of dialogues would not be expected. This is particularly true given the analysis of Experiment 1 which suggests that although the number of dialogues shows little variation, the amount of time taken per dialogue may vary considerably as a function of difficulty.

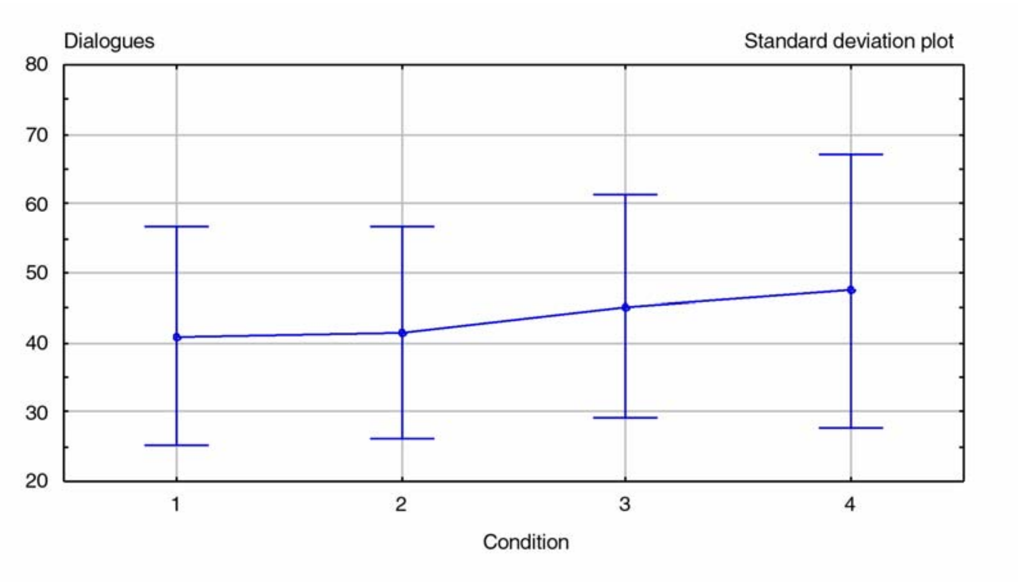
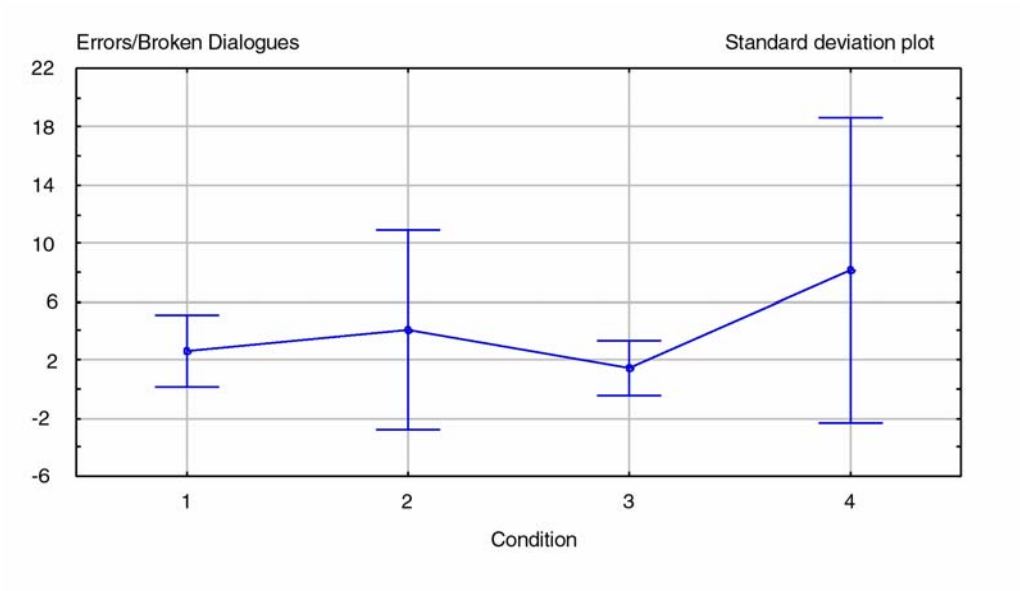


Figure 4.26. Mean and Standard Deviation of Dialogues per Condition

Conditions 2 and 3 show little variation in the mean number of dialogues. The introduction of a preferences dialogue (Conditions 3 and 4) seems to increase the mean number of dialogues slightly (again, not significantly). One reason for the increase in mean number of dialogues in Conditions 3 and 4 might be that the number of dialogues may be influenced by the number of erroneous



dialogues in these conditions or that the dialogue overhead in entering preferences has a disproportionate effect on the overall number of dialogues.

Figure 4.27 Mean and Standard Deviation of Broken Dialogues per Condition

An examination of Broken Dialogues (Errors) shows that there are lower mean numbers of errors in Conditions 1 and 3, compared with Agent-Led conditions (2 and 4) and Agent-led conditions have higher standard deviations than the User-Driven conditions. Condition 4 shows more errors than any other condition and this may contribute to the higher mean number of dialogues per condition in this condition. However, Condition 3 shows a decrease in errors over other conditions, so the increase in dialogues in Condition 4 may be due to overheads in completing preference dialogues.

Performance improvement is based on maximising the ability of the user to contextualize. Two of the hypothesized ways of helping this contextualization are to maintain a delegatory (or User-Driven) dialogue style, and to enable the system to adapt based on user preferences. Although there are no statistically significant differences between conditions (except Conditions 1 and 2), the preceding qualitative discussion supports the use of a delegative dialogue style and user preferences. Use of User-Driven dialogue (Condition 1) or User-Driven dialogue with preferences (Condition 3) is generally faster and has fewer errors (or broken dialogues) than use of Agent-Led dialogue or Agent-Led dialogue with preferences. Both Agent-Led Conditions show more errors (and a higher variance) when compared with User-Driven conditions. Contextualization suggests that a consistent, delegative dialogue will be preferred over an unpredictable, mixed-initiative dialogue, and the results tend to support this.

Contextualization also predicts that using preferences to tailor adaptation and system behaviour will also improve interaction. Use of preferences in the Agent-Led dialogue (Condition 4) seems to reduce average task completion time when compared with Agent-Led without preferences (Condition 2). However, the User-Driven dialogue with preferences (Condition 3) has a higher mean task time than User-Driven condition (Condition 1), although it has a fractionally lower average error rate. Here the use of preferences to guide interface adaptation would be expected to improve performance, but does not. This may be due to the number of user preferences dialogues becoming a disproportionate overhead in terms of the total number of dialogues in a given interaction.

Contextualization predicts that the use of interface adaptation with preferences should improve performance regardless of other factors. However, between Conditions 2 and 4, there are generally more errors and a greater variation in error in Agent-Led with preferences (Condition 4) than in Agent-Led (Condition 2). Condition 4 also shows a higher mean number of dialogues to complete a task than Condition 2, but this may be due to the greater number of errors per task in Condition 4. Conversely, the average total task time is faster in Condition 4 than in Condition 2. So although Condition 4 has more dialogue exchanges (which contain more errors) than Condition 2, it is generally faster, supporting predictions from contextualization.

These performance results, while not statistically significant, suggest that there may be effects on performance from changing techniques of dialogue instigation and adaptation via user preferences (See Chapter 5 for further discussions).

In addition to objective measures of interaction quality such as time, dialogues and errors, it is also important to examine subjective assessments of the interaction experience by the user. This information was elicited by means of a user satisfaction questionnaire.

4.4.4.2 User Satisfaction Questionnaire Results

The user satisfaction questionnaire used in Experiment 1 was again administered for obtaining subjective user feedback on the quality of the user interface and the interaction that takes place. The questionnaire was given to each experimental participant after they had completed the flight scheduling task using the appropriate software package.

The results from the questionnaire are shown in Table 4.10 and graphically in Figures 4.28 to 4.47. The data from the questionnaires was reliable (Condition3, Cronbach's alpha=0.88; Condition4, Cronbach's alpha=0.70).

	Condition 3				Condition 4			
	Mean	**Median**	**Mode**	**Freq. Mode**	**Mean**	**Median**	**Mode**	**Freq. Mode**
Q1	**3.1**	**3**	**m**	**-**	**3.9**	**4**	**5**	**3**
Q2	**3.2**	**3**	**m**	**-**	**3.9**	**3**	**3**	**4**
Q3	**307**	**4**	**m**	**-**	**4.5**	**5**	**5**	**4**
Q4	**4.3**	**4.5**	**6**	**3**	**4.4**	**4**	**m**	**-**
Q5	**3.1**	**3**	**2**	**4**	**4.1**	**4**	**4**	**4**
Q6	**3.6**	**4**	**4**	**4**	**4.1**	**4**	**4**	**3**
Q7	**4.1**	**4**	**m**	**-**	**5**	**5**	**7**	**3**
Q8	**3.9**	**3.5**	**3**	**3**	**4.6**	**4.5**	**4**	**3**
Q9	**3.0**	**3**	**m**	**-**	**4.3**	**4.5**	**7**	**3**
Q10	**2.9**	**2.5**	**2**	**4**	**3.8**	**3.5**	**m**	**-**
Q11	**2.7**	**3**	**m**	**-**	**3.7**	**4**	**m**	**-**
Q12	**2.7**	**2**	**2**	**5**	**3.6**	**3.5**	**3**	**4**
Q13	**3.5**	**3.5**	**m**	**-**	**4.6**	**4**	**4**	**5**
Q14	**3.1**	**2.5**	**2**	**4**	**3.5**	**3**	**2**	**4**
Q15	**2.7**	**2**	**2**	**4**	**2**	**1.5**	**1**	**5**
Q16	**5.1**	**5.5**	**6**	**4**	**5.4**	**6**	**m**	**-**
Q17	**3.1**	**2.5**	**2**	**4**	**2.6**	**3**	**3**	**4**
Q18	**4.7**	**5.0**	**5**	**3**	**4.9**	**5.5**	**6**	**4**
Q19	**4.7**	**5.0**	**5**	**3**	**4.5**	**5**	**5**	**4**
Q20	**3.5**	**3.0**	**2**	**3**	**5.1**	**5**	**5**	**3**

Table 4.10. Results from questionnaire survey for Experiment 2

Because of the nature of the 7-point Likert scale used, the questionnaire data was again described in terms of mean, median and mode.

To represent and compare the data graphically, box and whisker plots are used to describe the median point (circle), the 25% to 75% range, and the maximum and minimum responses. This gives a good indication of the median response together with the overall clustering of responses in comparison with the median as the quartile range includes 50% of the data around the median.

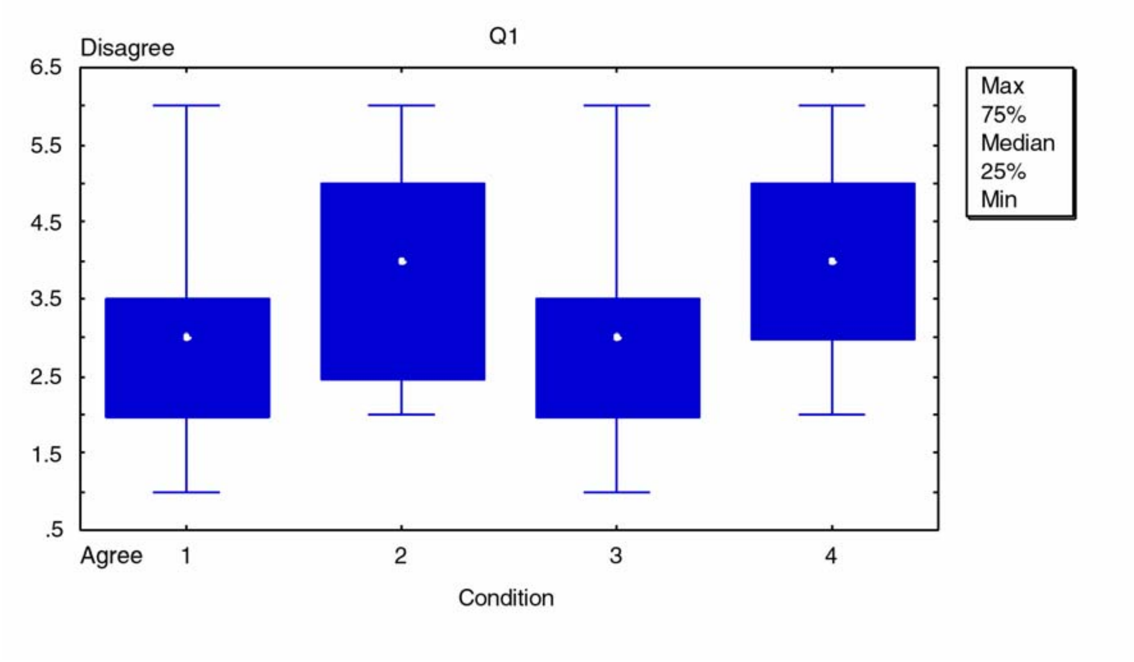


Figure 4.28. Results for Q1. *It was simple to use this software*

Conditions 1 and 3 are simpler than Conditions 2 and 4. Conditions 1 and 3 are approximately the same, as are Conditions 2 and 4.

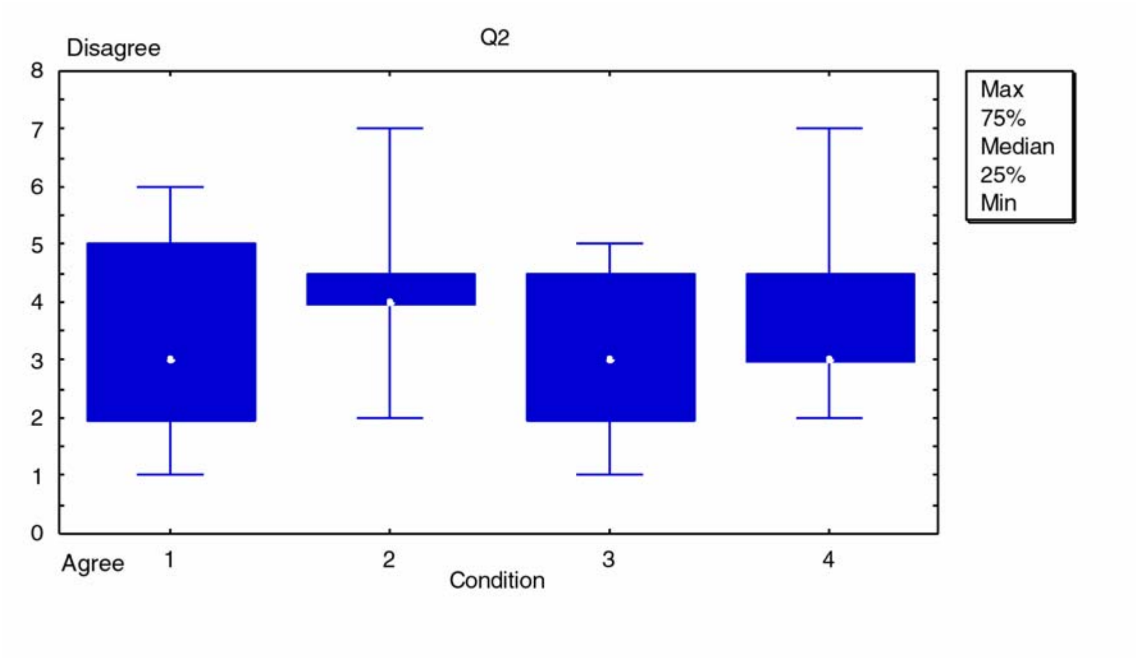


Figure 4.29. Results for Q2. *I felt comfortable using the software*

Conditions 1 and 3 are more comfortable than conditions 2 and 4.

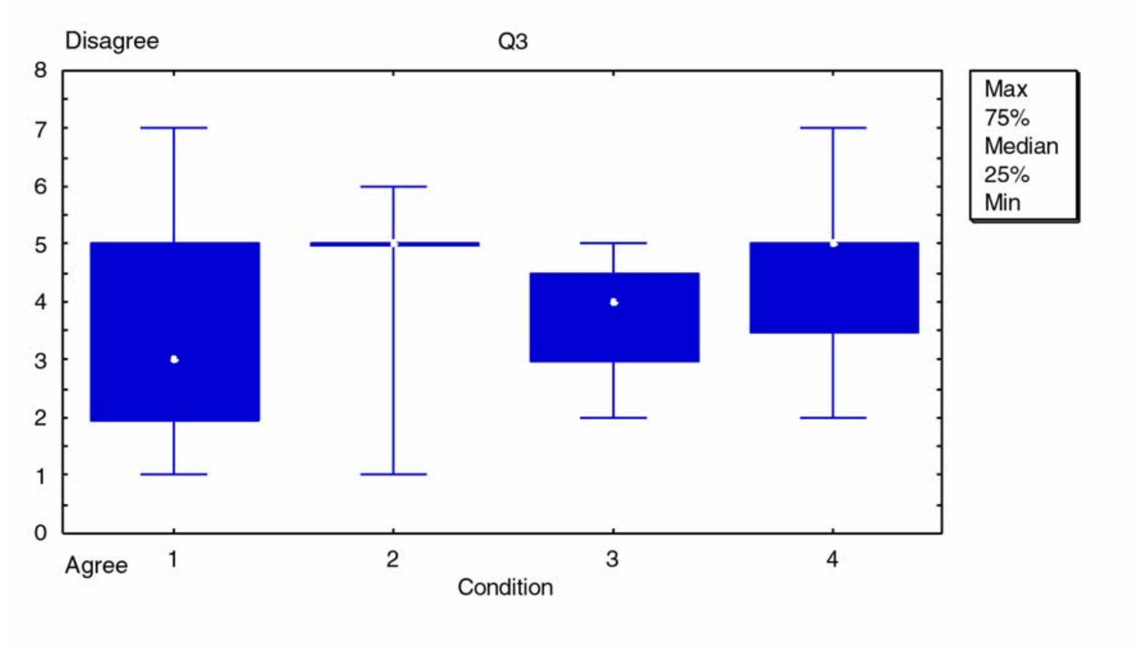


Figure 4.30. Results for Q3. *Information provided by the software was easy to understand*

Condition 1 was the easiest to understand, followed by Condition 3.

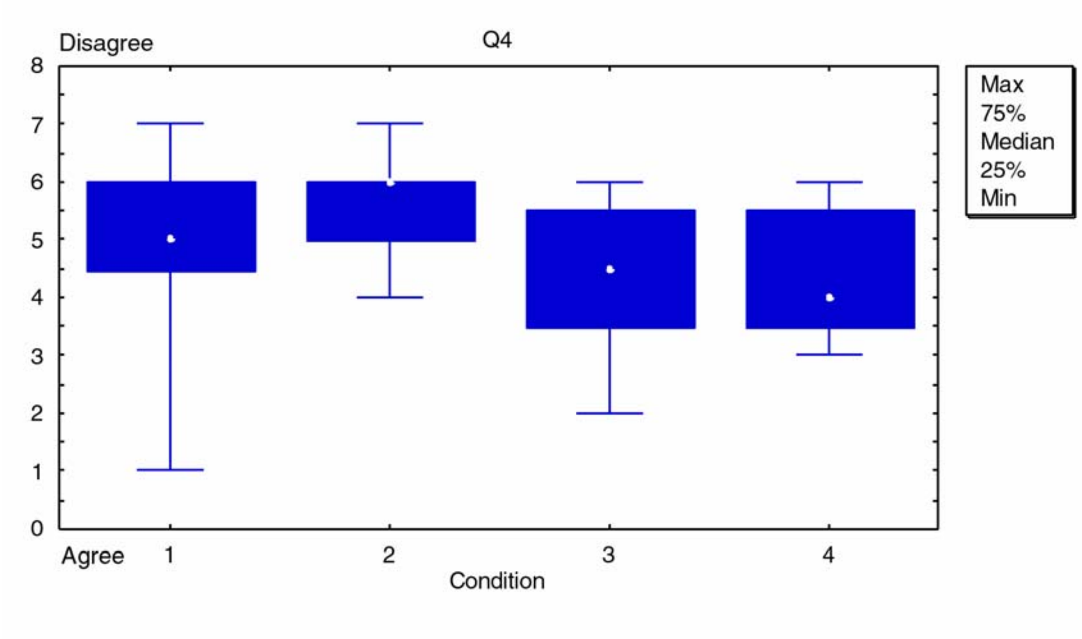


Figure 4.31. Results for Q4. *The interface to the software was pleasant*

Least pleasant in Condition 2. Most pleasant in Conditions 3 and 4.

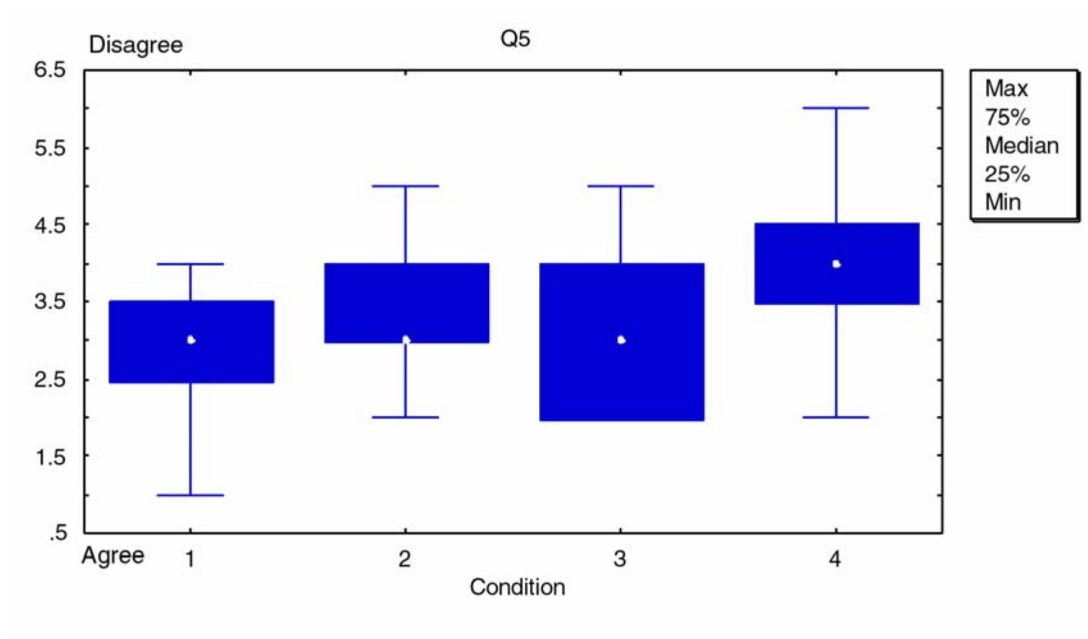


Figure 4.32. Results for Q5. *I could anticipate the behaviour of the software*

Most able to anticipate behaviour in Conditions 1 and 3.

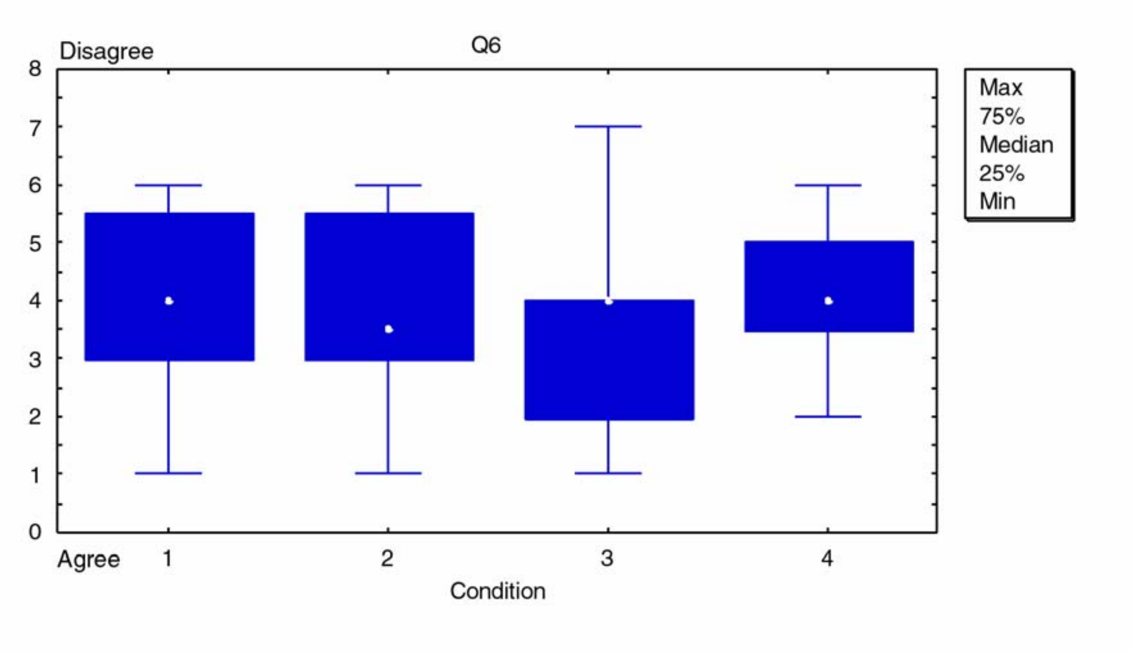


Figure 4.33. Results for Q6. *Prompts for me to take action were clear*

Little difference in Medians and Modes. Medians are 4,3.5,4,4; Modes are 6,-,4,4 respectively.

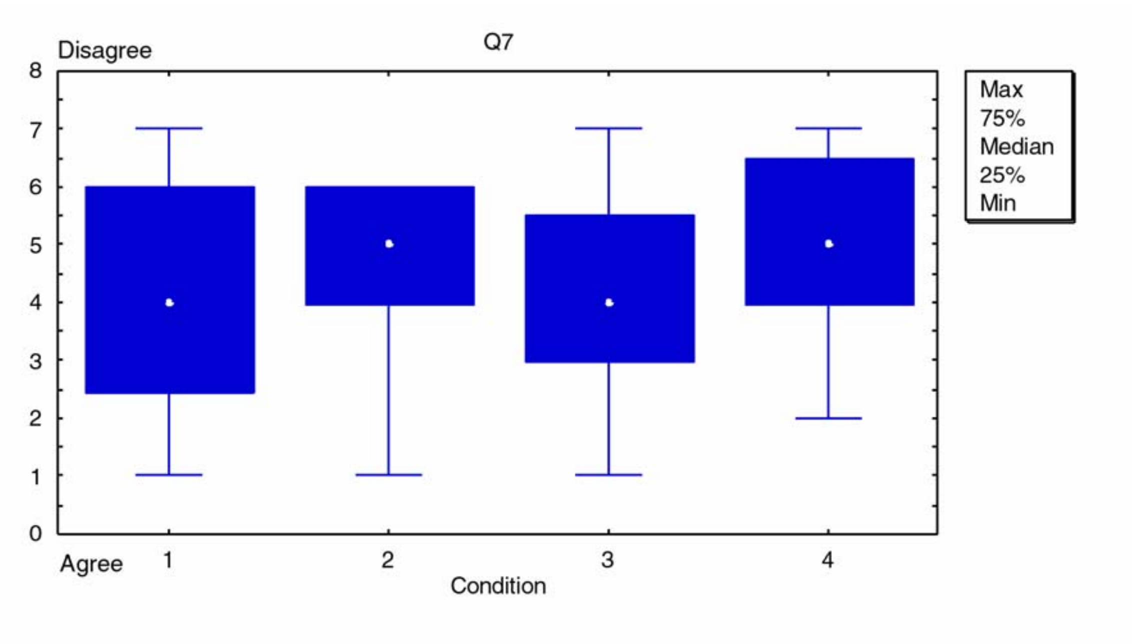


Figure 4.34. Results for Q7. *The software informed me about its progress*

More informed in Conditions 1 and 3

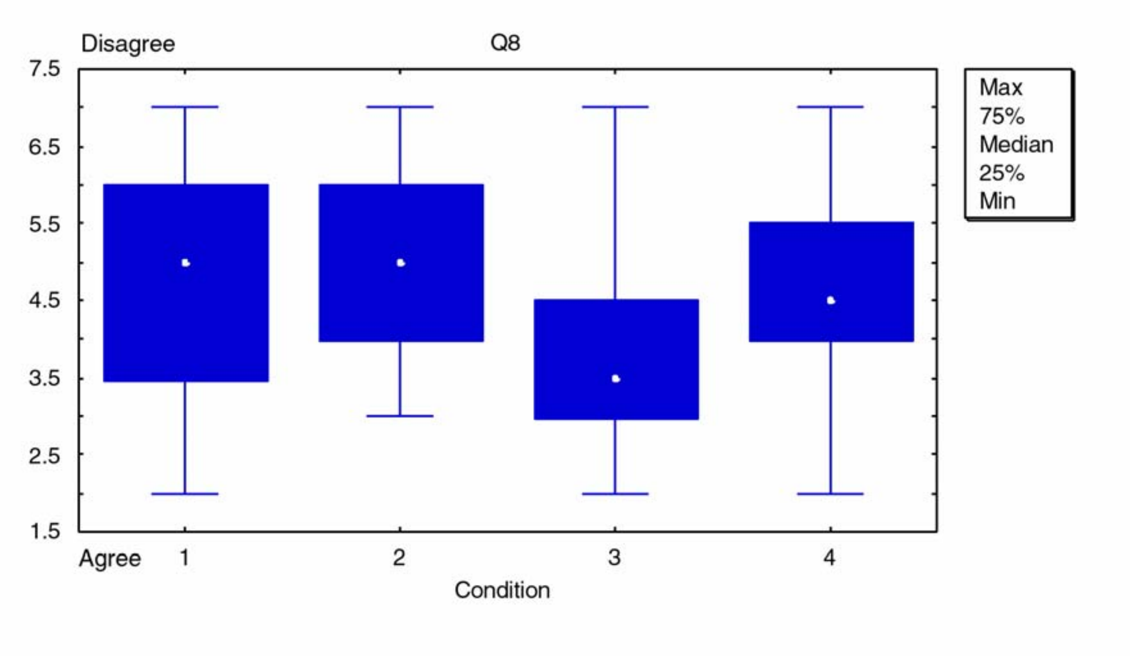


Figure 4.35. Results for Q8. *I enjoyed using the software*

Most enjoyable in Condition 3

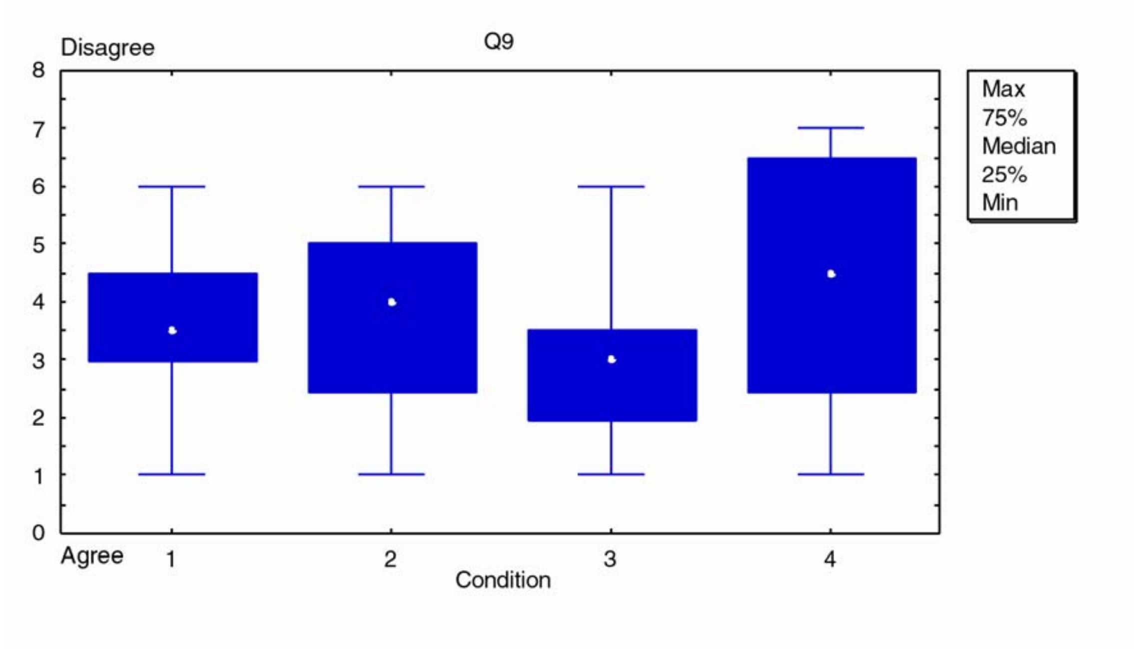


Figure 4.36. Results for Q9. *I knew what to do next at each point*

Condition 3 seems the best.

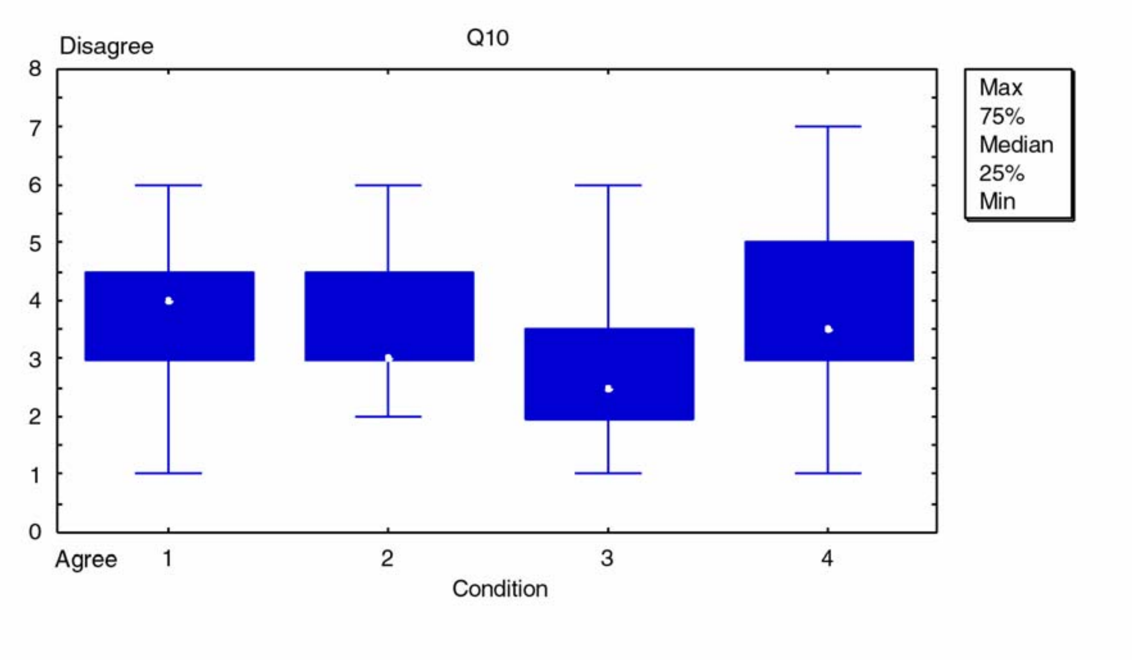


Figure 4.37. Results for Q10. *It was easy to recover from mistakes*
Easiest to recover in Condition 3.

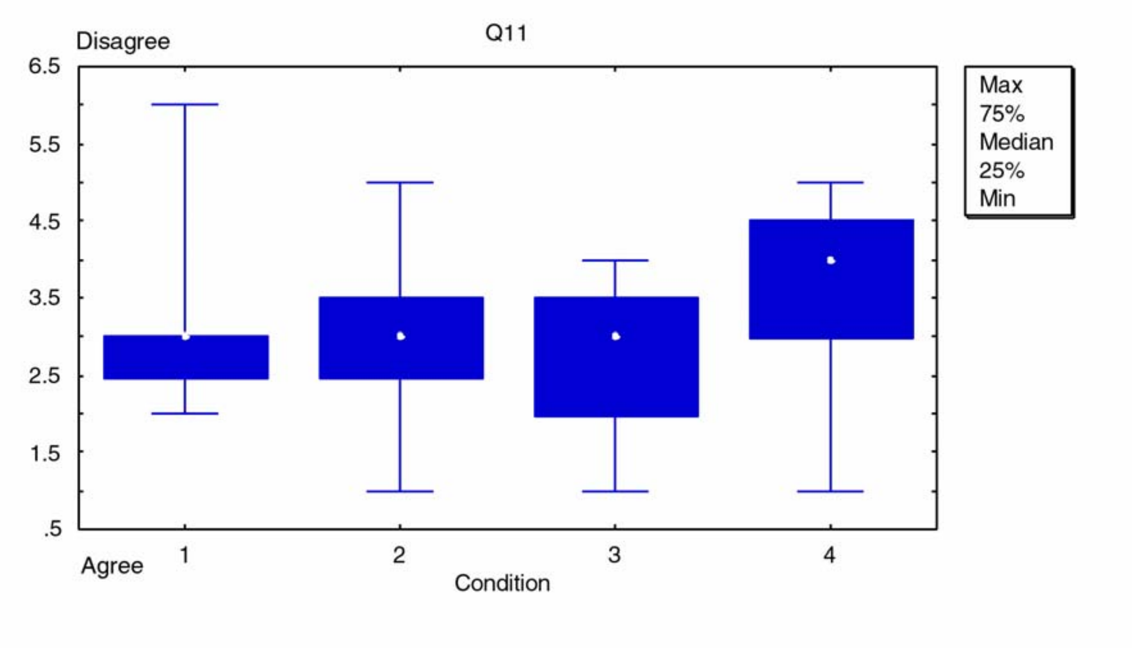


Figure 4.38. Results for Q11. *The software behaved predictably*
Least predictable in Condition 4.

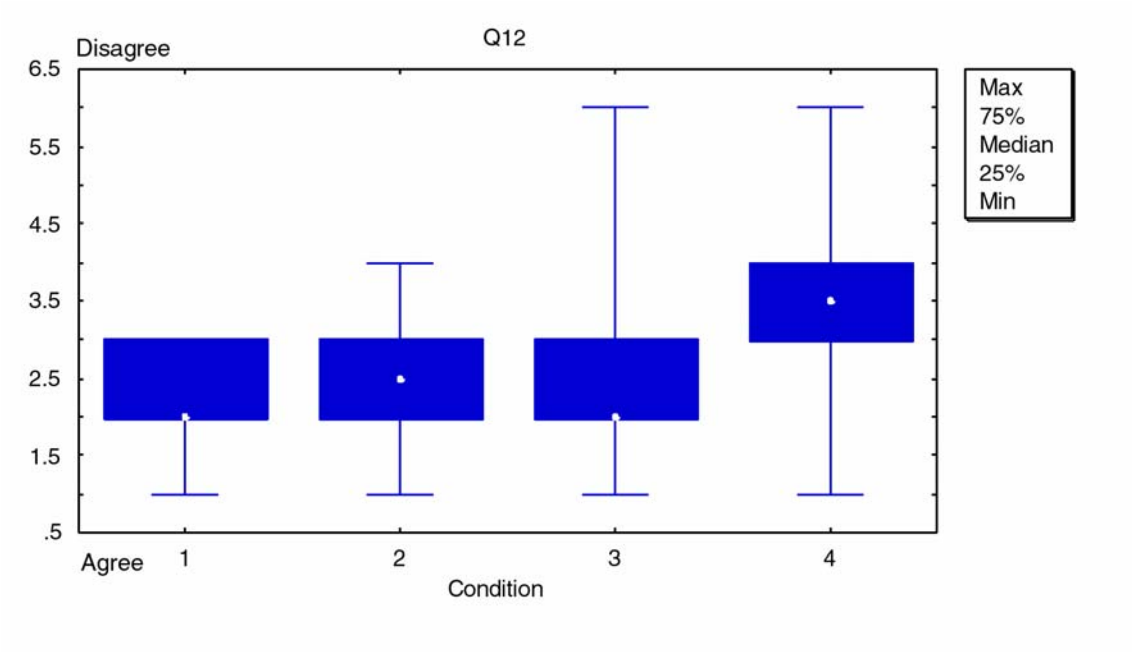


Figure 4.39. Results for Q12. *It was easy to learn how to use the software*

Least easy to learn in Condition 4.

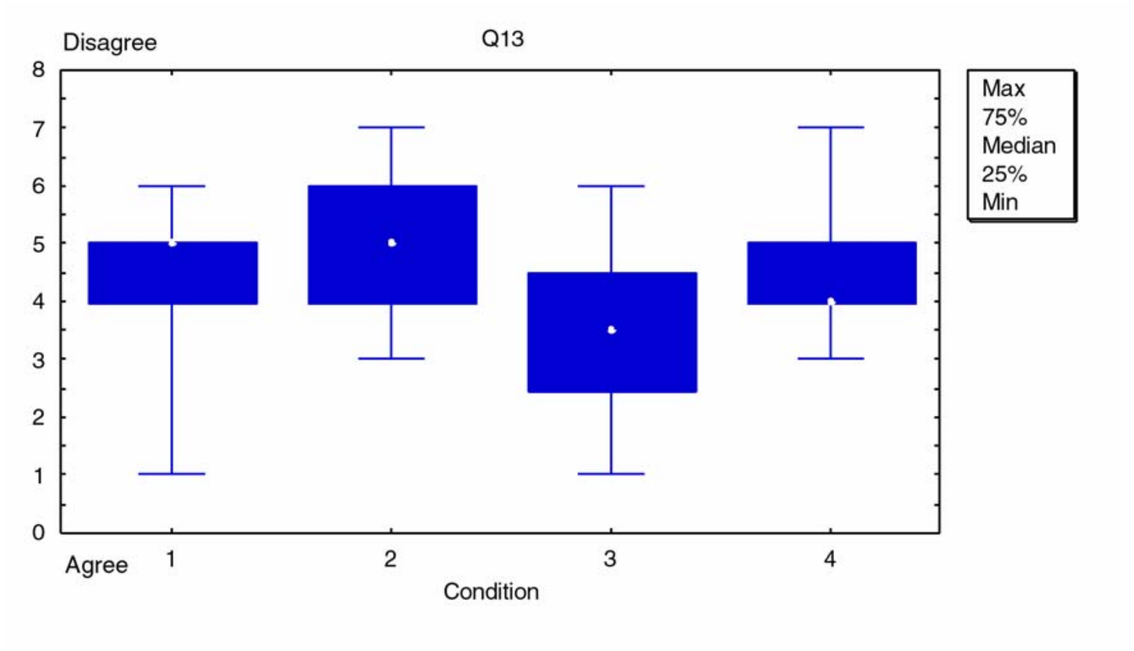


Figure 4.40. Results for Q13. *I found the software friendly to use*

Most friendly in Condition 3.

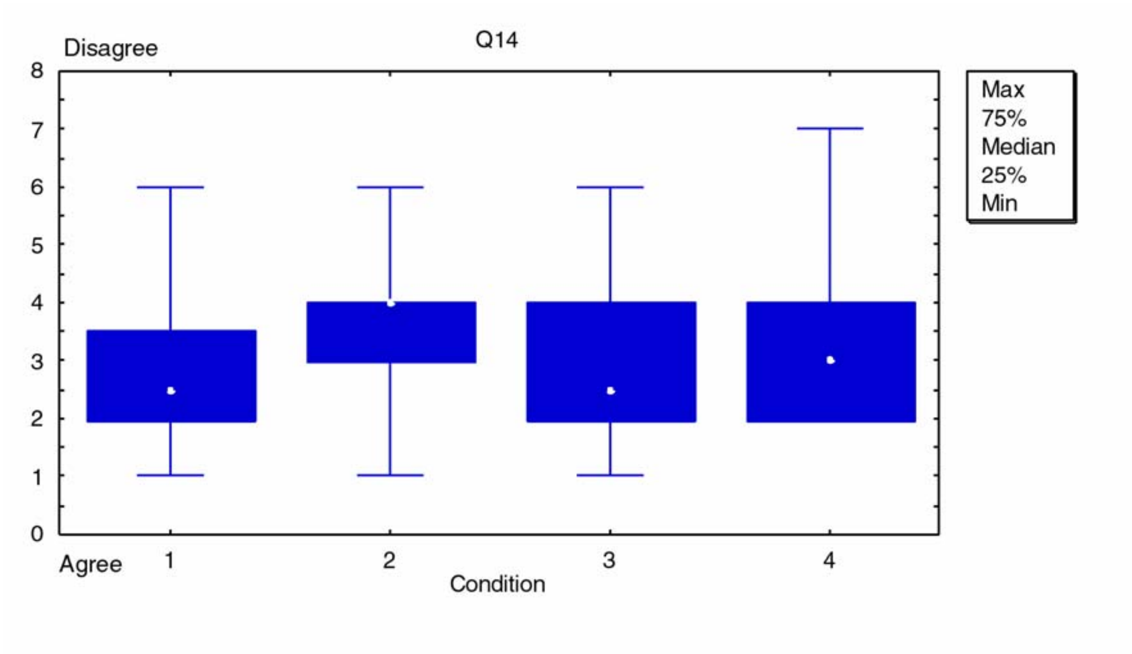


Figure 4.41. Results for Q14. *The software let me decide what to do next*

Less in Condition 2 (Agent led)

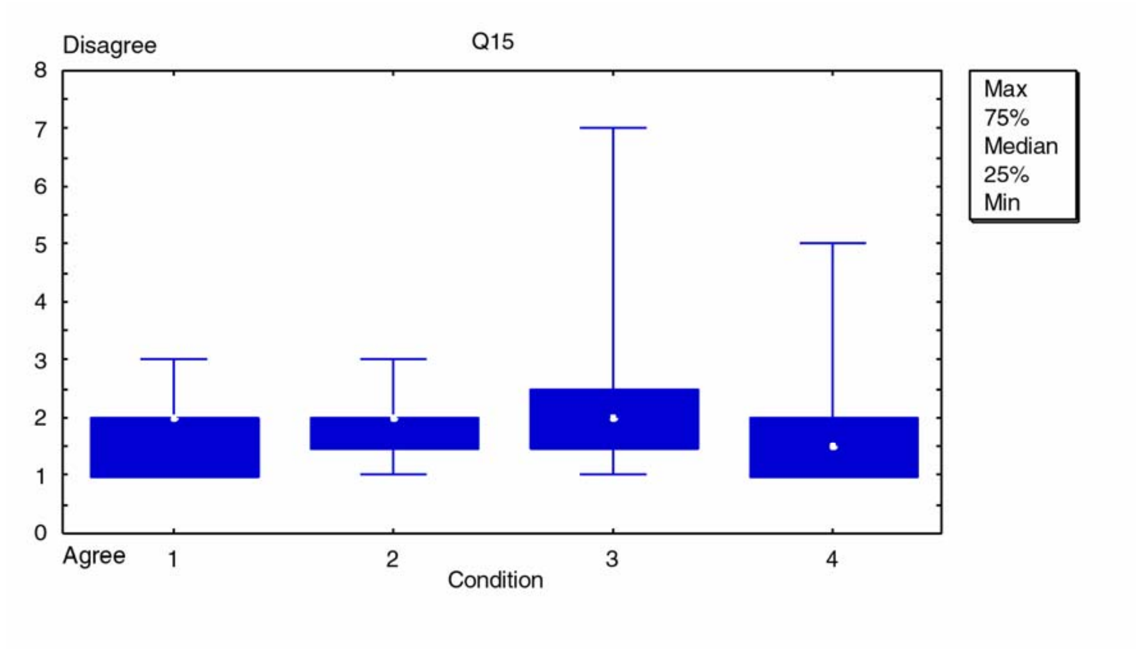


Figure 4.42. Results for Q15. *The software let me work at my own pace*

Little difference except in range of Conditions 3 and 4.

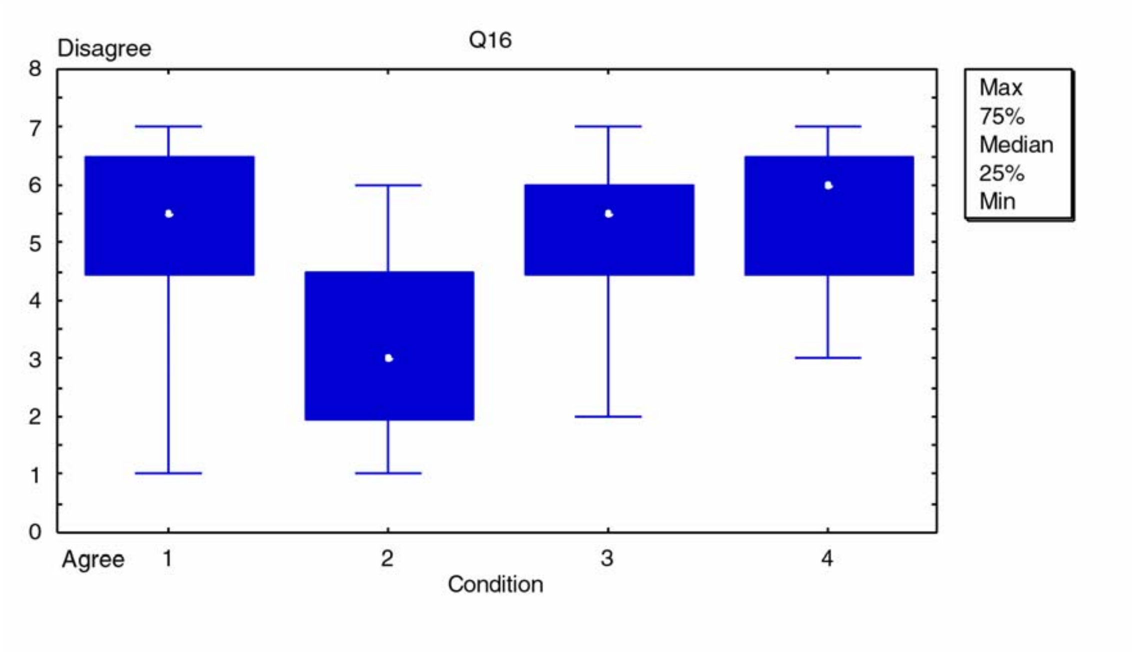


Figure 4.43. Results for Q16. *The software required me to input too much information*

Most information required in Condition 2.

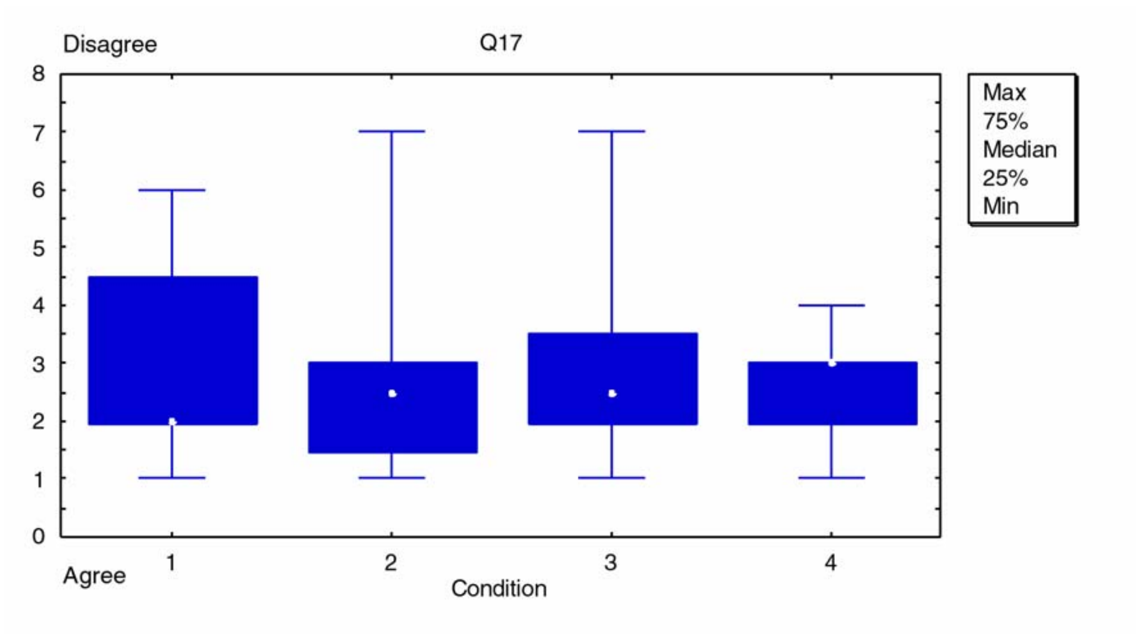


Figure 4.44. Results for Q17. *I understood the terminology used by the software*

Little to choose between conditions.

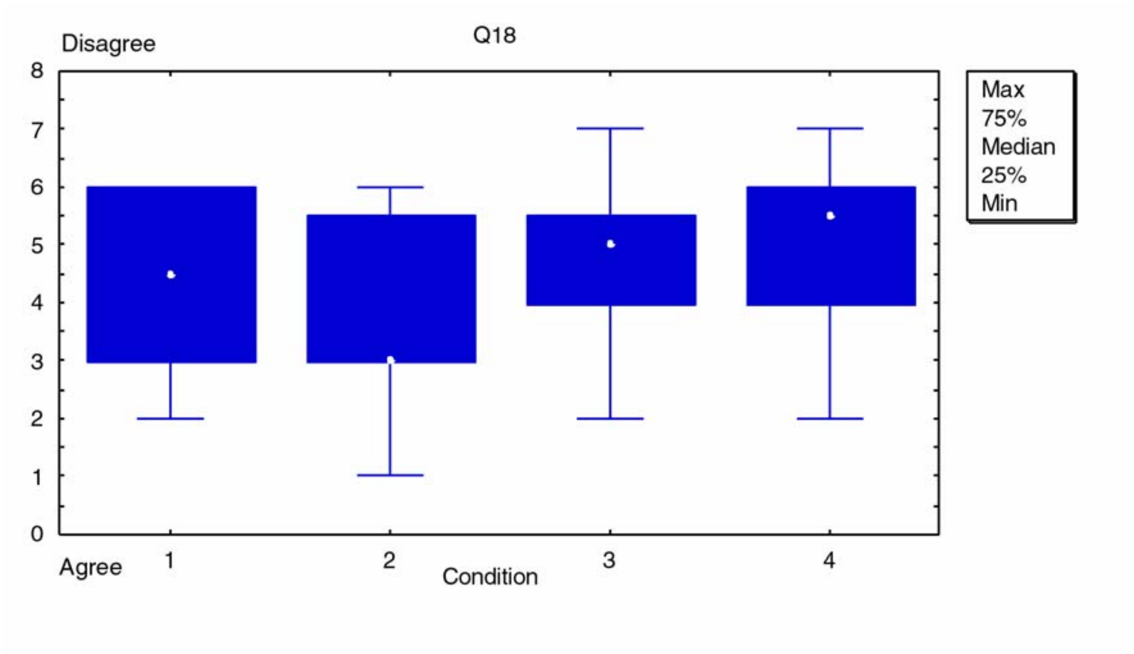


Figure 4.45. Results for Q18. *The software guided me about what action to take* Condition 2 provides slightly more guidance Median, Modes (3.5,6; 3,3; 5,5; 5.5,6) respectively

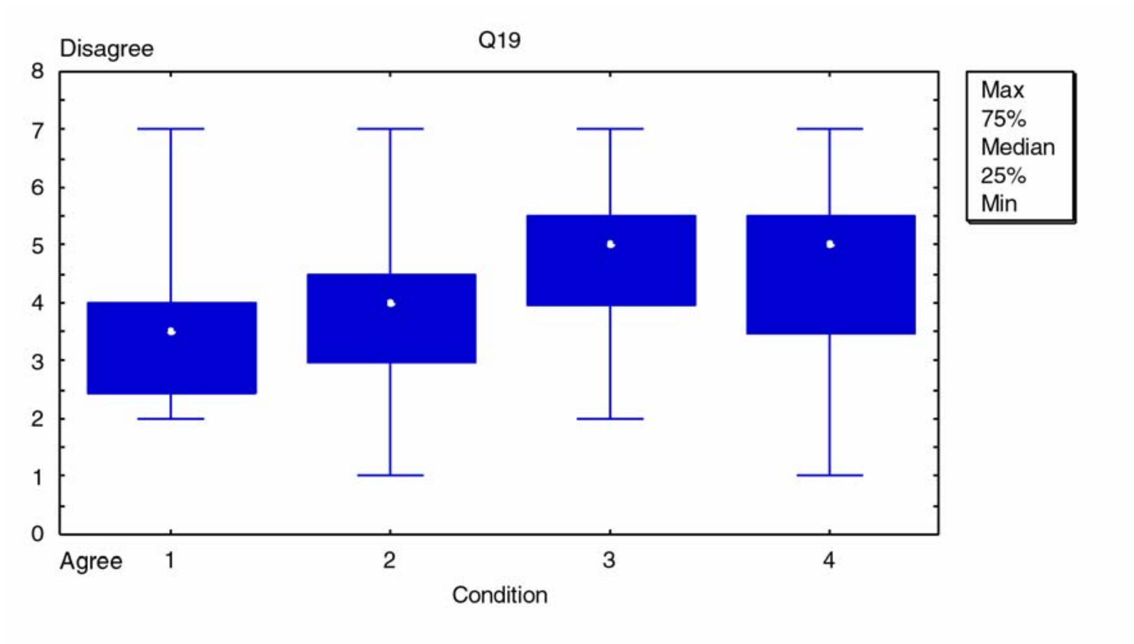


Figure 4.46. Results for Q19. *I found the software frustrating to use* Condition 3 the least frustrating.

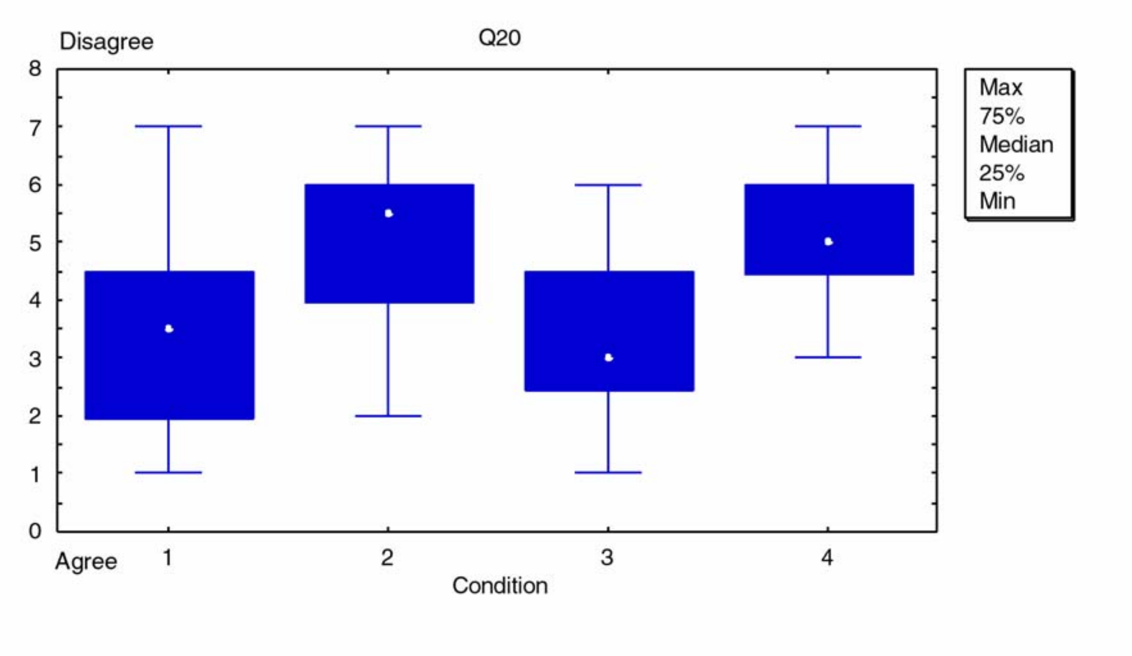


Figure 4.47. Results for Q20. *Overall, I was happy with the software*

Happier in Condition 1 and Condition 3.

4.4.5 Analysis of Results

From forecasting that both a delegative dialogue and aiding with user preferences will facilitate contextualization and therefore improve the quality of interaction, it follows that:

1. Condition 3 (User-Driven dialogue with preferences) should get the best rating;

2. Condition 2 (Agent-Led Dialogue) should get the worst rating;

3 If a delegative dialogue is better than an agent-led dialogue we would expect both Conditions 1 and 3 to have better ratings than 2 and 4;

4 If adaptation based on preferences has an effect on interaction, we should expect Condition 3 to be better than Condition 1 and Condition 4 to be better than Condition 2.

In examining the questionnaire data:

1 Condition 3 generally has the best results (questions 4, 5, 6, 8, 9, 10 and 13).

2 Condition 2 generally has the worst results (questions 1, 2, 3, 4, 8, 13, 14, 16 and 20).

3 Conditions 1 and 3 generally have better results than Conditions 2 and 4. (questions 1, 2, 3, 7, 8, 13, 14, and 20).

4 Condition 3 is generally rated better than or equal to Condition 1.
Condition 4 is generally rated worse than or equal to Condition 2.

Combining these findings, the relative subjective ratings of each condition produce a ranking as follows:

(worst) [4 < 2] < 1 < 3 (best)

The experimental hypotheses for experiments 1 and 2 asserted that:

Experiment 1: A User-Driven dialogue will be more effective than an Agent-Led dialogue,

Experiment 2: A dialogue that adapts to user preferences will be more effective than one which does not.

The results from the questionnaire data generally support these hypotheses. The effect seems to be most pronounced between the use of delagative dialogues (Conditions 1 and 3) versus agent-led dialogues (Conditions 2 and 4). Combining delagative dialogue with adaptation based on user preferences (Condition 3) has the best subjective rating, as would be predicted using contextualization, as this condition provides the greatest set of contextualization cues for the user. Condition 1 has the next best subjective rating, suggesting that dialogue instigation has a greater effect than use of adaptation with preferences (at least in these experiments).

The relative performance of the adaptation with user preferences Conditions (2 and 4) is less clear. The above findings suggest that Condition 4 is generally rated more poorly than condition 2 - the opposite from what would be expected based on contextualization cues. This relative ordering is confirmed by the performance results in which Condition 4 has both more errors and more dialogues than Condition 2, whereas it would be expected that the opposite would be true based on contextualization. From this the conclusion is that users have greater difficulty contextualizing in Condition 4 than the other conditions. There are also some contradictory results in several individual questions. Questions 4, 13 and 18 show interesting discrepancies with the responses to other questions. In Question 4 ("The interface to the software was pleasant") Condition 4 was the most highly rated. In Question 13 ("I found the software friendly to use"), Condition 4 was rated after Condition 3 but before conditions 1 and 2. In question 18 ("The software guided me about what action to take") Condition 2 (generally the worst rated condition) obtained the highest rating and in question 19 ("I found the software frustrating to use") Condition 4 was rated almost as well as condition 3.

One possible reason for the relatively poor performance of Condition 4 and the responses to questions 4, 13, 18 and 19 might be that other factors in individuals also influence the way in which they contextualize data, and this affects the results in these cases. Questions 4, 13, 18 and 19 ask about very subjective impressions of the software, and these may consequently be affected by the *personality* of the user.

4.5 Experiment 3: Social Factors: Personality and Dialogue Instigation

In Experiments 1 and 2 the style of dialogue instigation varied in a similar way to social human roles. One explanation for the variation in performance between conditions may be due to the fact that people with different personalities (e.g., dominant and submissive) may consequently prefer similar or complementary dialogue styles.

The investigation of social factors in human-computer interaction, including the effects of personality (both agent and user) suggest that the form of user-agent interaction may be influenced by preferred dialogue style. Work by Reeves and Nass (1996) addressed this phenomenon concerning the user-perceived computer personality on a dominant-submissive scale, postulating that:

1. People will perceive a computer that uses dominant text as having a dominant personality, and a computer that uses submissive text as having a submissive personality.

2. Users with a dominant personality will report that a dominant computer is more like them than a submissive computer, and users with a submissive personality will recognize a submissive computer as more like them than a dominant computer.

3. Users with a dominant personality will prefer the dominant computer, while submissive users will prefer the submissive computer.
(Reeves and Nass, 1996 pp90-95)

In an experimental evaluation of these hypotheses (Nass, Moon et al., 1995) these dominant-submissive characteristics were represented in a textual interaction in which the following characteristics were manipulated:

1. The phrasing of the text;
2. The confidence level expressed by the computer;
3. The order of the interaction turn-taking;
4. The name given to the computer (to signify character traits).

These characteristics were used to represent the dominant-submissive characteristics of the computer as agent. The results supported the hypotheses, namely:

1. The dominant computer was perceived as significantly more dominant than the submissive computer, and the submissive computer was perceived as more submissive than the dominant computer;

2. Subjects preferred to interact with a computer with a similar personality to themselves;

3. Subjects matched with similar computers found the interaction to be more satisfying.
(see also Moon and Nass, 1996a,b)

The important concern here is the dialogue style, which is effectively represented by the characteristics chosen by the experimenters to emphasize dominance and submissiveness. Nass, Moon *et al*. used these variables in combination and at their extremes to maximise their effect. It is obvious that these variables are components of contextualization in both agent and user, however, when a non-textual interaction dialogue is used the question remains as to whether the computer (or agent in this case) will still be perceived as having dominant-submissive personality characteristics, and what effect this has on the interaction.

Typically agent dialogues are implemented using a forms-structured approach, as this supplies additional contextual constraints in terms of describing the data which is required to complete a given task. The findings related above suggest that even this minimal set of cues will affect contextualization and consequently user behaviour. It is therefore necessary to address the impact that these personality factors may have in explaining the preferences results found in Experiment 1.

4.5.1 Methodology

It was neither practical nor feasible to administer the same test used by Nass and Moon, as the test used (BSRI; Bem 1974) is lengthy and measures a wide range of personality factors not directly relevant to this study (the test had already been administered to an entire psychology class as part of their undergraduate studies). In this work the definitions of dominance and submissiveness which are readily applicable are as follows (Kiesler, 1983):

Dominant behaviour is characterized by commanding and directing others to take certain actions.

Submissive behaviour avoids this tendency, and is characterized by a propensity to let others make decisions, be easily led and avoid responsibility.

In Experiments 1 and 2 the Keirsey Temperament Sorter (Keirsey and Bates, 1984), a modified Myers-Briggs test, was first administered to all participants. Although Keirsey introduces a different nomenclature for the 4 bipolar personality scales, the questionnaire also gives the more widely-used Myers-Briggs traits. The 4 traits measured by the test are:

Extraversion-Introversion
Sensation-Intuition
Thinking-Feeling
Judgement-Perception

Of particular interest here is the Extraversion-Introversion trait. This measures the degree to which a person takes the initiative in interacting with other people. In the personality assessment questionnaire (Appendix A4) this scale is represented by questions such as:

Q7 Do you consider yourself *a good conversationalist* or *a good listener?*

Q9 At a party do you *interact with many, even strangers* or *interact with a few friends?*

Q10 Waiting in line do you often *chat with others* or *stick to business?*

Comparing these factors with dialogue instigation (Section 3.5) it might be speculated that there is a relationship between the Extraversion-Introversion personality trait and the performance and preference of subjects between user-driven and agent-led dialogues. i.e., Extraverts prefer to instigate or take the initiative in initiating dialogue, whereas Introverts prefer others to initiate dialogue. This would be equally true of dialogue between people and dialogue between a person participating in a dialogue with a computer system or software agent. In this way the Extravert-Introvert Myers-Briggs trait may be taken as comparable to the Dominant-Submissive scale used by Nass and Reeves. As a parallel the delagative style of dialogue in which the user drives interaction may be viewed as "introverted" and the agent-led dialogue as "extraverted." This relationship may influence the way in which dialogue instigation is rated by users, as subjects possessing different personality traits may consequently prefer different styles of dialogue instigation (either matching or complementing their personality).

To examine any interaction between personality and dialogue style the allocation of subjects in Experiments 1 and 2 was balanced by the results of the Keirsey test on the Myers-Briggs Introvert-Extravert trait. The format of the measurement provided by the questionnaire is a letter (the orientation on each bipolar trait scale) followed by a number, representing an indication of the degree (or strength) of the trait. e.g., I+6, N+2, T+4, J+1 represents a personality with the traits Introversion (degree 6), iNtuition (degree 2), Thinking (degree 4), Judgemental (degree 1). All traits and their degrees were recorded for each subject (Appendix B3)

4.5.2 Experimental Conditions

The participants in the User-Driven and Agent-Driven conditions of Experiment 1 were asked to complete personality tests to ascertain personality attributes which might affect dialogue style preferences. These tests were completed before any of the experimental trials in both Experiment 1 and Experiment 2. The results were used to balance subjects between conditions.

From Nass, Moon et al (1995) the two conclusions of their work of interest are:

1. Subjects preferred to interact with a computer having a similar personality to themselves;

2. Subjects matched with similar computers found the interaction to be more satisfying.

However, in Isbister and Nass (in press) the second finding is contradicted, and here it was found that participants preferred a character whose personality was complementary, rather than similar to their own.

The questions that are of interest are therefore:

1. Is there a correlation between personality and the quality of interaction (Objective and Subjective)?

2. Is there a preference for one dialogue style over another, taking into account user personality?

It is also possible to compare both subjective responses (questionnaire data) and performance data with personality traits. Possible correlations between performance data and personality have not been made in any previous studies.

4.5.3 Data Collection

The questionnaire was administered prior to subjects participating in Experiment 1 and Experiment 2. The questionnaire was a self-scoring web-based questionnaire that allowed subjects to see their personality traits before submitting them to the experimenter. This procedure was adopted as a result of recommendations of the National Research Council of Canada's Ethics Committee on review of the experimental procedure, as it enables subjects to withdraw from the experiment without disclosing their personality scores (no subjects withdrew from the experiments).

4.5.4 Results

For each subject's score on the Introvert-Extravert trait was recorded as both a categorical response (I or E) and as point scale value from –10 (extravert) to +10 (introvert). The data are given in Appendix B3.

4.5.4.1 Performance Effects

No other studies have examined objective (performance) factors coupled with personality, concentrating instead on purely subjective factors. The data collected in the 3 experiments gives the opportunity to explore links between both objective and subjective data and personality traits.

In the 4 experimental conditions there are 2 User-Driven conditions (1 and 3), and 2 Agent-Led conditions (2 and 4). By grouping these conditions together it is possible to compare the response of subjects by personality in User-driven and Agent-led conditions.

	Time	Dialogue	Broken Dialogues	Log(time)	E-I
Time	**1**	**0.25667**	**-0.19482**	**0.98674**	**0.03666**
Dialogue	**0.25667**	**1**	**0.28521**	**0.24574**	**-0.27426**
Broken Dialogues	**-0.19482**	**0.28521**	**1**	**-0.22855**	**-0.21315**
Logtime	**0.98674**	**0.24574**	**-0.22855**	**1**	**0.07079**
E-I	**0.03666**	**-0.27426**	**-0.21315**	**0.07079**	**1**

Table 4.11. Correlation between variables in Conditions 1 and 3

	Time	Dialogue	Broken Dialogues	Log(time)	E-I
Time	**1**	**0.45028**	**0.12301**	**0.98541**	**0.05627**
Dialogue	**0.45028**	**1**	**0.68706**	**0.48653**	**0.26619**
Broken Dialogues	**0.12301**	**0.68706**	**1**	**0.18255**	**0.20261**
Logtime	**0.98541**	**0.48653**	**0.18255**	**1**	**0.05379**
E-I	**0.05627**	**0.26619**	**0.20261**	**0.05379**	**1**

Table 4.12. Correlation between variables in Conditions 2 and 4

The *Statistica* statistical analysis software was used to examine correlation (*Spearman R*) between performance data and personality trait score (+10 to –10 on a unit integer interval). Table 4.11 Shows the data for Conditions 1 and 3 (User-Driven) and Table 4.12 the results for Conditions 2 and 4 (Agent-Led).

For Time and Log(Time) there is no correlation in either table. It may be concluded that there is no correlation between total task time and personality.

The correlations between *number of dialogues* and *broken dialogues* is more interesting. Although the correlation is low in each case, the sign of the correlation reverses between conditions. This can be more clearly seen in Figures 4.48 and 4.49 (dialogues) and Figures 4.50 and 4.51 (broken dialogues).

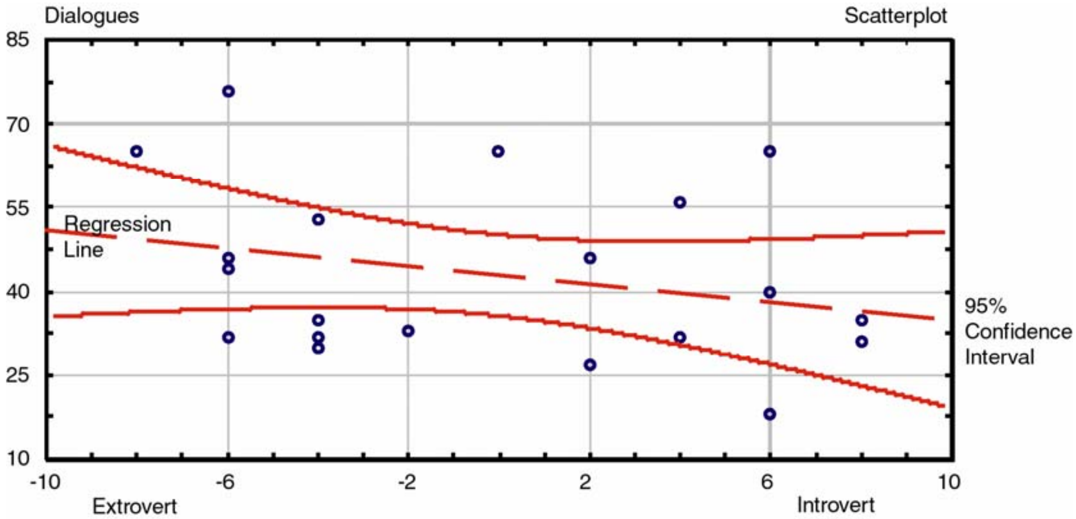


Figure 4.48. Correlation between Extravert-Introvert and number of dialogues in Conditions 1 and 3

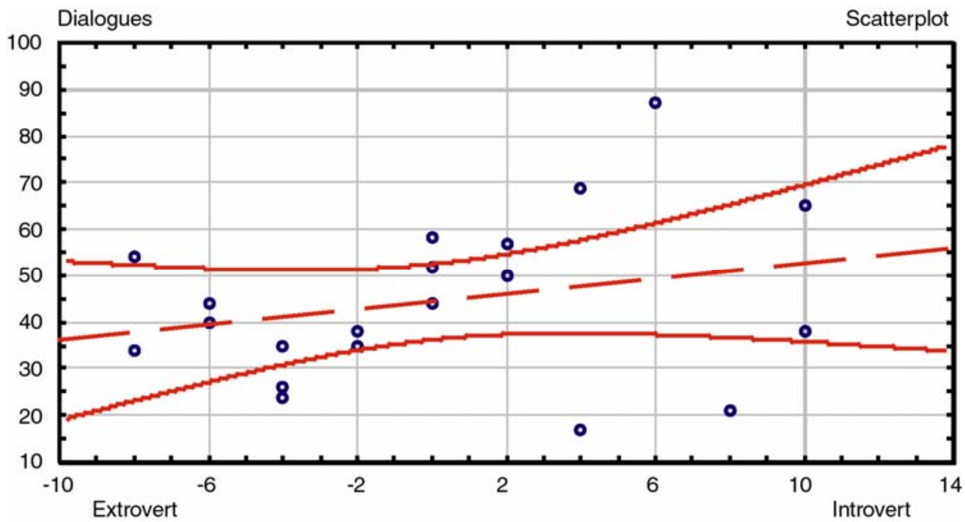


Figure 4.49. Correlation between Extravert-Introvert and number of dialogues in Conditions 2 and 4

Observing the slope of the regression line in each case it can be seen that introverts tend to have fewer dialogues than extraverts in the User-driven (delagatory) conditions (1 and 3), and extraverts have fewer dialogues than introverts in the Agent-Led conditions (2 and 4). From this it would seem that users prefer dialogue instigation that matches their own personality in terms of introversion and extroversion.

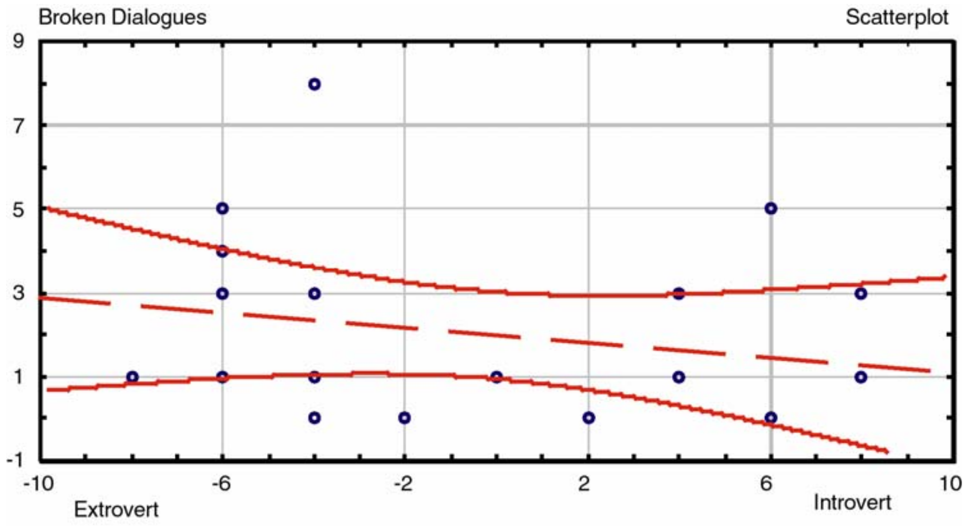


Figure 4.50. Correlation between Extravert-Introvert and number of broken dialogues in Conditions 1 and 3

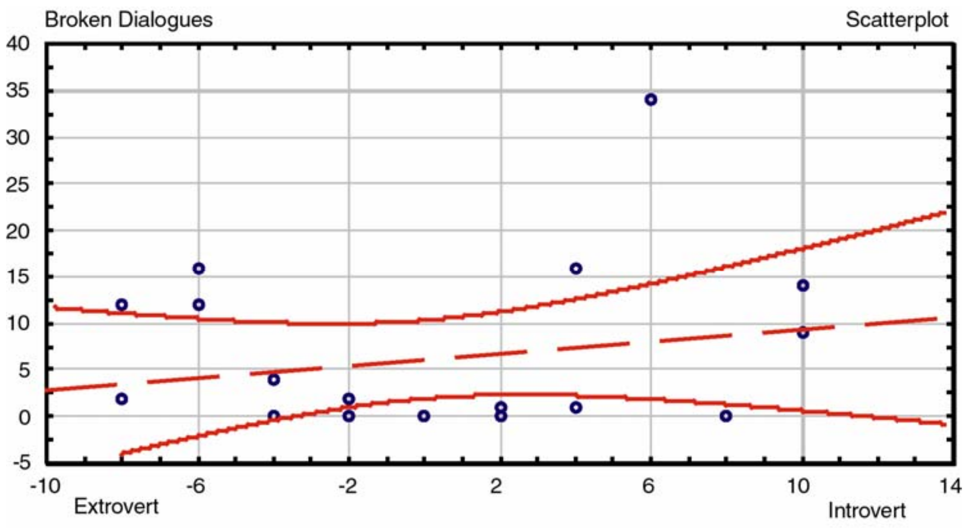


Figure 4.51. Correlation between Extravert-Introvert and number of broken dialogues in Conditions 2 and 4.

In the case of broken dialogues (errors), the slope of the regression line shows that introverts tend to have fewer broken dialogues than extraverts in the User-Driven (delagatory) conditions (1 and 3), and extraverts have fewer broken dialogues than introverts in the Agent-Led conditions (2 and 4). From this it would seem that users prefer dialogue instigation that matches their own personality in terms of introversion and extroversion in terms of both the number of dialogues and errors made.

However, the data are unreliable, as there is a wide spread of data points, and many lie outside the 95% confidence interval. Substantially more subjects are required to verify the conclusions drawn above (this is further discussed in Chapter 5).

4.5.4.2 Subjective Effects

There are effectively 8 conditions (4 experimental conditions each divided into the results for Introverts and Extraverts) with 5 subjects per condition. Because of the small number of subjects per condition, overall questionnaire response in each condition was again analyzed to find mean, median and mode (i.e., the overall mean, median and modes of the questionnaire mean, median and modes for each condition were calculated). Note that because of the phrasing of questions 16 and 19, they are scored in the opposite way to the other questions. The results are shown in Table 4.13 (lower scores are better) and Table 4.14 (higher scores are better for questions 16 and 19).

	1I	1E	2I	2E	3I	3E	4I	4E
mean	2.79	4.13	3.65	3.90	2.91	3.80	4.23	3.79
median	2.5	4	3	4	2	3.5	4	4
mode	2	6	6	3.5	2	3	3	-

Table 4.13. Average question scores in each condition (Questions except 16 & 19)

	1I	1E	2I	2E	3I	3E	4I	4E
mean	3.67	5.1	3.1	4.25	5.7	4.1	4.7	5.2
median	3.5	5	2.5	4.25	5.5	4	5.5	5.5
mode	-	-	-	-	-	-	-	-

Table 4.14. Average question scores in each condition (Questions 16 & 19)

4.5.4.2.1 User Driven Conditions

In Table 4.13, given the previous results in terms of performance, it would be expected that introverts score user driven interaction (Condition 1) lower (or better) than extraverts, and introverts score user driven interaction with preferences (Condition 3) lower (or better) than extraverts.

In Condition 1, it can be seen that introverted subjects give lower (better) mean, median and mode responses than extraverts, and in Condition 3 the same is also true.

From Table 4.14 (questions 16 and 19) it would be expected that introverts score user driven (Condition 1) higher (or better) than extraverts, and introverts score user driven with preferences higher (or better) than extraverts.

In Condition 1 the data show that introverts score lower (worse) than extraverts. However, in Condition 3 introverts score higher (better) than extraverts.

4.5.4.2.2 Agent-Led Conditions

It would also be expected that in Condition 2 and Condition 4, extraverts would have a lower (better) score than introverts (Table 4.13). In the same way, the results for questions 16 and 19 (Table 4.14) would be expected to show that extraverts would have a higher (better) overall score in these conditions.

In Condition 2, extraverts have a higher mean and median, but a lower mode score than introverts. In Condition 4, extraverts have lower mean and median scores (there is no mode for condition 4E).

For questions 16 and 19 it can be seen that extraverts do have better scores (mean, median and mode) than introverts.

Overall this data suggests a correlation of performance with the introvert-extravert personality trait. This may be summed up as *introverts prefer a User-Driven dialogue to an Agent-Led dialogue, and extraverts prefer an Agent-Led dialogue to a User-Driven dialogue.*

In conclusion subjects prefer a dialogue style which is similar to that of their own personality. (The only contradictory evidence is that in the rating of Condition 1 for questions 16 and 19. Because these questions are scored in an opposite fashion and constitute 10% of the questionnaire data, they do not represent compelling evidence to the contrary.)

4.5.4.2.3 Effect of Personality on Specific Questionnaire Results

In Experiment 2 there were some contradictory results concerning Questions 4, 13, 18 and 19, and one of the reasons suggested for this was that responses were influenced by the personality of the user.

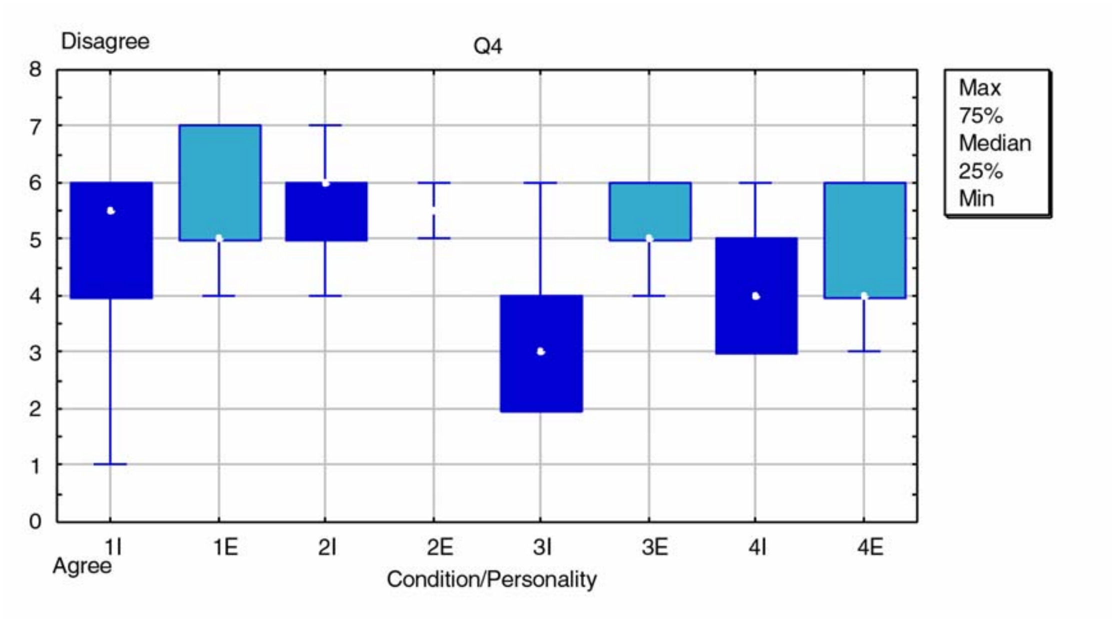


Figure 4.52. Responses in each condition by personality for Q4

Examining the responses to Question 4 "The interface to the software was pleasant" (Figure 4.52), it can be seen that although there was little difference between responses in Conditions 2 and 4 (Agent-Led), Conditions 1 and 3 (User-Driven) show a difference (particularly in Condition 3) with introverts agreeing with the question more than extraverts. However, in combining the scores of introverts and extraverts this difference is averaged out, masking the effect.

In Question 13 ("I found the software friendly to use") there is a similar division between personality conditions (Figure 4.53). Here introverts clearly preferred Condition 3 in comparison to extraverts (and to a lesser extent, the same is true of Condition 1). Condition 4 was rated similarly by both introverts and extraverts, the effect of using preferences in the agent-led dialogue seeming to have little effect.

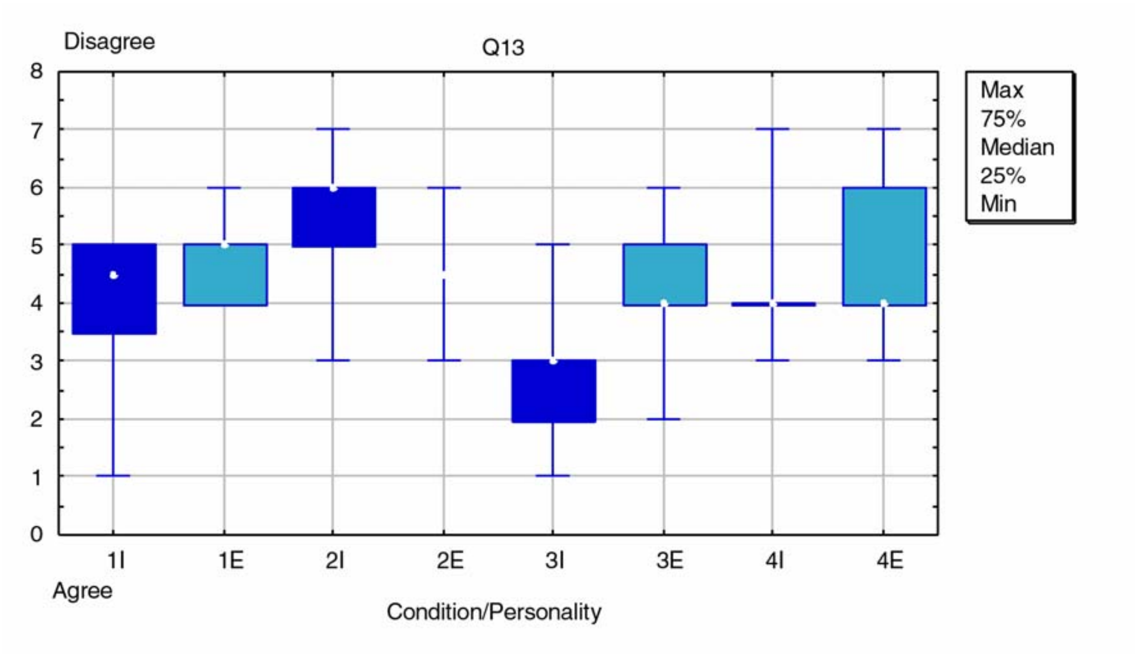


Figure 4.53. Responses in each condition by personality for Q13

In Question 18 ("The software guided me about what action to take") it can be seen that introverts agreed with the statement more than extraverts in condition

2, and more than introverts and extraverts in other conditions (Figure 4.54). If "guided" is taken as synonymous with "aided" or "helped" then this is a suprising result. One explanation is that introverts interpret the statement (and the process of being guided) in a similar way to being constrained or manipulated, hence the level of agreement.

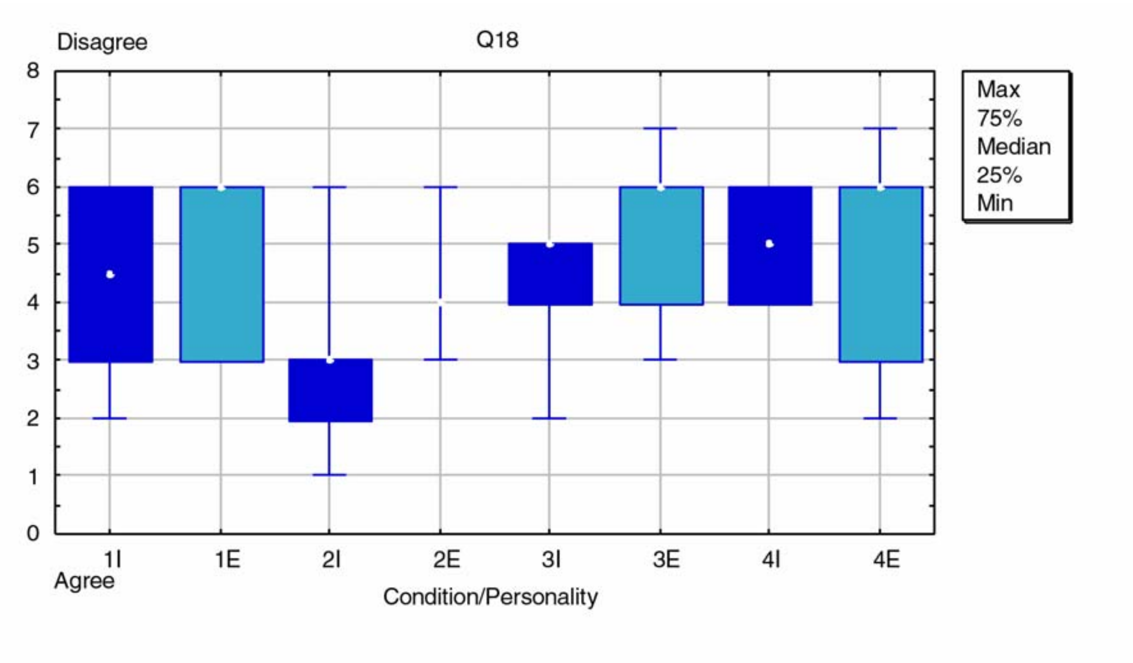


Figure 4.54 Responses in each condition by personality for Q18

In Question 19 - "I found the software frustrating to use" – the difference between introvert and extravert responses between Conditions 3 and 4 becomes clear.

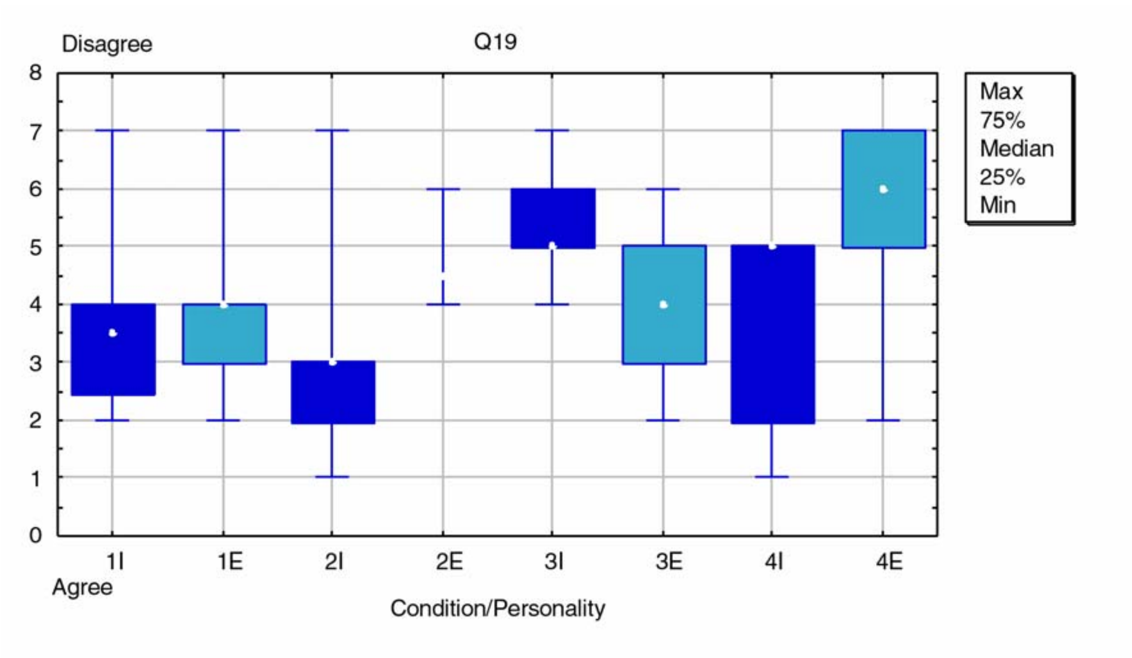


Figure 4.55 Responses in each condition by personality for Q19

Extraverts found Condition 4 less frustrating than introverts, but in Condition 2 the opposite is true (as we would expect if subjects prefer a dialogue style that matches their personality). When the differences are combined (as in Experiment 2) these differences average out making Conditions 3 and 4 rated almost equally overall.

4.6 Summary of Experimental Results

This chapter empirically evaluated several components of the model of contextualization developed in Chapter 3. These components concerned the use of dialogue styles (Experiment 1), the adaptation of interaction by means of user preferences (Experiment 2) and the influence of social cues and personality on behaviour (Experiment 3).

In Experiment 1 the effectiveness of two styles of dialogue instigation and control was examined. A User-driven dialogue and an Agent-led dialogue were compared in an experimental setting to examine any differences in user performance or preference. The model of contextualization suggested that a dialogue that was driven by the user in a delagatory fashion would be better than a dialogue that was led by the software (or agent). The results form this experiment confirmed the predictions. The user-driven condition was significantly ($p<0.05$) better than the agent-led conditions in terms of total time taken to complete the task. In confirmation of this, subjective feedback from the users rated the user-driven condition consistently better than the agent-led condition.

In Experiment 2, the effect of adapting the interface by making use of user preferences was examined in combination with the dialogue styles from Experiment 1. Predictions using the model of contextualization suggested that such adaptation would improve interaction. This was confirmed by experimental results, though at a lower level of confidence than Experiment 1 (a quantitative analysis of the data showed no significant difference in performance between conditions). A qualitative examination of the data confirmed predictions about the relative performance between conditions with the exceptions being the lack of an improvement (decrease) in total task time for Condition 3 over Condition 1, and a poorer performance in terms of number of dialogues and number of errors between Condition 2 and Condition 4. Subjective feedback from users via the questionnaire showed conditions ranked as:

3 (best), 1, 2, 4 (worst)

This is consistent with contextualization predictions except for the relative rankings of Conditions 2 and 4. There were also anomalous results for 4 of the questions.

In Experiment 3 the effect of personality as a component of contextualization and its effect on interaction were examined. From these results there appears to be a relationship between the number of dialogues and the Introvert-Extravert personality trait and the number of broken dialogues and the Introvert-Extravert trait that suggests that users perform better when interacting with a system with a similar personality to their own (in terms of dialogue instigation). This was confirmed by subjective data from the questionnaire. The questionnaire data also explained some of the anomalies in Experiment 2 by breaking down question responses by personality trait and examining responses in light of this.

In summary, the experiments validated the components of contextualization, although the results varied in significance and some results merit further discussion. These issues will be addressed in the next chapter.

Chapter 5

Analysis, Discussion and Future Work

5.0 Experimental Results and Contextualization

Following the results of the experimental investigation, there are several findings that require further discussion. This chapter addresses these findings and discusses their implications in terms of contextualization and possible future work.

5.1 Condition Ordering

Using a model of contextualization, the expected ordering of conditions in performance terms should have been 3 (best), 1 or 4 (in either order) and finally 2 (worst). This reflects that two components of contextualization are built into Condition 3 (user-driven with preferences), one component in Condition 1 (user-driven dialogue) and Condition 4 (adaptation with preferences), and none in Condition 2. At the outset of the experiment the relative effect of user-driven dialogue and adaptation could not be predicted and consequently the relative ordering of conditions 1 and 4 was unknown. The size of the effect that each component of contextualization would have on interaction was also unknown.

From the experimental results (see Figures 4.24 - 4.27) there is only a significant difference in time between Conditions 1 and 2, and a poorer than expected total time performance in Condition 3 (i.e., condition 3 had a larger mean total task time than Condition 1). However, Condition 4 shows a *decrease* in total task time over Condition 2, implying that use of interface adaptation via user preferences does reduce total task time. However, this does not seem to be the case between Conditions 1 and 3, so some other factor in Condition 3 may be affecting total task time.

Comparing the results for number of dialogues and number of errors it can be seen that the number of dialogues increases slightly in Conditions 3 and 4. The introduction of adaptation by user preferences therefore seems to *increase* the average total number of dialogues.

Comparing the number of errors in each condition, there are slightly more broken dialogues (and a higher standard deviation) in Conditions 2 and 4. From this it may be concluded that the agent-led dialogue conditions *increase* the propensity of dialogue errors.

Using a model of contextualization to predict performance, Condition 4 would be expected to have more dialogues and more broken dialogues than Condition 2, and this is confirmed by the results. However, the average task time is *less* for Condition 4 than for Condition 2, and Condition 4 is ranked after Condition 2 in the results of the questionnaire.

The difficulty in comparing the relative advantages of Conditions 2 and 4 and the result that Condition 3 is not better than Condition 1 suggest a more complex interaction between contextualization components than was originally anticipated.

The use of both a user-driven dialogue and adaptation of the interface using user preferences (Condition 3) should provide a lower total task time than use of user-driven dialogue alone (Condition 1), but does not. In statistical terms this (and the lack of significance for other performance measures) may be due to too few experimental subjects in each condition. The use of adaptation of the interface may increase the average number of dialogues but does not necessarily seem to increase overall task time (e.g., this is not the case comparing Conditions 2 and 4). Consequently there may be contextualization factors responsible for the time disparity of Conditions 1 and 3.

Allowing for the greater variation in error in the agent-led dialogue conditions of Conditions 2 and 4, it is possible that the difference in average total task times is actually a product of the effects of contextualization. Returning to the evaluation of adaptive and intelligent user interfaces (Section 2.5) it was noted that the inconsistency generated by unexpected interface adaptation often worsened rather than improved performance. The manifestation of changes in interface behaviour due to adaptation using user preferences without the added effect of delegative dialogue may increase the contextualization overheads for the user, with a consequent increase in task time. This would seem to be confirmed by the subjective findings of the questionnaire, which also rates Condition 4 more poorly than Condition 2. This in turn suggests that the style of dialogue used produces a more consistent and predictable effect than adaptation of the interface. From the perspective of using contextualization to design an interface, the implication is that it is important to choose a delegative dialogue style, and then design interface adaptation around this. It is possible that adapting the user interface dynamically influences the dialogue style perceived by the user (Section 4.3). Thus the components of the model of contextualization cannot be applied in isolation, and should not be taken as additive.

The increase in total number of dialogues per task in Conditions 3 and 4 may be due to the overhead of completing user preferences in relation to the overall task time. A longer task (more complex task) may reduce this apparent effect and give a result more in line with the model of contextualization.

One final issue is that the model of contextualization predicts that a delegative, user-driven dialogue will be most effective based on social models of interaction. The effect of other contextual factors (such as unidentified border resources) and personality may also influence the effectiveness of this style of dialogue over others, depending on the user's personality, etc.

5.2 Effects of Personality

The analysis of performance and subjective responses with respect to personality traits provided some insights into the results of earlier experiments. Although there was a relatively weak correlation between dialogues and broken dialogues with introvert-introvert personality trait (and a relatively large number of data points lay outside the 95% confidence interval), the significance in the results is due to the reversal in the slope of the regression line between dialogue conditions. The conclusion that people like to interact with a system that has a similar personality to their own (in terms of extroversion and introversion) may be contrasted with earlier findings in the literature.

Moon and Nass (1996) found that using a dominant-submissive personality trait, people preferred interacting with a computer that had a similar personality to their own, a finding which is predicted by the similarity-attraction association in the personality literature (e.g., Byrne and Nelson, 1965). However, in later work (Isbister and Nass, in press) these results were reversed (subjects preferred interacting with a character that had a complementary personality to their own).

Firstly it is important to note that the results of Chapter 4 encompass both objective and subjective results, and that both these results suggest that people prefer to interact with a system which has a similar personality to the user. In the work of Moon and Nass (1996) and Isbister and Nass (in press) only subjective effects are studied, and there are many independent variables that are used to manipulate the apparent personality of the computer. Moon and Nass (1996) manipulated a number of variables to give a non-personified computer "personality" (see Section 3.7.2). In Isbister and Nass (in press) the user's judgement of personality is based on interaction with computer generated *characters*. These characters are based on findings from both the art and psychology literature, with the motivation of being able to "take advantage of natural human social affordances" in interaction. The reversal of their earlier findings may therefore be due to the unpredictable interactions seen between the use of dialogue style and other user preferences in these experiments. This seems to be confirmed by both objective and subjective measures in confirming similarity attraction in Experiments 1, 2 and 3.

5.2.1 Other Personality Traits

Because of the likelihood that the Introvert-Extravert personality trait would interact most with dialogue instigation, subjects were balanced in conditions by this trait. The distinction between introvert and extravert is usually made on the basis of sociability – the extravert has as propensity to instigate interaction with others, whilst the introvert tends to prefer a more solitary, less interactive environment. However, the Myers-Briggs personality model has 3 other axes that may also effect performance and subjective impressions of software behaviour.

The Sensation (S) – Intuition (N) trait distinguishes a preference for experience (S) rather than theory (N). People who prefer sensation tend to value experience and want to be realistic. People who prefer intuition value hunches and tend to be speculative. Considering the implications for interaction, this may effect the way in which users approach an interface. Those valuing previous experience (S) may expect an interface to be consistent with previous systems, while those of an intuition type may hypothesize interaction behaviour and expect the interface to conform to these expectations. In Experiment 2 performed in Chapter 3, the questionnaire predictability ratings (Q9, Q11) show that there is a wide range between maximum and minimum ratings for predictability, suggesting that the effect of this personality trait may be worth further study.

People who make choices on an impersonal basis (e.g., according to a set of objective factors) are termed Thinking (T) while those who choose a more personal, subjective basis for decision-making are termed Feeling (F). When viewing interaction, this trait may influence the rating of an interface or interaction based on differing criteria. A person of a Thinking type may base their rating of a system on relatively objective, impersonal factors (such as interfaces they have used before, or design guidelines). However, people of a Feeling type may base their evaluation more on subjective factors such as artistic and unique features, quirkiness and so on. This obviously has implications for the design of dialogues and the rendering of the interface. In the experiments performed in this thesis the interface was designed to conform to standard windows-based programs, and the rating of the interface may consequently differ between people of thinking and feeling types.

People who choose closure over open options are likely to be Judging types (J), whereas people who prefer to keep their options open and fluid are termed Perceiving (P). One possible effect of this trait might relate to the number of possible choices that are available at any given point in an interaction. A Judging type may prefer a relatively constrained set of interface options, but a perceiving type may prefer greater freedom in deciding what to do next. In the experiments conducted in Chapter 3 the interface was constrained to some extent by the use of a task model, as the model of contextualization suggests that this will aid contextualization by 'signposting' the sequence in which things should be done. The open question for future research is whether this provided uniform performance and preference advantages when examined by this personality trait.

The partitioning of personality into these 4 traits give 16 possible personality types each of which may consequently rate a system in a different way. In the discussion of trait-based personality theory (Section 3.11.1) it was noted that some trait-based models of personality consist of tens or hundreds of traits, making the interaction between traits and the modeling of their effects very difficult. One possible option would be to analyze the proportion of personality types in the general population and design systems based on catering to the majority of the population (with the possibility that some users really dislike the system!).

Unfortunately, balancing all 4 personality traits between conditions would require more time and subjects than were available in the scope of this work. In the experiments presented here there was too little data to investigate interaction between traits, and having balanced subjects on the Introvert-Extravert trait, there was little correlation on other traits between conditions. However, it would seem that the development of a trait-based personality model designed specifically to aid contextualization is a potentially profitable area of future research.

5.3 Experimental Methodology

An analysis of the spread of data points in Figures 4.50 and 4.51 suggests that too few subjects were used to draw conclusive results from the data. This is reinforced by the lack of significance and large variance seen in Experiment 2. Unfortunately time and resources to increase the number subjects were beyond the scope of the present work. Hopefully this study provides a baseline indicator of sample size and issues in methodology that future work will be able to build upon.

5.4 Contextualization as a Design Model

The objectives underlying the development of a model of contextualization were to address the design of agent-based interfaces as a way for managing complexity. This encompasses a change in the interaction paradigm used, as feedback becomes dialogue and automation becomes agency. In addition, the model of contextualization emphasizes that in the design of a computer system there are a number of *border resources* that may influence interaction, including social issues such as the personality of the user (e.g., as part of the user preferences).

The model of contextualization presented in Chapter 3 was devised to encompass the components necessary for a system to promote contextualization in interaction. In this way the model provides both a prescriptive design model (in terms of components that should be considered) and a descriptive analysis model (concerning the relationship between components). The experimental part of this thesis assessed the effects of dialogue instigation and adaptation via user models as components of a contextualizing interface, and the interaction between model components. A number of design rules become clear for these components from examining the results of these experiments:

1. In combination with an appropriate style of dialogue, interface adaptation can improve subjective user impressions of interaction. However, adaptation of the interface may increase the number of dialogues required to complete a task, and in some instances may increase the apparent complexity of an interface, with corresponding decreases in performance.

2. An Agent-Led dialogue style increases the likelihood of broken dialogues or dialogue errors in an interaction. It is, however, the preferred dialogue style for extraverted users.

3. A User-Driven dialogue gives the best overall performance response when the personality of the users is not known.

4. A User-Driven dialogue combined with interface adaptation gives the best subjective assessment of performance when the personality of the users is not known.

5. The individual components of contextualization are not independent of one-another and this must be considered when designing an interface.

6. When considering the implementation of a model of contextualization for a specific application, all possible sources of contextual information should be considered as possibly affecting interaction.

The implementation of contextualization components will therefore depend on the application and the objectives of the software (at one level there may be a trade off between objective performance and subjective user impressions, for example). The diversity of the user population (and the contextualization required) will also play a part in the design of a contextualizing interface. As an example, personality and other border elements of contextualization might be built into the user model. The form of dialogue used and the way that adaptation takes place may then be tailored to each individual user in an appropriate fashion.

Thus the model of contextualization may be further developed in the light of empirical evidence and used in the design of contextualizing interfaces in agent-based (and other) systems. Future work may therefore consider exploring the effects of other contextualization components, and addressing the interaction of components in more depth.

The current model of contextualization has been designed to allow the computer (agent, software, intelligent or adaptive interface) to manipulate the cues it produces in order to aid the user in understanding the information by actively placing it in the appropriate context. The ability of the system to manipulate the dialogue style and the way dialogue is rendered allows the model to produce interfaces that provide the appropriate contextual information for the user. At the same time the interface must allow the user to provide enough data for the model to contextualize the user's response. In the model of contextualization the design of the interface (and the ability of the contextualization model to contextualize) was based on a task analysis of the domain. However, there may be other methods of implementing such interfaces, and this represents another area of future research.

5.6 Border Resources and Contextualization

The experiments in Chapter 4 focussed on personality as one of the factors which influence interaction through contextualization. Previous work (Reeves and Nass, 1996) had suggested that certain elements of personality may influence the way in which people view computers and this may in turn influence their interaction. However, personality is one of a range of user characteristics that are rarely taken into account in the design of interactive systems and remain beyond the borders of interface design.

The increasing personalisation of systems will require systems to make use of all resources and preferences to allow appropriate contextualization.

5.6.1 Making use of Existing Data and Techniques

One method of improving contextualization is to provide more data that can be made use of in rendering an interaction. The increasing use of digital devices for storing personal data together with the increasing network connectivity between systems makes the use of such data increasingly attractive. As part of a large-scale implementation of a contextualizing interface for messaging (Meech et al., under submission) digital calendar data is used as a contextual resource to infer the availability of a user to receive messages (Meech and Law, under submission). The use of these already existing contextual resources and their integration into contextualizing systems offers a promising way forward in the design of future interface systems.

Examining the components of the model of contextualization, there are also other paradigms that may be investigated in the implementation of models for dialogue instigation and rendering, etc. One interesting issue concerns the relationship between interaction and narrative in agent systems (Section 3.7). Initial investigations into this area suggest a good fit between narrative and contextualization (Meech, 1999).

5.7 Conclusion: New Perspectives On Interaction

In examining the factors that contribute to contextualization (Section 3.6), five components were identified as necessary to encompass all the elements that might effect interaction. Four of these factors concern issues that are external to the user and these are the subject of study by several research paradigms (Section 3.2). In exploring issues concerning the *user* (the fifth component of contextualization) it has become clear that there are many factors influencing what constitutes *user preferences* in an interaction. In this thesis the impact of personality has been identified and explored as a component of the user model that experiments have shown to effect both objective and subjective evaluations of interaction. These effects were identified from addressing the design of system interfaces by using active, agent-based models of the interface to control subjective system complexity. However, several other user-based contextual factors were identified as being present in emergent paradigms for agent-based interfaces (Section 3.7). Issues such as emotion, entertainment, storytelling (narrative) and other behavioural traits highlight the importance of social abilities in facilitating contextualization.

In recent research the impact of socially adept systems is becoming recognised as contributing to elements such as trust, dependability, character and other attributes usually ascribed to sentient systems. For example, returning to the original catalyst for this research – complex automated systems – Muir (1994) notes that the supervision of automated systems and the behaviour of their users is (to some extent) based on the supervisory operator's *trust* in the automation. This implies that in an agent-based system the user's trust in the agent will have some influence on the way the user interacts with the agent. How trust as a concept fits into models of interaction is still a little uncertain, but it is worth noting that several models of personality include *trustworthiness* as a personality trait.

As a paradigm for designing and evaluation interactive systems, contextualization has the capacity to encompass many of these social issues in a structured and integrated manner. Whilst much work remains to be done, contextualization offers a potentially profitable way forward in the design of personalized interfaces to complex interactive systems.

References

Addlesee, M.D., Jones, A., Livesey, F. Samaria, F. (1997) The ORL Active Floor. *IEE Personal Communications,* Vol. 4, No.5. October, 1997.

Allport, G. W. (1961) *Pattern and Growth in Personality.* Holt, Rienhart and Winston. New York.

Alty, J.L., (1984) The Application of Path Algebras to Interactive Dialogue Design. *Behaviour and Information Technology,* Vol.3, No.2. pp119-132.

Alty, J. L. and Mullin, J. (1989) Dialogue Specification in the GRADIENT Dialogue System User Interface Management Systems. *Proceedings of the HCI'89 Conference on People and Computers V*, 1989 p.151-168

Alty, J.L. and Bergan, J., (1995) The Design of the PROMISE Multimedia System and its Use in a Chemical Plant. in *Multimedia Systems and Applications*, Earnshaw, R. and Vince, J.A., (eds), Academic Press, London, pp.53-75 .

Amalberti, R and Deblon, F. (1992) Cognitive modelling of fighter aircraft process control: a step towards an intelligent on-board assistance system. *Int. J. Man-Machine Studies,* Vol.36. pp639-671.

Avouris, N.M and Van Liedekerke, M.H. (1993) User interface design for cooperating agents in industrial process supervision and control applications. *Int. J. Man-Machine Studies*, Vol.38. pp873-890.

Bailey, C.C. (1989) *The aftermath of Chernobyl : history's worst nuclear power reactor accident*. Kendall/Hunt Publishing Co. Dubuque, Iowa. USA.

Balint, L. (1995) Adaptive Human-Computer Interfaces for Man-Machine Interaction in Computer-Integrated Systems. *Computer Integrated Manufacturing Systems*, Vol.8, No.2. pp133-142. May, 1995. Butterworth-Heinemann.

Balzer, W.K, Doherty, M.E., O'Connor, R. (1989) Effects of Cognitive Feedback on Performance. *Psychological Bulletin,* Vol.106, No.3. pp410-433.

Bainbridge, L. (1987) The Ironies of Automation. In J. Rassmussen, K. Duncan and J. Leplat (Eds) *New Technology and Human Error*. London. Wiley.

Bates, J. (1994) The Role of Emotion in Believable Agents. *Communications of the ACM,* Vol.37, No.7. July, 1994.

Beale, R. and Wood, A. (1994) Agent-Based Interaction. In G. Cockton, S.W. Draper, G.R.S. Weir (Eds) *People and Computers IX*, Proceedings of HCI'94, Glasgow, U.K. Cambridge University Press.

Bellik, Y and Teil, D. (1993) A Multimodal Dialogue Controller for Multimodal user interface management system application: a multimodal window manager. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93,* April 24-29, 1993. Amsterdam, The Netherlands

Bem, S. L. (1947) The Measurement of Psychological Androgeny. *Journal of Consulting Clinical Psychology.* Vol. 42. pp155-162.

Benyon, D. (1984) Monitor – a self adaptive interface. In Shackel, B. (Ed) *INTERACT'84*. Elsevier Science. North-Holland.

Benyon, D. and Murray, D. (1988) Experience with Adaptive Interfaces. *The Computer Journal.* Vol.31, No. 5. 1988.

Beyer, H. and Holtzblatt, K. (1998) *Contextual Design.* Morgan Kaufman, San Francisco, CA. USA.

Bickmore, T.W., Cook, L.K., Churchill, E.F., Sullivan, J.W. (1998) Animated Autonomous Personal representatives*. Proceedings of the 2nd Int. Conf. On Autonomous Agents*, Minneapolis/St. Paul. MN. USA. May 9-13, 1998.

Billings, C.E. (1991) *Human-Centered Aircraft Automation Philosophy. Rationale and Concepts Guidelines.* Technical Report RTOP 505-67-21. NASA AMES Research Center, NASA. USA.

Bird, S.D. (1993) Towards a Taxonomy of multi-agent systems*. Int. J. Man-Machine Studies* Vol. 39. pp689-704.

Blandford, A.E. (1993) An agent-theoretic approach to computer participation in dialogue. *Int. J. Man Machine Studies,* Vol.39. pp965-998.

Booth, P. A. (1990) *An Introduction to Human Computer Interaction*. LEA.

Boy, G. (1991) *Intelligent Assistant Systems.* Academic Press. London.

Brazier, F.M.T., Ruttkay, Z. A (1993) Compositional, Knowledge-Based Architecture for Intelligent Query User Interfaces. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93,* April 24-29, 1993. Amsterdam, The Netherlands.

Brown, P.J. (1998) Some Lessons for Location-Aware Applications. *Proceedings of the First Workshop on Human-Computer Interaction with Mobile Devices.* GIST Technical Report G98-1. 21-23 May 1998. Glasgow, UK.

Brown, J.S. and Duguid, P. (1994) Borderline Issues: Social and Material Aspects of Design. *Human-Computer Interaction*, Vol. 9, No.1. 1994.

Brooks, A. and Thorburn, C. (1988) User-Driven Adaptive Behaviour; A comparative evaluation and an inductive analysis. In D. Jones and R. Winder (Eds) *People and Computers IV Proceedings of the Fourth Conference of the British Computer Society.* pp237-255. Cambridge University Press.

Brooks, G.L. and Siddall, E. (1980) An analysis of the Three Mile Island accident. *Atomic Energy of Canada Limited,* Mississauga, Ont. Canada.

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

Browne, D. Norman, M., Riches, D. (1990) Why build adaptive systems? In D. Browne, P. Totterdell and M. Norman (Eds) *Adaptive User Interfaces.* Academic Press. London.

Browne, D. Norman, M. and Adhami (1990) Methods for building adaptive systems. In D. Browne, P. Totterdell and M. Norman (Eds) *Adaptive User Interfaces.* Academic Press. London.

Bruner, J. (1986) *Actual Minds, Possible Worlds.* Harvard University Press, Cambridge, MA. USA.

Bruner, J. (1990) *Acts of Meaning*. Harvard University Press, Cambridge, MA. USA.

Bruner, J. (1991) The Narrative Construction of Reality. *Critical Inquiry* Vol.18, No.1. pp1-21.

Brusilovsky, P. (1996) Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, Vol.6. pp87-129.

Buck, L. (1989) Human Operators and Real-Time Expert Systems. *Expert Systems.*Vol.6, No.4. November 1989.

Byrne, D., and Nelson, D. (1965) Attraction as a linear function of proportion of positive reinforcements. *Journal of Personality and Social Psychology Bulletin,*Vol. 4. pp240-243.

Card S.K., (1989) Human Factors and Artificial Intelligence. In Hancock, P.A. and Chignell, M.H. (Eds) *Intelligent Interfaces: Theory, Research and Design.* Human Factors in Information Technology Vol.1. Elsevier Science B.V. North-Holland

Carter, J.A. (1990) The Dimensions and Degrees of Adaptation: A synergistic analysis. *Proceedings of the Human Factors Society 34th Annual Meeting.* Vol.1. pp336-340.

Carey, M.S., Stammers, R.B., Astley, J.A. (1989) Human-Computer Interaction Design: The Potential and Pitfalls of Hierarchical Task Analysis. In D, Diaper (Ed) *Task Analysis for Human-Computer Interaction*. Ellis Horwood.

Castelfranchi, C., de Rosis, F., Falcone, R. (1997) Social Attitudes and Personalities in Agents. *Proceedings of the 1997 AAAI Fall Symposium.* Report No. FS-97-02 AAAI Press. USA.

Catarci, T., Chang, S.K., Costabile, M.F., Levialdi, S., Santucci, G. A (1993) Multiparadigmatic Visual Environment for Adaptive Access to Databases.

Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93, April 24-29, 1993. Amsterdam, The Netherlands

Cattell, R. B. and Kline, P. (1976) *The scientific analysis of personality and motivation*. Academic Press, New York. USA.

Chappell, A.R. and Mitchell, C.M. (1996) Use of Visualization and Contextualization in Training Operators of Complex Syatems. *Proceedings of the Human Factors and Ergonomics Society 40th annual Meeting,* Santa Monica, CA. USA. pp264-8.

Checkland, P. (1981). *Systems Thinking, Systems Practice*. John Wiley and Sons.

Cherny, L. Clanton, C. Ostrom, E. (1997) Entertainment is a Human Factor: A CHI'97 Workshop on Game Design and HCI. *SIGCHI Bulletin* Vol.29, No.4. October 1997. ACM Press.

Chignell, M.H, Hancock, P.A., Loewenthal, A. (1989) An Introduction to Intelligent Interfaces. In P.A. Hancock and M.H. Chignell (Eds*) Intelligent Interfaces: Theory Research and Design.* Elsevier Science Publishers B.V. North-Holland.

Chignell, M.H and Hancock, P.A. (1988) Intelligent Interface Design. In M. Helander (Ed) *The Handbook of Human-Computer Interaction.* Elsevier Science Publishers B.V. North-Holland.

Chin, D.N. (1991) Intelligent Interfaces as Agents. In Sullivan, J.W. and Tyler, S.W. (Eds) *Intelligent User Interfaces.* ACM Press.

Chin, J.P., Diehl, V.A., Norman, K.L. (1988) Development of an Instrument Measuring User Satisfaction of the Human Computer Interface. *Proceedings of the ACM Conference on Human factors in Computer Systems, CHI'88*. Washington, DC, USA. pp213-218

Clarke, A. A. and Smyth, M. G. G. (1993) A Cooperative Computer Based on the Principles of Human Cooperation. *Int. J. Man-Machine Studies*, Vol.38, No. 1. pp3-22.

Clibbon, K., (1995) Contextually Adapted Hypertext for Learning. In *Companion Proceedings of ACM Conference on Human factors in Computing Systems CHI'95*. May 7-11, 1995. Denver, Colorado, USA.

Cohen, B., Harwood, W. T., and Jackson, I. M. (1986). *The Specification of Complex Systems.* Addison -Wesley.

Cohen, R., Allaby, C., Cumbaa, C. (1998) What is Initiative? *Journal of User Modelling and User-Adapted Interaction*, Vol. 8, Nos. 3-4. pp171-214.

Cooper, G. (1991) Context and its Representation. *Interacting with Computers*, Vol.3, No.3. pp243-252.

Copeland, C.J. and Eccles, S.R. (1992) Skill metrics on a genetic graph as a mechanism for driving adaptation in an operating system interface. *Int. J. Man-Machine Studies* Vol.36. pp697-718.

Crews, T.R. (1995) AdventurePlayer: An Intelligent Learning Environment. In *Companion Proceedings ACM Conference on Human Factors in Computing Systems. CHI'95*. May 7-11. Denver Colorado, USA.

Croft, W.B. (1984) The role of context and adaptation in user interfaces. *Int. J. Man-Machine Studies*. Vol. 21. pp283-292.

Curtis, P. (1992) Mudding: Social Phenomena in text-based virtual realities. Proceedings of *Directions and Applications of Advanced Computing*, Berkely, CA.USA. May 1992.

Cypher, A. (1991) Eager: Programming repetitive tasks by example. Proceedings of the *ACM Conference on Human Factors in Computing Systems' CHI'91*. New Orleans, LA. USA. pp33-39.

Danielsen, R.L., Farrand, A.B, Wolfe, S.J. (1993) Searching for Help vs Having it Handed to You. The Relative Advantages of Index-accessed Help and Context-Sensitive Help. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands.

Debevc, M. (1993) Adaptive Bar. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands. pp117-118.

Debevc, M., Meyer, B., Donlagic, D., Svecko, R. (1996) Design and Evaluation of an Adaptive Icon Toolbar. *User Modelling and user-Adapted Interaction* Vol.6, pp1-21.

De Greene, K. B. (1991) Emergent Complexity and Person-Machine Systems. *Int. J. Man-Machine Studies* Vol.35. pp219-234.

Department of Transport (1990). Report on the accident to Boeing 737-400 G-OBME near Kegworth, Leicestershire on 8 January 1989. *Department of*

Transport, Air Accidents Invvestigation Branch, Aircraft Accident Report 4/90. London, HMSO. ISBN0 11 550986 0

Desmarais, M.C., Giroux, L., Larochelle, S. (1993) An advice-giving interface based on plan-recognition and user-knowledge assessment. *Int. J. Man-Machine Studies* Vol.39. pp901-924.

Desmarais, M. C., Liu, J. (1993) Exploring the Applications of User-Expertise Assessment for Intelligent Interfaces. *Proceedings of the ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands. pp308-313.

Dieterich, H., Malinowski, U., Kuhme, T., Schneider-Hufschmidt,M. (1993) State of the Art in Adaptive User Interfaces. In Schneider-Hufschmidt, M., Kuhme, T., Malinowski, U. (Eds) (1993) *Adaptive User Interfaces: principles and practice.* Elsevier Science Publishers B.V. North-Holland.

Doyle, P. and Hayes-Roth, B. (1997) An Intelligent Guide for Virtual Environments. *IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent.* Nagoya, Japan. August 25, 1997.

Dix, A. J., Finlay, J. E., Abowd, G. D., Beale, R. (1992) *Human Computer Interaction*. Prentice-Hall. London.

Dryer, D.C. (1997) Wizards, Guides, and Beyond: Rational and Empirical Methods for Selecting Optimal Intelligent User Interface Agents. *Proceedings of the 1997 International Conference on Intelligent User Interfaces. IUI'97.* ACM. New York. USA.

Eberts, RE, and Eberts, C.G. (1989) Four Approaches to Human Computer Interaction. In Hancock, P.A. and Chignell, M.H. (Eds) *Intelligent Interfaces: Theory, Research and Design.* Human Factors in Information Technology. Vol.1 Elsevier Science B.V. North-Holland.

Edmonds, E.A. (1981) Adaptive Man-Computer Interfaces. In M.J. Coombs (Ed) *Computing Skills and the User Interface*. London. Academic Press.

Edmonds, E.A. (1987) Adaptation, Response and Knowledge (Editorial*)* *Knowledge-Based Systems* Vol.1, No.1.

Edmonds, E.A. Murray, B.S., Ghazikhanian, J., Heggie, S.P. (1992) The Re-Use and Integration of Existing Software: A central role for the intelligent user interface. *Proceedings of the Seventh Conference of the British Computer Society Human Computer Interaction Specialist Group :People and Computers VII*. University of York, U.K. pp415-427.

Edmondson, W.H. (1993) A Taxonomy for Human Behaviour and Human-Computer Interaction. *Proceedings of the Fifth International Conference on Human-Computer Interaction*, Orlando, Florida, August 8-13, 1993.

Edmondson, W. H. and Meech J. F. (1994) Putting Task Analysis into Context. *SIGCHI Bulletin* Vol. 26, No. 4. October 1994. ACM Press.

Edwards, J. L. and Mason, J. A. (1988) Evaluating the intelligence in dialogue systems. *Int. J. Man-Machine Studies.* Vol.28. pp139-173.

Egan, D.E. (1988) Individual Differences in Human-Computer Interaction. In M. Helander (Ed) *Handbook of Human-Computer Interaction*. pp343-568. Elsevier Science Publishers, North-Holland.

Elkerton, J. and Williges, R.C. (1989) Dialogue Design for Intelligent Interfaces. In Hancock, P.A. and Chignell, M.H. (Eds) *Intelligent Interfaces: Theory, Research and Design.* Human Factors in Information Technology Vol.1 Elsevier Science B.V. North-Holland.

Elliot, C., Brzezinski, J., Sheth, S., Salvatoriello, R. (1998) Story-morphing in the Affective Reasoning paradigm: Generating stories semi-automatically for use with "emotionally intelligent" multimedia agents. *Proceedings of Autonomous Agents '98.* May 9-13, 1998. Minneapolis MN. USA.

Etzioni, O.E. and Weld, D.S. (1995) Intelligent Agents on the Internet: Fact, Fiction and Forecast. *IEEE Expert*, August 1995. pp44-49.

Ewen, R. B. (1998) *An introduction to theories of personality*. Lawerence Erlbaum Associates, London.

Fels, S. and Hinton, G. (1995) Glove-TalkII: An Adaptive Gesture-to-Format Interface. *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95*. May 7-11, 1995. Denver, Colorado, USA.

Fischer, G (1991) The Importance of Models in Making Complex Systems Comprehensible. In Tauber, M. J. and Ackerman D. (Eds) *Mental Models and Human-Computer Interaction 2.* North-Holland.

Fischer, G., Lemke, A. C., Mastaglio, T., Morch, I.A. (1991) Critics: An Emerging Approach to Knowledge-Based Human-Computer Interaction. *Int. J. Man-Machine Studies*, Vol.35, No.5. pp695-721, 1991.

Flood, R. L., & Carson, E. R. (1988). *Dealing with Complexity*. New York: Plenum Press.

Folleso, K., Kaarstad, M., Droivoldsmo, A. and Kirwan, B. (1995) Relations between task complexity, diagnostic strategies and performance in diagnosing process disturbances. *Proceedings of the 5th European Conference on cognitive science approaches to process control*, August 30 - September 1 1995, Espoo, Finland. VTT Technical Publication 158.

Foner, L.N. (1997) Entertaining Agents: A Sociological Case Study. *Proceedings of Autonomous Agents 1997*. Marina del Ray, CA. ACM Press.

Foster, D. and Meech, J.F. (1995) Social Dimensions of Virtual Reality. In K. Carr and R. England (Eds) *Simulated and Virtual Realities, Elements of Perception.* Taylor and Francis, London 1995.

Fowler, C.J.H., Macauley, L.A., Siripoksup, S. (1987) An Evaluation of the Effectiveness of the Adaptive Interface Module (AIM) in Matching Dialogues to Users. *Proceedings of the Third Conference of the British Computer Society Human Computer Interaction Specialist Group HCI'87: People and Computers III*. University of Exeter. September 7-11, 1987. pp345-359.

Franklin, S and Graesser, A. (1996) Is it an agent, or just a program?: A taxonomy for autonomous agents. *Proceedings of the Third International Workshop on Agent Theories, Architectures and Languages*. Springer-Verlag.

Friedman, B. (1995) "It's the Computers Fault" – Reasoning about computers as moral agents. In *Companion proceedings of ACM Conference on Human factors in Computing Systems, CHI'95*. May 7-11, Denver, Colorado, USA

Friedman, B. and Nissenbaum, H. (1997) Software Agents and User Autonomy. *Proceedings of Autonomous Agents* 1997. Marina del Ray, CA. ACM.

Funk, P. and Lind, J. (1997) What is a friendly agent ? Socially Intelligent Agents *Proceedings of the 1997 AAAI Fall Symposium.* Report No. FS-97-02 AAAI Press.

Funke, D.J., Neal, J.G., Paul, R.D. (1993) An approach to intelligent automated window management. *Int. J. Man-Machine Studies* Vol.38. pp949-983.

Furnas, G.W. (1985) Experience with an adaptive indexing scheme. *Proceedings of ACM Conference on Human Factors in Computing Systems CHI'85.* San Francisco, California 14-18 April,1985. ACM Press.

Galyean, T.A. (1995) *Narrative Guidance of Interactivity.* (PhD Thesis), MIT Media Lab. Boston. MA. USA.

Grand, S., Cliff, D., Malhotra, A. (1997) Creatures: Artificial Life Autonomous Software Agents for Home Entertainment. *Proceedings of Autonomous Agents 1997.* Marina del Ray, CA. ACM.

Green, T.R.G., Davies, S.P., Gilmore, D.J. (1996) Delivering cognitive psychology to HCI: the problems of common language and knowledge transfer. *Interacting with Computers.* Vol. 8, No. 1. March 1996. pp89-111.

Greenberg, S. and Witten, I.H. (1985) Adaptive Personalised Interfaces – a question of viability. *Behaviour and Information Technology* Vol.4, No.1. pp31-45.

Hagen, L., Breugst, M., Magedanz, T. (1998) Impacts of Mobile Agent Technology on Mobile Communication System Evolution. *IEEE Personal Communications* Vol.5, No.4.

Hancock, P.A. and Chignell, M.H. (Eds) (1989) *Intelligent Interfaces: Theory, Research and Design. Human Factors in Information Technology Vol.1*. Elsevier Science B.V. North-Holland.

Hayek, F. (1967) The Theory of Complex Phenomena. *in Studies in Philosophy, Politics and Economics*. University of Chicago Press.

Hayes-Roth, B. and van Gent, R. (1997) Story-Making with Improvisational Puppets. *IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent*. Nagoya, Japan. August 25, 1997.

Helander, M. (Ed) (1988) *Handbook of Human Computer Interaction*. Amsterdam, North-Holland.

Hellman, R. (1990) User Support: Illustrating Computer Use in Collaborative Work Contexts. *Proceedings of ACM CSCW'90 Conference on Computer-Supported Cooperative Work.* Los Angeles, California. 7-10 October, 1990. ACM Press. pp 255-67

Higgins, T.J., Chignell, M.H., Hancock, P.A. (1989) Knowledge-Based Supervisory Control for Aerospace Applications. In Hancock, P.A. and Chignell, M.H. (Eds) (1989) *Intelligent Interfaces: Theory, Research and Design. Human Factors in Information Technology Vol.1* Elsevier Science B.V. North-Holland.

Hiyoshi, M., Shimazu, H. Takashima, Y A (1993) Construction Tool for Context-Sensative Guidance systems. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands.

Hockley, A.T. (1986) Adaptive user interfaces for information systems: an evaluation. In P. Zunde and J.C. Agrawal (Eds) *Empirical Foundations of Information and Software Science IV: Empirical Methods of Evaluation of Man-Machine Interfaces*. New York. Plenum Press.

Hollnagel, E. (1987) Information and reasoning in intelligent decision support systems. *Int. J. Man-Machine Studies* Vol.27. pp665-678.

Hollnagel, E. (1993) Models of Cognition: Procedural Prototypes and Contextual Control. *Le Travail Humain*, Vol. 56, No. 1. 1993 pp27-51.

Hollnagel, E. (1995) Interface evaluation and cognitive ergonomics. *Proceedings of the 5th European Conference on cognitive science approaches to process control,* August 30 - September 1, 1995, Espoo, Finland. VTT Technical Publication 158.

Holm, P. and Karlgren, K. (1995) Theories of meaning and different perspectives on information systems. *Proceedings of the IFIP International Conference on Information Systems Concepts*. Chapman and Hall, London, UK.

Hoyle, M.A. and Lueg, C. (1997) Open Sesame!: A look at personal assistants. *Proceedings of the second international conference on the practical application of intelligent agents and multi-agent technology. PAAM'97*. 21-23 April 1997. London. ISBN 0952555468

Hutchins, E (1989) Metaphors for Interface Design. In M. M. Taylor, F. Neel and D. G. Bouwhuis (Eds) *The Structure of Multimodal Dialogue*. Amsterdam, North-Holland.

Isbister, K. and Nass, C. (in press) Consistency of Personality in Interactive Characters: Verbal Cues, Non-Verbal Cues, and User Characteristics. *International Journal of Human-Computer Interaction*

Innocent, P.R. (1982) Towards self-adaptive systems. *Int. J. Man-Machine Studies* Vol.16. pp287-299.

Jennings, N.R., Sycara, K., Wooldridge, M. (1998) A Roadmap of Agent research and Development. *Autonomous Agents and Multi-Agent Systems,* Vol.1, pp275-306. Kluwer Academic Publishers, Boston. USA.

Kay, A. (1990) The User Interface: A Personal View. In B. Laural (Ed*) The Art of Human-Computer Interface Design*. Addison-Wesley, Reading, Mass. pp191-207.

Kautz, H.A., Selman, B., Coen, M., Ketchpel, S. (1994) An Experiment in the Design of Software Agents. *Proceedings of the 1994 AAAI Spring Symposium on Software Agents*. March 21-23, Stanford University, USA.

Kedar, S., Baudin, C., Birnbaum, L. (1993) Ask How It Works: An Intelligent Interactive Manual for Devices. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93,* April 24-29, 1993. Amsterdam, The Netherlands

Keirsey, D. and Bates, M. (1984*)* *Please Understand Me: Character and Temperament Types*. Prometheus Nemesis Book Co Inc. USA.

Kelso, M.T and Weyhrauch, P. (1992) *Dramatic Presence*. Carnegie Mellon University Report CMU-CS-92-195. Pittsburgh, PA. USA.

Kessler, M. and Kilgore, R. (1997) Bringing an Agent to Life. *IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent*. Nagoya, Japan. August 25, 1997

Kiesler, D. J. (1983) The 1982 interpersonal circle: a taxonomy for complementarity in human transactions. *Psychological Review*, Vol. 90. pp185-214.

Kieras, D. E. and Polson, P. G. (1985) An approach to the formal analysis of user complexity. *International Journal of Man-Machine Studies*, Vol. 22, pp365-394

Kirlik, A. and Markert, W.J. (1990) Perceptual and Contextual Influences on Dynamic Decision-Making Performance. *Systems, Man and Cybernetics*. IEEE Press.

Knowles, C. (1988) Can Cognitive Complexity Theory (CTT) Produce an Adequate Measure of System Usability? *Proceedings of the Fourth Conference of the British Computer Society Human Computer Interaction Specialist Group: People and Computers IV*. University of Manchester. pp291-307

Koda, T. and P. Maes, (1996) Agents with Faces: The Effects of Personification of Agents, *Proceedings of HCI'96,* London, UK, August 20-23, 1996, p98-103, The British HCI Group.

Kossakowski, M (1989) Adaptive Man-Computer Dialogues. In F. Klix, N.A. Streitz, Y. Waern and H. Wandke (Eds) *Man-Computer Interaction Research MACINTER II*. Amsterdam. North-Holland.

Kullberg, R.L. (1995) Mark Your Calendar!: Learning personalized annotation from integrated sketch and speech. *Companion Proceedings ACM Conference on Human Factors in Computing Systems CHI'95.* May 7-11, Denver Colorado, USA.

Kushmerick, N. (1996) Cognitivism and Situated Action: Two views on intelligent agency. *Computers and Artificial Intelligence*. Vol.15, No.5. pp393-417.

Kushiro, N., Sato, K., Komiya, N., Nakata, M., Inoue, M., Tsubota, H. (1996) Agent-based graphical user interface architecture for consumer products. *IEEE Trans. On Consumer Electronics* Vol.42, No.4. November 1996.

Lachman, R. W. (1997) *Animist Interface: Experiments in Mapping Character Anaimation to Computer Interface.* (Master's Thesis), Massachusetts Institute of Technology, 1997.

Lashkari, Y., Metral, M., Maes, P. (1994) Collaborative Interface Agents. *Proceedings of the 12th National Conference on Artificial Intelligence (AAAI)* Vol.1. 31 July – 4 Aug, 1994. Seattle, Washington USA.

Laurel, B., Oren, T., Don, A. (1990) Issues in Multimedia Interface Design: Media integration and interface agents *Proceedings of ACM Conference on Human Factors in Computing Systems: Empowering People (CHI'90*). Seattle, Washington Mach 1-5, 1990. pp 133-139

Lee, A. (1993) User Support: Considerations, Features and Issues In R.H. Hartson and D. Hix (Eds) *Advances In Human-Computer Interaction* Vol.3. Ablex Publishing.

Lester, J.C., Converse, S.A., Kahler, S.E., Barlow, S.T., Stone, B.A., Bhogal, R.S. (1997) The Personal Effect: Affective Impact of Animated Pedagogical Agents. *Proceedings of ACM Conference on Human Factors in Computing Systems CHI'97*. 22-27 March 1997, Atlanta, GA. USA. pp359-366.

Leveson, N.G. (1995) *Safeware: System Safety and Computers*. Addison Wesley, NY. USA.

Lewis, J.R. (1995) IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *International Journal of Human-Computer Interaction.* Vol.7, No.1, pp57-78.

Lieberman, H. (1997) Autonomous Interface Agents. *Proceedings of ACM Conference on Human Factors in Computing Systems, CHI'97*. 22-27 March 1997, Atlanta, GA. USA. pp924-929.

Lin, H.X., Choong, Y.-Y., Salvendy, G. (1997) A Proposed Index of Usability: A Method for Comparing the Relative Usability of Different Software Systems. *Behaviour and Information Technology*. Vol. 16, No.4-5. pp267-278.

Liscano, R., Impey, R., Gordon, P., Abu-Hakima, S., (1996) A System for the Seamless Integration of Personal Messages using Agents Developed on a Lotus Notes Platform. *CASCON 96 Meeting of Minds* November 12-14, 1996. IBM Canada.

Lokuge, I., and Ishizaki, S. (1995) GeoSpace: An Interactive Visualization System for Exploring Complex Information Spaces. *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95*. May 7-11, 1995. Denver, Colorado, USA.

Lucas, M, Meech, J.F., Purcell, C. (1997) Re-Engineering a Complex Network Interface. *Proceedings of HCI International '97*, San Francisco, CA. USA. Elsevier Science.

Macer, P.J., (1998) Ubiquitous Remote Control of Network Appliances using the JetSend Communications Protocol. *IEEE Workshop on Networked Appliances IWNA98*, 4-6 November 1998. Kyoto, Japan.

Macer, P.J., Thomas, P., Chalabi, N., Meech, J. F. (1996) Finding the Cut of the Wrong Trousers: Fast Video Search Using Automatic Storyboard Generation. *ACM Conference on Human Factors in Computing Systems Conference CHI'96*, April 14-18, Vancouver, Canada.

Maes, P. (1994a) Agents that Reduce Work and Information Overload. *Communications of the ACM*, Vol.37, No.7. July 1994. pp31-40.

Maes, P. (1994b) Social Interface Agents: Acquiring Competence by Learning from Users and other Agents. *AAAI Spring Symposium on Software Agents*, March 21-23, Stanford University, USA. Report 55-94-03. AAAI Press, Malno Park CA. USA.

Malinowski, U., Kuhme, T., Dieterich, H., Schneider-Hufschmidt, M., (1992) A Taxonomy of Adaptive User Interfaces. *People and Computers VII; Proceedings of HCI'92,* York. UK. pp391-414.

Malinowski, U., Kuhme, T., Dieterich, H., Schneider-Hufschmidt, M. (1993) Computer-Aided Adaptaion of User Interfaces with Menus and Dialog Boxes *Proceedings of the Eighth Conference of the British Computer Society Human Computer Interaction Specialist Group: People and Computers VIII.* Loughborough University. 7-10 September. Vol.2. pp122-127.

Mallen, C. (1996) Designing Intelligent Help for Information Processing Systems. *Int. J. Human-Computer Studies.* Vol.45, No.3. pp349-377.

Malone, T.W., Grant, K.R., Lai, K-Y. (1997) Agents for Information Sharing and Coordination: A History and Some Reflections. In J.M. Bradshaw (Ed) *Software Agents* AAAI Press/MIT Press.

Mase, K. (1997) Aspects of Interface Agents: Avatar, Assistant, Actor. *IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent.* Nagoya, Japan. August 25, 1997.

Maskery, H.S. (1984) Adaptive Interfaces for Naïve Users – an experimental study. In Shackel, B. (Ed) *INTERACT'84*. Elsevier Science. North-Holland.

Maskery, H., and Meads, J (1992) Context: In the Eyes of Users in Computer Systems. *SIGCHI Bulletin* Vol.24, No.2. April 1992. pp12-22.

Maskery, H., Hopkins, G. and Dudley, T. (1992) Context: What does it mean to Application Design. *SIGCHI Bulletin* Vol.24, No.2. April 1992. pp22-31.

Mason M.V. and Thomas, R.C. (1984) Experimental Adaptive Interface. *Information Technology* Vol.3, No.3. pp162-167.

Mateas, M. (1997) *An Oz-Centric Review of Interactive Drama in Believable Agents.* Carnegie Mellon University Report CMU-CS-97-156. Pittsburgh, PA. USA.

Maulsby, D., Greenberg, S. Mander, R. (1993) Prototyping an Intelligent Agent through Wizard of Oz. *Proceedings of the ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands. pp165-171.

Meech, J. F. (1992a) Addressing Operator Errors in Supervisory Systems. *Proceedings of Information-Decision-Action Systems in Complex Organizations IDASCO 92*, IEE Digest 353, April 1992, Oxford, U.K.

Meech, J. F. (1992b) Intelligent Aiding in Complex, Supervisory Systems. *Proceedings of the 5th IFIP/IFORS/IFAC Conference on Man-Machine Systems,* The Hague, The Netherlands. June 1992.

Meech, J. F. (1994a) The Intelligent Management of Information for Human Interaction with Complex, Real-Time Systems. *AAAI Spring Symposium on Intelligent Multi-Media, Multi-Modal Systems*, Stanford University, USA. March 21-23, 1994.

Meech, J. F. (1994b) Operator User Interface for Navigation for LHC Experiments. *CERN Computing Seminar*, CERN Prevessin, Geneva. 8th November 1994.

Meech, J. F. (1995) Interfaces that help the user contextualize information. In *Symbiosis of Human and Artifact*, Y. Anzai, K. Ogawa and H. Mori (Eds), Elsevier Science, 1995.

Meech, J. F. (1999) Narrative Theories as Contextual Constraints for Agent Interaction. Paper to be presented at the *AAAI Fall Symposium on Narrative Intelligence*. Falmouth, MA. USA. November 5-7, 1999.

Meech, J. F. and Abu-Hakima, S. (1998) Multi-Modal Seamless Messaging. *Proceedings of AAAI'98 Workshop on Representations for Multi-Modal Human-Computer Interaction*. AAAI Press. CA. USA.

Meech, J.F. Baker, K., Law, E. Liscano, R. (under submission) An Intelligent Interface for Personal Messaging. Paper submitted to *the ACM Conference on Intelligent User Interfaces, IUI2000.* New Orleans, USA. January 9-12, 2000.

Meech, J. F., Huuskonen, P., Le Goff, J-M and Wagner, E. (1995) An Analysis of the Human-Computer Interfaces to High-Energy Physics Control Systems at CERN. *Symbiosis of Human and Artifact* Y. Anzai, K. Ogawa and H. Mori (Eds), Elsevier Science.

Meech, J. F. and Law, E. (under submission) Learning User Availability for Messaging from Digital Calendar Data. Paper submitted to *the ACM Conference on Human Factors in Computing Systems, CHI2000*. The Hague, The Netherlands. April 1-7, 2000.

Milewski, A.E. and Lewis, S.H. (1997) Delegating to Software Agents. *Int. J. Human-Computer Studies* Vol.46, pp485-500.

Miller, J.R., Sullivan, J.W., Tyler, S.W. (1991) Introduction to Sullivan, J.W. and Tyler, S.W. (Eds) *Intelligent User Interfaces*. ACM Press

Mittal, V.O. and Paris, C.L. (1993) Context: identifying its elements from the Communication point of view. *Proceedings of the IJCAI-93 workshop on Using Knowledge in its Context.* August 29, 1993. Chambery, France.

Mittal, V.O. and Moore, J.D. (1995) Dynamic Generation of Follow up Question Menus: Facilitating Interactive Natural Language Dialogues. *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95*. May 7-11, 1995. Denver, Colorado, USA.

Moon, Y. and Nass, C. (1996a) Adaptive agents and personality change: complementarity versus similarity as forms of adaptation. *Proceedings of Human Factors in Computer Systems CHI'96.* April 13-18, 1996. Vancouver, Canada. ACM Press.

Moon, Y. and Nass, C. (1996b) How "real" are computer personalities? Psychological responses to personality types in human-computer communication. *Communication Research.* 1996.

Muir, B. M. (1994) Trust in Automation: Part 1. Theoretical issues in the study of trust and human intervention in automated systems. *Ergonomics*. Vol. 37, No.11. pp1905-1922.

Murray, D. (1991) Modelling for Adaptivity. In M.J. Tauber and D. Ackerman (Eds) *Mental Models and Human-Computer Interaction 2*. Elsevier Science. North-Holland

Nardi, B.A. (1992) Studying Context: A Comparison of Activity Theory, Situated Action Models and Distributed Cognition. *Proceedings of the East-West International Conference on Human-Computer Interaction, EWHCI'92*. Moscow, Russia.

Nardi, B.A. (1996) Activity Theory and Human-Computer Interaction. In Nardi, B.A. (Ed) *Context and Consciousness: Activity Theory and Human Computer Interaction.* MIT Press, Cambridge, Mass. USA.

Nardi, B.A., Miller, J.R, Wright, D.J. (1998) Collaborative, Programmable Intelligent Agents. *Communications of the ACM*, Vol.41, No.3. March 1998.

Nass, C., Moon, Y., Fogg, B. J., Reeves, B. and Dryer, D.C. (1995) Can computer personalities be human Personalities? *Int. J. Human-Computer Studies* Vol.43, No.2 pp223-239.

Neal, J.G. and Shapiro, S.C. (1991) Intelligent Multi-Media Interface Technology. In Sullivan, J.W. and Tyler, S.W. (Eds) *Intelligent User Interfaces*. ACM Press

Neerincx, M. and de Greef, P. (1993) How to Aid Non-Experts. *Proceedings of the ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands. pp165-171.

Negroponte, N. (1990) Hospital Corners. In B. Laurel (Ed) *The Art of Human-Computer Interface Design*. Addison-Wesley, Reading, Mass. USA. pp347-353

Nielsen, J. (1997) Telephone Usability: Voice is Just Another Datatype. www.useit.com/papers/telephone_usability.htm

Noma, T. and Badler, N. I. (1997) A Virtual Human Presenter. *IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent*. Nagoya, Japan. August 25, 1997

Norico, A.F. and Stanley, J. (1989) Adaptive Human-Computer Interfaces: A Literature Survey and Perspective. *IEEE Trans. on Systems, Man and Cybernetics* Vol.19, No.2. March-April 1989. pp399-408.

Norman, D.A. (1986) Cognitive Engineering. In D.A. Norman and Draper, S. (Eds) *User Centred System Design: New perspectives on Human-Computer Interaction*. LEA.

Norman, D.A. (1990) The Problem with Automation: Inappropriate Feedback and Interaction, Not "over automation". In D.E. Broadbent, J. Reason & A. Baddeley (Eds) *Human factors in Hazardous Situations*. Clarendon Press. Oxford. U.K.

Norman, D.A. (1994) How People Might Interact With Agents. *Communications of the ACM*, Vol.37, No.7. July 1994. pp68-71.

Nwana, H.S. (1996) Software Agents: an overview. *The Knowledge Engineering Review*. Vol.11, No.3. 1996, pp205-244.

Opperman, R. (Ed) (1994) *Adaptive User Support*. LEA Press, Hillsdale N.Y. USA

Opperman, R. (1994) Adaptively Supported Adaptability. *Int. J. Human-Computer Studies* Vol.40, No.3. pp455-472.

Oren, T., Salomon, G., Krietman, K. and Don, A. (1990) Guides: Characterizing the Interface. In B. Laurel (Ed) *The Art of Human-Computer Interface Design*. Addison Wesley. Reading, MA.

Perlin, K., and Goldberg, A. (1998) *Improvisational Animation. White-paper available at the New York University Media Research Lab*, New York, NY. USA. http://www.mrl.nyu.edu/improv/whitepaper/index.html

Perrow, C. (1984). *Normal Accidents*. Basic Books.

Picard, R. (1997) *Affective Computing*. MIT Press.

Pickering, J.A., Arnott, J.L., Wolff, J.G., Swiffin, A.L. (1984) Prediction and adaption in a communication aid for the disabled. In Shackel, B. (Ed) *INTERACT'84*. Elsevier Science. North-Holland.

Polson, M.C., Wickens, C.D., Klapp, S.T., Colle, H.A. (1989) Human Interactive Informational Processes. In Hancock, P.A. and Chignell, M.H. (Eds) *Intelligent Interfaces: Theory, Research and Design. Human Factors in Information Technology* Vol.1 Elsevier Science B.V. North-Holland.

Preece, J., Rogers, Y., Sharp, H., Benyon, D. (1994) *Human-Computer Interaction*. Addison Wesley, London.

Rasmussen, J. (1990) Mental Models and the Control of Action in Complex Environments. In D. Ackerman and M.J. Tauber (Eds*) Mental Models and Human-Computer Interaction 1*. Elsevier Science Publishers B.V. North-Holland.

Rasmussen, J and Vicente, K.J. (1990) Ecological Interfaces: A Technological Imperative in High-Tech Systems*? International Journal of Human-Computer Interaction* Vol.2, No.2. pp93-110.

Reed, F.W. (1990) Intelligent Tutoring Systems as Models for Adaptive Operational Decision Aids. *Proceedings of IEEE Conference on Systems, Man and Cybernetics* 1990, pp808-811.

Reeves, B. and Nass, C. (1996) *The Media Equation: How people treat computers, television and new medial like real people and places.* CSLI Publications. CA. USA.

Reiter, J. E. (1984) A Review of the PROSPECTOR Project. *IEEE Data Engineering Bulletin* Vol.6, No.4. December 1984.

Rich, C. and Sidner, C.L. (1997) COLLAGEN: When Agents Collaborate with People. *Proceedings of the 1st Int. Conf. On Autonomous Agents*, Marina del Ray, CA, USA. Feb 1997.

Rich, C. and Sidner, C.L. (1998) COLLAGEN: A collaboration Manager for Software Interface Agents. *User Modelling and User-Adapted Interaction*. Vol.8, No.3-4. 1998.

Richardson, T., Bennett, F., Mapp, G., Hopper, A. (1994) Teleporting in an X Window System Environment. *IEEE Personal Communications* Vol.1, No.3. 1994.

Rimmon-Kenan, S. (1983) *Narrative Fiction: Contemporary Poetics*. Methuen.

Rissland, E.L. (1984) Ingredients of Intelligent User Interfaces. *Int. J. Man-Machine Studies* Vol.21. pp377-388.

de Rosis, F. De Carolis, N., Pizzutilo, S. (1993) User Tailored Hypermedia Explanations. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands.

Roth, S.F., Kolojejcick, J., Mattis, J., Goldstein, J. (1994) Interactive Graphic Design Using Automatic Presentation Knowledge. *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'94*, April 24-28, 1994. Boston Massachusetts, USA.

Rouse, W.B. (1991) *Design for Success: A human centered approach for designing successful products and systems*. John Wiley and Sons. London.

Salber, D., Dey, A.K., Abowd, G.D. (1999) The Context Toolkit: Aiding in the Development of Context-Enabled Applications. *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'99.* 15-20 May 1999. Pittsburgh, PA, USA.

Savidis, A. and Stephanidis, C. (1995) Developing Dual User Interfaces for Integrating Blind and Sighted Users: The HOMER UIMS. *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95.* May 7-11, 1995. Denver, Colorado, USA.

Schneider-Hufschmidt, M., Kuhme, T., Malinowski, U. (Eds) (1993) *Adaptive User Interfaces: principles and practice.* Elsevier Science Publishers B.V. North-Holland.

Schweighardt, M.F. (1990) Using the Context of Interactions to Adapt to Users. *Proceedings of the Human Factors Society 34th Annual Meeting* 1990.

Schwind, C.B. (1990) An intelligent language tutoring system. *Int. J. Man-Machine Systems* Vol.33. pp557-579.

Sen, S., Haynes, T., Arora, N. (1997) Satisfying user preferences while negotiating meetings. *Int. J. Human-Computer Studies* Vol.47. pp407-427.

Sengers, P. (1998) *Anti-Boxology: Agent Design in Cultural Context*. (PhD Thesis). Carnegie Mellon University Report CMU-CS-98-151. Pittsburgh, PA. USA.

Shardanand, U. and Maes, P. (1995) Social Information Filtering: Algorithms for Automating "word of mouth". *Proceedings of the ACM Conference on Human Factors in Computing Systems, CHI'95*. May 7-11, 1995. Denver, Colorado, USA. pp 210-217

Sheridan, T.B. (1984) Telerobotics. *Automatica* Vol.25, No.4. pp487-507.

Sheridan, T. B., and Hennessy, R.T. (eds) (1994) *Research and Modeling of Supervisory Control Behavior*. National Academy Press. Washington DC, USA.

Shneiderman, B (1987) *Designing the User Interface*. Addison Wesley, Reading Mass. USA.

Shneiderman, B. and Maes, P. (1997) Direct manipulation vs. interface agents: Excerpts from debates at IUI 97 and CHI 97. *ACM Interactions*. Vol. 4, No. 6. November 1997.

Shortliffe, E.H. and Buchanen, B. (1984) *Rule Based Expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison Wesley, Reading UK.

Silverman, B.,G. (1992) Critiquing human error: a knowledge based human-computer collaboration approach. *Knowledge-based systems.* Vol. 9. Academic Press.

Simon, H.A. (1996) *The Sciences of the Artificial*. MIT Press. Boston. Mass. USA

Simonsen, J., Kensing, F. (1997) Using ethnography in contextual design. *Communications of the ACM*, Vol.40, No.7. pp82-88.

Small, D and Ishii, H. (1997) Design of Spatially Aware Graspable Displays. *Extended Abstracts of the ACM Conference on Human Factors in Computer Systems CHI'97*, March 22-27, 1997. Atlanta, Georgia, USA.

Smith, S. and Bates, J. (1989) *Towards a Theory of Narrative for Interactive Fiction*. Carnegie Mellon University Technical Report CMU-CS-89-121. Pittsburgh, PA.

Soltysiak, S. and Crabtree, B. (1998) Knowing Me, Knowing You: Practical Issues in the Personalisation of Agent Technology. *Proceedings of 3rd Int. Conf. Practical Applications of Intelligent Agents and Multi-Agent Technology (PAAM'98),* London, March 23-25, 1998.

Stajano, F. and Jones, A. (1998) The Thinnest of Clients: Controlling It All Via Cellphone. *ACM Mobile Computing and Communications Review*, Vol.2. No.4. October, 1998.

Stevens, R., D. and Edwards, A.D.N., (1996) An Approach to the Evaluation of Assistive Technology. *Proceedings of the Second Annual ACM Conference on Assistive Technologies, Empirical Studies*, pp64-71, 1996. ACM Press.

Storrs, G. (1989) A Conceptual Model of Human-Computer Interaction? Towards a Theory of Human Computer Interaction. *Behaviour and Information Technology* Vol.8, No.5. pp323-334, 1989.

Sukaviriya, P. (1993) From User Interface design to the Support of Intelligent and Adaptive Interfaces: an overhaul of user interface software infrastructure. *Knowledge Based Systems* Vol.6, No.4. pp220-230.

Sullivan, J.W. and Tyler, S.W. (Eds) (1991) *Intelligent User Interfaces*. ACM Press.

Sutcliffe, A. (1988) *Human Computer Interface Design.* Springer-Verlag.

Thelwell, P. (1994) What defines complexity? In S. A. Robertson (Ed). *Contemporary Ergonomics: Ergonomics for All*. Taylor and Francis, London.

Thimbleby, H. and Addison, M. (1996) Intelligent Adaptive Assistance and its Automatic Generation. *Interacting with Computers* Vol.8, No.1 pp51-68. Elsevier Science.

Thomas, C.G. (1993) Design, implementation and evaluation of an adaptive user interface. *Knowledge Based Systems* Vol.6, No.4. pp230-239.

Thorisson, K.R. (1993) Dialogue Control in Social Interface Agents. *Adjunct Proceedings ACM Conference on Human Factors in Computing Systems, INTERCHI'93*, April 24-29, 1993. Amsterdam, The Netherlands

Thorisson, K.R. and Cassell J. (1997) Communicative Feedback in Human-Humanoid Dialogue. *IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent*. Nagoya, Japan. August 25, 1997

Thurman, D.A. and Mitchell, C.M., (1995) Multi-System Management – The next step after supervisory control ? *Proc. IEEE Int. Conf. On Systems, Man and Cybernetics*, Vancouver, B.C. Canada.

Trumbly, J.E. Arnett, K.P. and Johnson, P.C. (1994) Productivity gains via an adaptive user interface: an empirical analysis. *Int. J. Human-Computer Studies* Vol.46, pp63-81.

Totterdell, P. and Rautenbach, P (1990) Adaptation as a Problem of Design ? In D. Browne, P. Totterdell and M. Norman (Eds) *Adaptive User Interfaces*. Academic Press. London.

Tyler, S.W., Schlossberg, J.L, Gargan, R.A., Cook, L.K., Sullivan, J.W. (1991) An Intelligent Interface Architecture for Adaptive Interaction. In Sullivan, J.W. and Tyler, S.W. (Eds) *Intelligent User Interfaces*. ACM Press.

Tyler, S. W. and Treu, S. (1989) An Interface Architecture to Provide Adaptive Task-Specific Context for the User. *Int. J. Man-Machine Studies* Vol.30, pp303-327.

Uyama, M. (1995) A Three-Step Filtering Mechanism. *Companion Proceedings ACM Conference on Human Factors in Computing Systems, CHI'95*. May 7-11, Denver, Colorado, USA.

Want, R and Hopper, A. (1992) Active Badges and Personal Interactive Computing Objects. *IEEE Transactions on Consumer Electronics*. Vol.38, No.1. February 1992.

Wahlster, W. (1989) Intelligent Interfaces as cooperative agents: from stick shift to automatic transmission in human-computer interaction. *Proceedings of the IFIP 11th World Computer Congress,* Amsterdam, The Netherlands. North-Holland.

Ward, A. Jones, A. Hopper, A. (1997) A New Location Technique for the Active Office. *IEEE Personal Communications* Vol.4, No.5. October 1997. pp42-47.

Wavish, P. and Connah, D. (1997) Virtual actors that can perform scripts and improvise roles. *Proceedings of Autonomous Agents 1997.* Marina del Ray, CA. ACM.

Weizenbaum, J. (1966) Eliza. *Communications of the ACM*, Vol.9. pp36-45.

Wier, G. (1991) Living with Complex Interactive Systems. in *Human-Computer Interaction and Complex Systems*, G. Wier and J. Alty (Eds) Academic Press, London. pp 1-21

Wilson, M. (1997) Metaphor to Personality: the role of animation in intelligent interface agents. *IJCAI'97 Workshop on Animated Interface Agents: Making them Intelligent*. Nagoya, Japan. August 25, 1997.

Winograd, T., & Flores, F. (1986) *Understanding Computers and Cognition*. Ablex Publishing Corporation.

Woods, A., Dey, A., Abowd, G.D. (1997) CyberDesk: Automated Integration of Desktop and Network Services. *Proceedings of ACM Conference on Human Factors in Computing Systems CHI'97*. 22-27 March 1997, Atlanta, GA. USA.

Woods, D.D., Johannesen, L. Potter, S.S. (1991) Human Interaction with Intelligent Systems: An Overview and Bibliography. *ACM SIGART Bulletin* Vol.2, No.5. pp39-50.

Wooffitt, R., Fraser, N.M., Gilbert, N., McGlashan, S. (1997) *Humans, Computers and Wizards: Analysing human (simulated) computer interaction.* Routledge. London.

Wooldridge, M and Jenings, N.R. (1994) Agent Theories, Architectures and Languages: A Survey. *Proceedings of Intelligent Agents ECAI-94.* pp1-22. ISBN 3540588558

Yates, F. E. (1988) Complexity and the limits to knowledge. *American Journal of Psychology*, Vol.4. pp201-204.

Young, R.L. (1991) A Dialogue User Interface Architecture. In Sullivan, J.W. and Tyler, S.W. (Eds) *Intelligent User Interfaces*. ACM Press.

Zachary, W., Le Mentec, J-C, Ryder, J. (1995) Interface Agents in Complex Systems. *Proceedings of Human Interaction with Complex Systems.* Kluwer, Norwell, MA. USA.

Appendix A: Experimental Materials

A1:Experimental Protocol Consent Form

Experimental Protocol Consent Form

The Effect of Dialogue Styles on Human-Computer Interaction

Study Purpose

You are invited to participate in a research study which consists of an experiment to examine how well people can work with certain types of computer software. If you agree to participate you will be one of about 40 people taking part in this experiment.

The purpose of this experiment is to gain an understanding of how changing the way that information appears on a computer screen affects the way that people interact with that information. The experiment is particularly concerned with whether the way in which the computer prompts the user for information is in any way related to the working style and expectations of you, the user.

Procedure

The experiment consists of 3 parts:

1. You will be given a questionnaire to identify elements of your working style
2. You will be asked to complete a task using a piece of software,
3. You will be asked to fill in another questionnaire to provide your comments about how easy to use you found the software.

We estimate that this experiment will require approximately 1 hour of your time.

The experiment is not assessing your ability to interact with a computer, and it is not concerned with finding out your knowledge of computers. You are being asked to participate because you are reasonably familiar with computer use.

Benefits

Although there will be no immediate benefit to you for participating in this experiment, the objective of this research is to make computers easier to use, and to better match the presentation of information to a wide range of people. Your responses may offer valuable insights into how software may be improved.

Confidentiality

Please note that all information gathered from you will be treated as confidential. Your participation is strictly voluntary, and your identity will not be revealed in any presentations or publications that result from this research.

Because of the nature of the experiment, the reasons for each part of the experiment cannot be given in advance, as this may affect the way in which you perform the tasks. However, after the experiment the full reasons will be disclosed.

The results will be available only in a summarized form in which individual responses are combined into averages, and if will not be possible to identify your individual responses from other participants. These results will be available at the end of the research if you would like a copy.

If you have any questions regarding the experiment and your participation in it, please feel free to ask.

John F Meech

Please Turn Over

Conditions

Please read carefully.

Payment

I acknowledge that my participation in this experiment is entirely voluntary and that there will be no payment for my participation

Insurance

I understand that NRC is self-insured against claims or losses or injuries caused by the negligence of NRC or its employees. It has also been explained to me that legal proceedings may be required to determine the validity of any claim which I may have against NRC. I also understand that NRC has no insurance for losses or injuries that are not caused by negligence, and that in the absence of negligence, no compensation may be obtained from NRC for such losses or injuries.

Withdrawal

I understand that the investigator can end my participation in the experimental protocol for financial, scientific of ethical reasons at any time. I also understand that I may end my participation in the experiment at any time or for any reason.

Time for Consideration

I acknowledge that I have been given sufficient time to consider my participation in the experiment.

I understand that the experiment involves research. I hereby confirm that I have received, read and understood all the information above, and give my informed consent to participate in the protocol.

Signed :

____

Date:_______________

A2: Briefing Form
Experimental Briefing Script

"The Effect of Dialogue Styles on Human-Computer Interaction"

Set Up
Experimenter should ensure that the subject has completed and signed an Informed Consent form.
Assign a subject number to the subject and record against name in notebook.

Introduction
Hello, my name is _John Meech__ and I am conducting an experiment to help understand how to make software easier to use by taking into account the preferences of individual users. Thank you for agreeing to help by taking part in this study.

This study investigates how people use certain types of computer software. The data collected will help to make software easier to use and improve our understanding of how to design software.

Part 1
Firstly I would like you to complete a web-based questionnaire to identify certain elements of your personal working styles. When you have finished the questionnaire, please print the last page which shows your temperament or working style and hand it back to me.

[Experimenter should hand subject a Scenario Description/ Instructions at this point.]

Part 2/3
The second part of the experiment requires you to use some software to perform an experimental task.
The software you are about to use is an application for scheduling air travel between cities. In the experiment you play the role of a salesperson who is trying to schedule several flights to multiple cities in Canada and the North-Eastern USA. The journey plan is described in your instructions.

If you have difficulty using the software or understanding what to do at any point, don't feel bad or embarrassed. We are testing the software, not you or your understanding of travel planning.

This session should take about 30 minutes. After using the software, you will be given a questionnaire for you to rate several aspects of your experience in using the software.

Experimenter should check that subject understands what they are going to do, and ask if there are any questions.

Use of Software
Experimenter should ensure that the subject enters their subject number into the start dialogue.

User Feedback Questionnaire
Experimenter should record subject number on questionnaire.

Working Styles Questionnaire
Experimenter should record subject number on questionnaire.

Closing
Thank you again for helping in this experiment.
Do you have any questions or comments before you go ?
If anything does occur to you later please feel free to get in touch using the contact information on the consent form.

A3: Scenario Description

Experimental Instructions **You are subject:**

Part 1

Personal Working Styles

The first part of the experiment requires you to take an on-line questionnaire which provides a rough indication of your "working temperament".
Please point your web browser to:

http://keirsey.com/cgi-bin/keirsey/newkts.cgi

(Please complete the questionnaire first, but then feel free to browse the site later at your leisure)

When you have completed the questionnaire by clicking in the circle next to the choice you agree with, press the "score questionnaire" button. You will receive a page entitled "Keirsey Temperament Sorter II Results". Press the "Print" icon from the Netscape menu bar, and then press "OK" in the print options dialogue. Please return the printed results to John Meech

For the second part of the experiment, please turn over the page…

Scenario Description

You are an Ottawa-based sales representative planning a round-trip from Ottawa to New York, then to Boston, Toronto and back to Ottawa. Today is Monday, October 25. You are due in New York on Tuesday (October 26) to visit a client who is available between 1pm and 3pm. On Wednesday (October 27) you are visiting a client in Boston who is available from 10am to 2pm. On Thursday (October 28) you are due in Toronto to visit a client at 3pm. You fly back to Ottawa on Friday. You would like to avoid having more than one flight each day. Your preferred airline is Air Canada.

Use the software on the computer to search for flights which will enable you to attend all the meetings. Double click on the program icon to start the program, and enter the subject number provided by the experimenter. The software is designed to work in the same way as other desktop computer software, so no instructions are provided for its operation. When you have selected all the required flights, click on the “finished” button and inform the experimenter.

A4: Personality Assessment Questionnaire (from HTML document)

The Keirsey Temperament Sorter II

Keirsey Temperament Web Site: www.keirsey.com

1.In sizing up others do you tend to be
objective and impersonal
friendly and personal

2.Children often do not
make themselves useful enough
exercise their fantasy enough

3.Do you see yourself as basically
thick-skinned
thin-skinned

4.Do you value in yourself more that you are
reasonable
devoted

5.Which appeals to you more
consistency of thought
harmonious relationships

6.Do you think of yourself as a
tough-minded person
tender-hearted person

7.Do you consider yourself
a good conversationalist
a good listener

8.Is it preferable mostly to
make sure things are arranged
just let things happen naturally

9.At a party, do you
interact with many, even strangers
interact with a few friends

10.Waiting in line, do you often
chat with others
stick to business

11.Do you tend to
say right out what's on your mind
keep your ears open

12.Are you more satisified having
a finished product
work in progress

13.Are you more likely to trust
your experiences
your conceptions

14.Facts
speak for themselves
illustrate principles

15.If you must disappoint someone are you usually
frank and straightforward
warm and considerate

16.Are you more inclined to feel
down to earth
somewhat removed

17.Do you prefer contracts to be
signed, sealed, and delivered
settled on a handshake

18.In making up in your mind are you more likely to go by
data
desires

19.Do you feel better about
coming to closure
keeping your options open

20.Is it worse to
have your head in the clouds
be in a rut

21.Do you more often see
what's right in front of you
what can only be imagined

22.Which seems the greater fault:
to be too compassionate
to be too dispassionate

23.At work do you tend to
be sociable with your colleagues
keep more to yourself

24.Are you more frequently
a practical sort of person
a fanciful sort of person

25.Are you inclined to take what is said
more literally
more figuratively

26.Do you find visionaries and theorists
somewhat annoying
rather fascinating

27.In a heated discussion, do you
stick to your guns
look for common grounds

28.Are you more
sensible than ideational
ideational than sensible

29.Do you tend to be more
factual than speculative
speculative than factual
30.Are you more
observant than introspective
introspective than observant
31.Are you inclined to be
easy to approach
somewhat reserved
32.Is clutter in the worksplace something you
take time to straighten up
tolerate pretty well
33.Are you more often
a cool-headed person
a warm-hearted person
34.Is it easier for you to
put others to good use
identify with others
35.On the job do you want your activities
scheduled
unscheduled
36.At work, is it more natural for you to
point out mistakes
try to please others
37.Which rules you more
your thoughts
your feelings
38.In stories do you prefer
action and adventure
fantasy and heroism
39.Are you more interested in
what is actual
what is possible
40.Are you the kind of person who
is rather talkative
doesn't miss much
41.Are you swayed more by
convincing evidence
a touching appeal
42.Are you more comfortable
after a decision
before a decision
43.Do you more often prefer
final, unalterable statements
tentative, preliminary statements

44.Is it your way to
make up your mind quickly
pick and choose at some length
45.Are you more comfortable in making
critical judgements
value judgements
46.Do you prize in yourself
a strong hold of reality
a vivid imagination
47.Do you speak more in
particulars than generalities
generalities than particulars
48.Would you say you are more
serious and determined
easy going
49.Does interacting with strangers
energize you
tax your reserves
50.Do you tend to notice
disorderliness
opportunities for change
51.Which is more of a compliment:
"There's a logical person"
"There's a sentimental person"
52.Are you drawn more to
fundamentals
overtones
53.When in charge of others do you tend to be
firm and unbending
forgiving and lenient
54.Do you prefer to work
to deadlines
just whenever
55.When finishing a job, do you like to
tie up all the loose ends
move on to something else
56.In trying circumstances are you sometimes
too unsympathetic
too sympathetic
57.Do you usually want things
settled and decided
just penciled in
58.Do you think of yourself as
an outgoing person
a private person

59.When the phone rings do you
    hurry to get it first
    hope someone else will answer
60.Do you like writers who
    say what they mean
    use metaphors and symbolism
61.In most situations are you more
    deliberate than spontaneous
    spontaneous than deliberate
62.Are you inclined to be more
    hurried than leisurely
    leisurely than hurried
63.With people are you usually more
    firm than gentle
    gentle than firm
64.Common sense is
    usually reliable
    frequently questionable
65.Do you tend to choose
    rather carefully
    somewhat impulsively
66.Are you prone to
    nailing things down
    exploring the possibilities
67.Is it worse to be
    a softy
    hard-nosed
68.Which do you wish more for yourself:
    strength of will
    strength of emotion
69.Is it better to be
    just
    merciful
70.Are you more
    routinized than whimsical
    whimsical than routinize

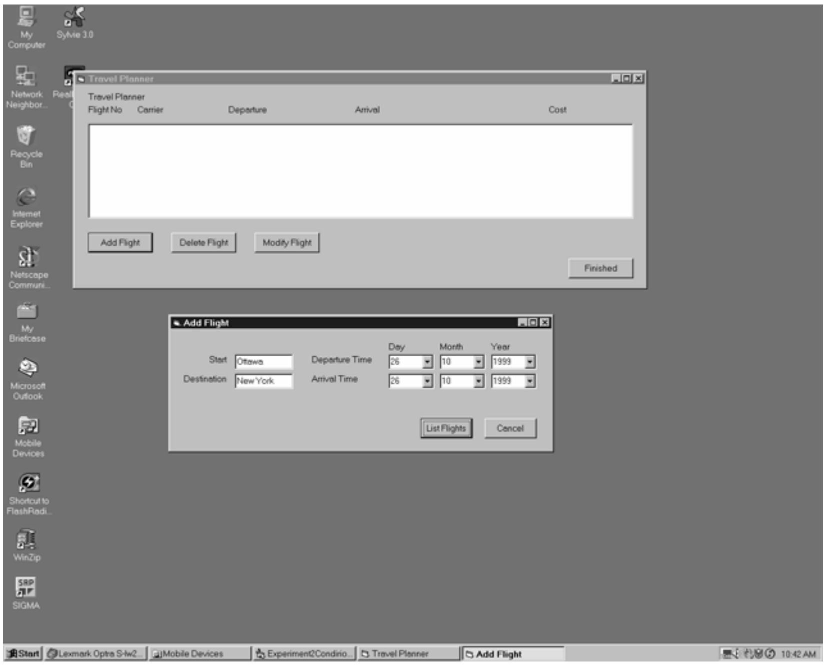
CGI Keirsey Temperament Sorter II by David M. Keirsey
Keirsey Temperament Sorter by Dr. David Keirsey

A5: User Feedback Questionnaire for the Computer Dialogue Styles Experiment

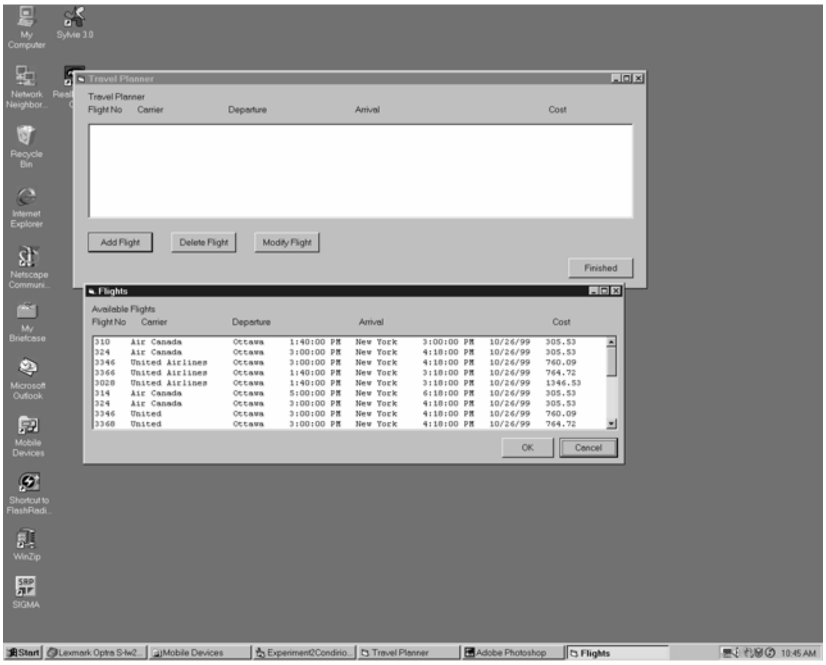
Please rate your satisfaction with the system. Try to respond to all items. Show your response by circling a in a number along the scale from "Strongly Agree" to "Strongly Disagree".

1.	It was simple to use this software	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
2.	I felt comfortable using the software	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
3.	Information provided by the software was easy to understand	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
4.	The interface to the software was pleasant	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
5.	I could anticipate the behaviour of the software	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
6.	Prompts for me to take action were clear	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
7.	The software informed me about its progress	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
8.	I enjoyed using the software	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
9	I knew what to do next at each point	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree.
10.	It was easy to recover from mistakes	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
11.	The software behaved predictably	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
12.	It was easy to learn how to use the software	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
13.	I found the software friendly to use	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
14.	The software let me decide what to do next	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
15.	The software let me work at my own pace	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
16.	The software required me to input too much information	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
17.	I understood the terminology used by the software	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
18.	The software guided me about what action to take	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
19.	I found the software frustrating to use	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree
20.	Overall, I was happy with the software	Strongly Agree 1 2 3 4 5 6 7 Strongly Disagree

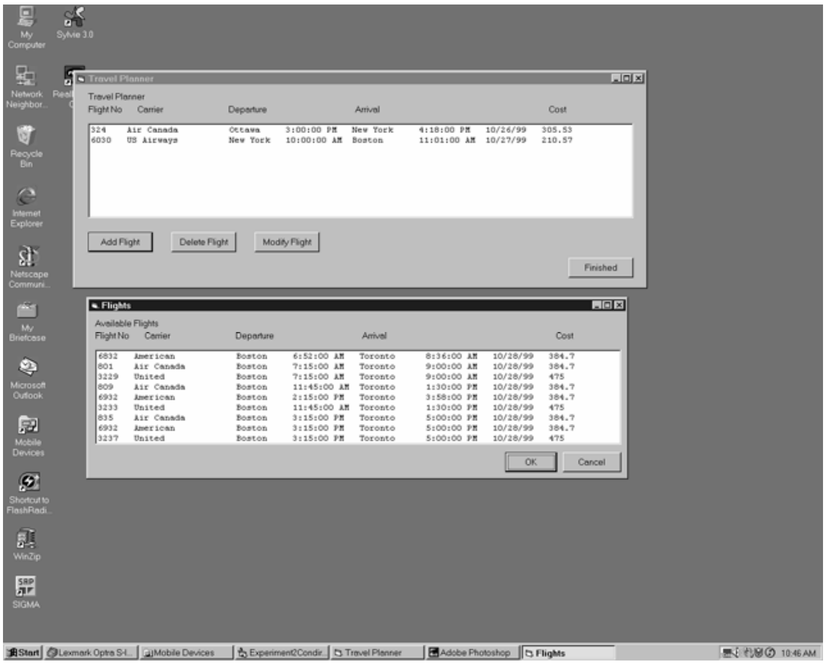
A6: Experimental Program Interface (Experiment 1, Condition 1)



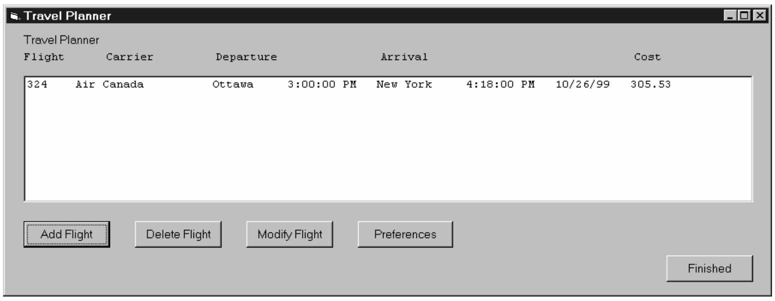
A6.1 Flight selection interface



A6.2 Flight listing



A6.3 Putting selected flights into the itinerary



A6.4 Main Form in Experiment 2 showing the "preferences" button.

A6.5 The preferences form in Experiment 2

Appendix B
Experimental Results

B1 Example Software Data Log

```
Exp1 Cond1    14
addflight on mainform        9.1
list flights from add flight form          23.2
no matching flights from add flight form   23.2
accepted a flight listed on listflights    27.7
no flight selected from listflights        27.7
cancel on listflights        31.3
addflight on mainform        34.1
list flights from add flight form          96.3
no matching flights from add flight form   96.3
cancel on listflights        103.5
addflight on mainform        108.7
list flights from add flight form          129.7
selected flight in list on listflights     187.6
accepted a flight listed on listflights    198.7
318  Air Canada Ottawa 7:40:00 AM  New York 8:58:00 AM10/26/99   305.5
addflight on mainform        200.8031
list flights from add flight form          230.0
selected flight in list on listflights     310.3
accepted a flight listed on listflights    319.2
6030  US  Airways  New  York  10:00:00  AM  Boston  11:01:00  AM  10/27/99
210.5
addflight on mainform        321.6
list flights from add flight form          344.6
selected flight in list on listflights     375.8
accepted a flight listed on listflights    392.2
809  Air Canada Boston 11:45:00 AM Toronto 1:30:00 PM 10/28/99  384.7
addflight on mainform        407.1
list flights from add flight form          435.3
selected flight in list on listflights     443.2
selected flight in list on listflights     457.6
selected flight in list on listflights     488.3
selected flight in list on listflights     546.0
accepted a flight listed on listflights    548.6
134 Air Canada Toronto 11:55:00 AM Ottawa 12:51:00 AM 10/29/99  249.5
selected flight on mainform  552.6
modifly flight on mainform   554.0
list flights on modifyflightform           559.3
cancel on modifiedflightlist               610.4
cancel on modifyflightform   612.5
finished on mainform         641.2
318 Air Canada Ottawa  7:40:00 AM New York 8:58:00 AM 10/26/99 305.5
6030 US Airways New York 10:00:00 AM Boston 11:01:00 AM 10/27/99 210.5
809 Air Canada Boston 11:45:00 AM Toronto 1:30:00 PM 10/28/9  384.
134 Air Canada Toronto 11:55:00 AM Ottawa 12:51:00 AM 10/29/99   249.5
```

Each line represents a dialogue interaction, with a time in seconds from start. If the dialogue operation selected a flight, the flight details are also given. If a dialogue error occurs, the dialogue description starts with a "no…" (e.g., line 3: "`no matching flights from add flight form    23.2`"). Each line (excluding flight details) therefore represents one dialogue exchange. Dialogues are counted and this data is the *Number of Dialogues* and *Broken Dialogues* count in the results.

B2 Experimental Results

Subject	Condition	Time (secs)	Dialogues	Errors	Log(Time)
1	1	380	35	3	5.940171
2	1	549	65	1	6.308098
3	2	399	21	0	5.988961
4	2	949	69	16	6.855409
5	2	1161	58	0	7.057037
6	1	728	46	1	6.590301
7	3	559	35	1	6.326149
8	1	460	18	0	6.131227
9	1	406	65	1	6.006353
10	1	613	31	1	6.418365
11	2	517	26	0	6.248043
12	2	807	44	16	6.693324
13	3	681	65	5	6.523562
14	1	641	32	3	6.463029
15	2	780	38	9	6.659294
16	2	612	52	0	6.416732
17	1	407	32	5	6.008813
18	2	807	44	0	6.693324
19	2	468	24	0	6.148468
20	1	643	32	3	6.466145
21	4	1101	50	1	7.003974
22	1	271	53	8	5.602119
23	3	916	33	0	6.820016
24	2	549	38	0	6.308098
25	3	333	30	0	5.808143
26	3	803	56	1	6.688354
27	3	320	27	0	7.211557
28	3	777	76	3	6.65544
29	4	413	17	1	6.023448
32	4	815	35	2	6.703188
33	4	481	65	14	6.175867
34	4	655	54	12	6.484635
35	4	379	34	2	5.937536
36	3	418	44	4	6.035481
37	4	780	40	12	6.659294
38	3	538	46	0	6.287858
39	3	417	40	0	6.033086
40	4	629	57	0	6.444131
41	4	657	87	34	6.487684
42	4	367	35	4	5.905362

Appendix B2.1: Performance Results

SUB	COND	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
1	1	2	5	2	6	4	3	4	6	3	2	3	3	5	2	2	4	2	6	2	5
2	1	1	1	1	1	1	1	1	2	1	1	2	1	1	1	1	7	2	2	7	1
3	2	5	7	1	6	5	2	4	7	2	3	2	2	6	1	1	4	1	1	1	7
4	2	3	4	6	7	4	6	6	7	4	4	4	4	7	6	1	2	1	3	7	6
5	2	2	2	5	4	2	1	1	4	1	2	3	1	3	6	2	1	1	2	3	4
6	1	6	5	6	7	4	6	7	7	6	5	6	3	4	2	2	6	6	6	4	7
7	3	2	5	4	6	2	2	5	6	3	2	1	2	6	6	1	2	3	6	3	6
8	1	3	3	2	3	4	4	3	3	4	4	2	2	3	2	2	5	2	6	4	3
9	1	6	5	5	7	2	5	7	6	4	3	6	2	6	6	2	6	6	3	2	2
10	1	3	6	2	5	3	1	1	6	3	6	3	3	5	1	1	1	5	3	2	6
11	2	2	2	5	6	2	3	6	3	4	2	1	2	4	3	2	3	3	6	4	2
12	2	5	4	5	5	4	6	6	4	6	6	4	4	5	3	3	5	3	6	4	5
13	3	3	3	5	6	4	4	4	4	3	6	4	3	3	4	2	6	3	5	5	3
14	1	2	2	3	6	3	3	2	5	3	4	3	3	5	3	3	2	2	3	3	4
15	2	6	6	6	6	3	6	6	6	6	6	3	3	6	4	2	2	4	3	2	6
16	2	2	4	4	5	3	3	3	6	5	3	3	2	5	4	1	6	2	6	3	6
17	1	4	3	5	5	3	6	6	5	5	3	3	2	5	5	3	5	4	6	4	4
18	2	4	4	5	6	3	4	4	5	2	3	2	1	4	4	2	3	2	3	6	4
19	2	5	5	5	5	3	3	5	3	3	5	5	3	3	3	3	1	3	3	5	6
20	1	2	1	7	4	2	6	6	4	6	4	3	2	4	3	1	7	2	6	7	2
21	4	5	3	6	4	4	6	4	4	2	2	4	5	4	2	1	6	3	6	5	5
22	1	3	2	3	5	3	4	4	3	3	5	2	2	5	4	1	7	1	3	3	2
23	3	2	2	4	4	3	4	6	3	3	2	2	2	2	2	1	6	2	3	6	2
24	2	4	4	5	6	5	5	5	5	5	3	3	3	6	4	2	6	7	5	4	3
25	3	6	5	4	6	4	7	6	7	6	3	3	3	5	3	3	4	2	7	5	6
26	3	4	3	3	2	5	4	4	2	2	4	4	6	5	4	7	5	7	4	5	3
27	3	3	2	3	4	2	2	3	3	2	2	3	2	3	2	2	6	2	5	4	3
28	3	5	4	5	5	3	6	7	5	4	3	2	2	4	2	2	4	4	6	2	5
29	4	3	3	3	3	4	5	4	4	5	5	4	3	4	4	4	6	4	5	5	4
32	4	5	7	5	6	6	4	5	7	6	3	5	3	7	3	1	3	1	7	7	7
33	4	6	7	7	6	5	6	7	6	7	5	3	4	7	6	5	4	3	6	1	7
34	4	4	2	5	3	6	4	3	2	7	3	4	4	4	4	2	5	4	6	5	3
35	4	2	2	5	4	2	2	7	5	1	1	1	1	3	2	2	7	2	6	6	5
36	3	3	5	5	5	4	4	3	4	4	4	4	4	4	5	6	5	5	4	4	4
37	4	2	3	4	4	4	4	6	3	4	4	5	6	4	2	1	7	1	2	7	3
38	3	2	2	2	2	2	2	2	3	2	2	3	2	2	2	2	6	2	2	6	2
39	3	1	1	2	3	2	1	1	2	1	1	1	1	1	1	1	7	1	5	7	1
40	4	3	4	3	3	4	2	2	4	3	5	3	4	4	2	2	6	2	4	5	5
41	4	4	3	2	5	3	3	7	5	7	7	5	3	3	3	1	7	3	4	2	6
42	4	5	5	5	6	3	5	5	6	1	3	3	3	6	7	1	3	3	3	2	6

Appendix B2.2: Questionnaire Results

Subject	Condition	IE	SN	TF	JP	IE Score
1	1	I	N	F	P	8
2	1	I	N	F	J	0
3	2	I	S	F	J	8
4	2	I	S	T	J	4
5	2	I	N	F	J	0
6	1	E	N	F	J	-6
7	3	E	N	F	P	-4
8	1	I	N	T	J	6
9	1	E	S	T	J	-8
10	1	I	S	T	J	8
11	2	E	N	T	P	-4
12	2	E	S	T	P	-6
13	3	I	S	T	P	6
14	1	I	N	T	P	4
15	2	I	N	T	P	10
16	2	I	S	T	J	0
17	1	E	N	T	P	-6
18	2	E	S	F	P	0
19	2	E	S	T	J	-4
20	1	E	S	T	J	-4
21	4	I	S	T	J	2
22	1	E	N	F	J	-4
23	3	E	S	T	J	-2
24	2	E	S	T	J	-2
25	3	E	N	F	P	-4
26	3	I	S	T	J	4
27	3	I	S	T	J	2
28	3	E	S	F	J	-6
29	4	I	S	T	J	4
32	4	E	N	F	J	-2
33	4	I	N	T	J	10
34	4	E	S	F	J	-8
35	4	E	S	T	J	-8
36	3	E	N	F	J	-6
37	4	E	S	F	P	-6
38	3	I	S	F	J	2
39	3	I	N	F	J	6
40	4	I	N	F	J	2
41	4	I	S	T	J	6
42	4	E	S	T	J	-4

Appendix B2.3: Personality Results