DISOBEDIENCE AND AMBIGUITY IN THE USER INTERFACE

Janet C Read	Stuart MacFarlane	Chris Casey
Department of Computing	Department of Computing	Department of Computing
University of Central Lancashire	University of Central Lancashire	University of Central Lancashire
Preston, Lancs	Preston, Lancs	Preston, Lancs
JCRead@uclan.ac.uk	SJMacFarlane@uclan.ac.uk	Ccasey@uclan.ac.uk
www.uclan.ac.uk	www.uclan.ac.uk	www.uclan.ac.uk

ABSTRACT

This paper describes the features that distinguish recognition based computer interfaces by considering the interaction process and using a simple model to assist in the definition of some of the terms associated with these interfaces.

With reference to the user experience, Disobedience, Ambiguity, and Error-Proneness are described. The authors suggest that disobedience is a useful term that separates recognition technologies from other input technologies.

Whilst acknowledging that disobedience is very difficult to eliminate, practical strategies are proposed which can limit the causes of disobedience in recognition based interfaces.

Keywords

Disobedience, Errors, Recognition technology; Ambiguity, Interaction.

1. INTRODUCTION

The term ‘disobedient interface’ was first used in (Snape, Casey, MacFarlane, & Robertson, 1997) to describe those interfaces that are inherently erroneous, typically associated with recognition software such as is used in speech, handwriting, and gesture input. These same interfaces are variously described as ambiguous, brittle and error prone (Mankoff, Abowd, & Hudson, 2000), (Oviatt, MacEachern, & Levow, 1998).

The approach taken in this paper is to firstly present ‘disobedience’ as a concept and then to describe empirical research that has been carried out to ascertain how the user discerns and deals with disobedience in an interface. Discussion then follows, which illuminates disobedience and conclusions are drawn together with ideas for further study.

2. DESCRIBING DISOBEDIENT SYSTEMS

Whereas it is possible to describe a system as disobedient, it is in fact the interactions supported by the system that makes it appear disobedient. This can be demonstrated using the analogy of an Adult – Child interactive system. In this analogy, the adult would represent the user, the child the computer. ‘Dialogue’ between them would make the system interactive. This dialogue would take place using a language that was understood by both child and adult. Components of the dialogue would include speech, hearing, visual clues and contextual clues. An interaction would be the combined processes of intention by the adult, articulation by the adult, hearing by the child, action by the child.

In an interactive system, the user identifies a task to be done, he communicates this to the computer which then interprets the user articulation and then provides feedback.

2.1 Errors and Ambiguity

Within any system there is capacity for error. User detected errors are referred to as ‘gulfs of evaluation’ by (Norman, 1988). This term is used to describe those instances when the user does not ‘see’ what they ‘expected’.

An interaction is ambiguous if it ‘has more than one possible meaning, is doubtful or uncertain’. We define ambiguity to be that feature by which there is more than one interpretation of the data as presented. (Mankoff et al., 2000) place ambiguity in this same context, a product of conflicting interpretations of an input

2.2 Classifying Errors

In the following example, the adult wants the child to feed the cat with a piece of fish that has been bought for its supper. This is the intention of the adult. The interaction between adult and child would be considered successful if the child fed the fish to the cat.

Adult says	Error type
Feed the fish with some	Slip

cat	
Put the cat out	Mistake

There are two erroneous actions that the adult can make. Instead of saying ‘Feed the cat with some fish’, the adult may be distracted and say ‘Feed the fish with some cat’; this would be described as a slip. However, had the adult said ‘Put the cat out’, that would be considered to be a mistake as the intention was flawed [Lewis, 1986 #151].

Irrespective of the action of the adult, the child may not feed the cat with the fish. She may have only heard a part of the instruction e.g. ‘Feed the fish’, she may have misinterpreted the instruction, perhaps feeding the pet goldfish and the cat at the same time, and she may do nothing or turn up the TV! The two parties of the interaction are not wholly separable; as the child may misinterpret an incorrect articulation by the adult and by some fluke, the interaction may succeed.

2.2 User Responses to Erroneous Interactions

Consider the user – computer system. The user requires the computer to do something. The computer doesn’t do it. At this point the user will wonder what went wrong; possible explanations may include;

- The user made an error
- The computer made an error

Considering the cause of the breakdown of the interaction can expand this simplistic ‘blame based’ model

When a user encounters an error, he or she will attempt to establish the cause before attempting a repair. If the computer makes no response, the user will assume that the computer is somehow faulty, and will probably repeat the action that has been ‘ignored’. At this point the user has attributed blame to the computer. This would be considered to be a significant computer breakdown. Research has shown that this is the case in most non-recognition based systems. However, as soon as the computer responds in some way, the user is left guessing as to why the response does not match their expectation.

User observations have concluded that there is a hierarchy of user error explanation whereby the blame progresses from invalid articulation (user mistake or slip), through to ambiguous articulation (language of the articulation / user fault) Beyond this, the user blames the system.

This model allows us to consider three ‘causes’ of error, these being;

Wrong articulation by the user

Ambiguous articulation by the user

Computer makes a mistake

The model is deliberately simplified in a way that suggests that valid articulations can only be subdivided into ambiguous and non-ambiguous articulations. This is not the case, as other classifications could be applied. However, as this discussion is about disobedience and ambiguity, this is a useful division.

The mistake by the computer could be caused by the hardware or software ; these could be separated.

This model clarifies why it is that recognition technologies are generally more error-prone than others. The natural language used to articulate to them is likely to be ambiguous, the recognition algorithms used by the software are error-prone and the hardware that captures analogue signals and turns them into digital data is likely to fail. Compare this with a …………………….

As alluded to in an earlier paragraph, an interaction could comprise a number of errors; for instance, it is possible that an ambiguous articulation could be unheard by the computer. Dyads of articulation / reception are created which the user has to interpret in order to proceed with the interactive process.

	User	Computer	User state
A	Valid, non-ambiguous articulation	Gets it right	Happy
B	Valid, non-ambiguous articulation	Gets it wrong	Unhappy

C	Valid ambiguous articulation	Gets it right	Happy
D	Valid ambiguous articulation	Gets it wrong	Puzzled
E	Invalid articulation	Gets it right	Happy
F	Invalid articulation	Gets it wrong	Puzzled

It can be seen from this that the user is only unhappy if they have considered that they have made a valid, non-ambiguous articulation that the system has got wrong. The state A would be described as 'robust', obedient. States B, D, and F are all erroneous. (States C and E could also be described as erroneous, but to the user they have resulted in correct interactions) If the user is aware of his intentions and actions, he can rationalize states D and F. In these instances the informed user will apportion blame on themselves. State B is different. The user cannot understand why the interaction has failed. This state is described as disobedience by the authors of this paper.

A system that is disobedient is one with the potential for disobedient interactions. The system causes disobedient interactions.

Ambiguous systems have potential for ambiguous interactions. 'Flaws' in the language of the interaction cause ambiguous interactions.

Error-prone systems have the potential for error-prone interactions. Erroneous interactions can be caused by the user, the system or the language of the interaction.

{Disobedient systems} $\subset$ {Error-prone systems}

{Ambiguous systems} $\subset$ {Error-prone systems}

3 THREE RESEARCH QUESTIONS FOR RECOGNITION BASED INTERFACES

1.Can the system 'deafness' be cured? Just as a child may not hear an adult; so the hardware in a system may fail to 'hear' the user articulation. However, the case where the child hears but does not listen is more problematic. This system 'listening' can be enhanced by improving the user articulation (training);

2.Can the system 'ignorance' be cured? The system can only be as 'smart' as the software put into it. In the same way that a child has a limited understanding of the adult world, so, a recognition interface has a limited understanding of the human user. There are two strategies that are useful here. The first is to simplify the user; the second is to educate the system. In a recognition based interface, this may be done by limiting the language that is used by the user, or by improving the algorithms that are used by the system.

3.What about ambiguity that masquerades as disobedience? The difficulty for any interactive system is the 'I thought you meant' breakdown.

4. CONCLUSION

We have examined the notion of disobedience as a user-perceived notion. The interaction process for disobedient systems has been described and a range of interaction descriptors has been discussed.

There are three strategies that could be employed by users and system designers to minimize disobedience within recognition based systems.

These are;

- Enabling the user to modify the system profiles to allow for the individual identification of non-unique articulations
- Optimising the design and positioning of data capture devices in order to minimise noise.
- Informing the user of the defining characteristics of the data set being recognised, in order that they can adapt their articulation to maximise the chance of good recognition.

Further work is planned in this area, as we investigate multi-modal interfaces and consider how they can be modelled and determine whether or not the attributes and strategies referred to in this paper can be recycled in a more complex interaction. .

REFERENCES

Mankoff, J., Abowd, G., & Hudson, S. (2000). OOPS: a toolkit supporting mediation techniques for resolving ambiguity in recognition-based interfaces. Computers and Graphics, 24, 819 - 834.

Norman, D., A. (1988). The Psychology of Everyday Things. New York: Basic Books.

Oviatt, S., MacEachern, M., & Levow, G. (1998). Predicting hyperarticulate speech during human-computer error resolution. Speech Communication, 24, 87 - 110.

Read, J. C., MacFarlane, S. J., & Casey, C. (2001). Measuring the Usability of Text Input Methods for Children. Paper presented at the HCI2001, Lille, France.

Snape, L., Casey, C., MacFarlane, S. J., & Robertson, L. (1997). Using Speech in Multimedia

Applications. Paper presented at the TCS Conference, Bangor, Wales.

COLUMNS ON LAST PAGE SHOULD BE OF EQUAL LENGTH AND SHOULD LEAVE AT LEAST 5CM BLANK