Dealing with Undo

Roberta Mancini*, Alan J. Dix⁺ and Stefano Levialdi*

* Dipartimento di Scienze dell'Informazione, Università degli Studi di Roma "La Sapienza", Via Salaria 113, 00198, Rome, Italy

⁺ School of Computing, Staffordshire University, PO Box 334, Beaconside, Stafford ST18 0DG, United Kingdom

e-mail address: roberta@dsi.uniroma1.it, A.J.Dix@soc.staffs.ac.uk,
levialdi@dsi.uniroma1.it

ABSTRACT: In this paper we are going to take into account non-determinism in human-computer interaction. When a user is in control on the dialogue, that is he knows the system state and the system behaviour after any performed action, then the interaction is deterministic. Generally, an undo helps the user in *handling* non-determinism when he is employing such function as a navigational tool, and/or in reducing non-determinism when he is employing it as a recovery function. Unfortunately, not all functions at the application level may be undone; moreover, an undo may behave differently in different situations. As result, we have that the undo also adds risk along the interaction: the more powerful the system, the higher the risk!

KEYWORDS: HCI, undo, non-determinism, dialogue, granularity, PIE model.

1. Introduction

Predictions have always been very important to humans. This happened and happens because, in order to predict something, it is necessary to know the object of the prediction, and such a knowledge is, in some sense, a form of control, of power. For a human, a prediction about something is the expression of his power on that thing. It is possible to do very precise predictions in a closed world, accounted by mathematically expressed physical laws, because these are independent from the human behaviour (Barrow, 1988). But when humans are involved in something, it is not possible to do predictions any more. Human behaviour follows no physical law. Since humans play

a role in HCI and they are unpredictable, the resulting interaction is non-deterministic. In this paper, we analyse non-determinism in interaction and we identify in the undo function a way to handle and reduce non-determinism.

2. Undo and non-determinism

While interacting with a computer, any user would like to feel in control on the dialogue. In order to do this, Cole, Lansdale and Christie (Cole 1985) suggested that any user should be able to answer the following questions: *Where am I? How did I get here? What can I do? Where can I go next?*

Analysing the answers of the four above questions we may note that, in order to really feel in control of his dialogue, the user basically needs to know the system state and the system behaviour after any one of his actions. When using interactive systems, the knowledge on system state and behaviour may be increased by employing the undo. In fact, not only undo allows the user to repair an error, if it occurs, by reaching a past state, but it also enables the user to handle and reduce non-determinism. More precisely, when a user is navigating through a system, exploiting the exploratory aspect, typical of interactive systems, he is enriching his knowledge about the system's behaviour. In this case he *handles* non-determinism. However, his knowledge is related to the external system behaviour, not to the internal system structure or status. Instead, if there is a mismatch between the user's and system's model, then an inexact matching between the interpretation of the user's action from the system's point of view and from the user's point of view occurs. In this case, the undo functions may provide more information on the internal system structure, so *reducing* non-determinism (Mancini 1996).

Often, undo is a very powerful tool, allowing the user to repair an error, but sometimes it may add confusion to the user while interacting. How many times have we tried to perform undo and, although present on the interface, such function was not allowable? Or how many times have we performed undo obtaining something different from what we expected? Such confusion is due to the fact undo is not as easy as it looks; in fact, it is a very complex function and we cannot simply say that it cancels the effect of the last performed action(s). The last action with respect to what? Last from the user's point of view, that is the last action in the interaction (the last user performed action), or from the computer's point of view, that is the last action in the computation, the last action which modified the result? For example, if working with a word processor a user types a word, then uses the scrollbar and then applies undo, which action will be cancelled? The typing, the scroll or both? The incomplete knowledge about what and how undo acts, makes its behaviour unpredictable, so adding non-determinism in interaction. An exhaustive discussion about different undo semantics is provided in (Yang 1988, Mancini 1996, Mancini 1997).

2. Undo and risky interaction

The undo function is useful to reduce the risk in interaction (Dix 1996). But what is this risk? The first kind of risk may be due to the presence of user's errors (slips) while interacting, so the undo can help the user in recovering from errors. Moreover, the other risk is due to the mismatch between the user's and system's point of view on the same action.

Anyway, not always the presence of undo assures a reliable interaction: in fact, it often can make a system more risky when, after performed an undo, the system behaves differently from the foreseen way. This means that the behaviour of this system function may add non-determinism in interaction when it is slightly inconsistent. But does it matter? For 99% of the time undo works, this is better than systems where undo was either absent or frequently did not work. Or is it? If undo works almost all the time, then users may become used to it. Indeed, one of the advantages of having undo is that users can take a more exploratory approach to their interaction, trying out possible courses of action, but then retracing their steps if negative results occur, i.e. their behaviour is more risky. But if they become more risky then they are more likely to do things that need undo, and if undo does not work uniformly then they will have problems which would never have occurred if they had been more careful. So although a 99% effective undo may be better than having no undo at all, there will be errors that occur when you have undo which would not have happened if you had never had undo at all!

8. Conclusions

In this paper issues related to non-determinism in interaction have been investigated. It has been stressed that undo function allows the user to feel more in control in his computer dialogue. In particular, such a function allows the user to handle and reduce non-determinism when it is used as navigational tool and recovery function respectively. Even if the undo allows the user to handle and reduce non-determinism, it is not always clear how and on which objects it acts. This incomplete knowledge on its behaviour adds, by itself, non-determinism to interaction, making it more risky.

9. References

- [Barrow 1988] J.D. Barrow, *The world within the world*, Oxford University Press, 1988.
- [Cole 1985] I. Cole, M.W. Lansdale and B. Christie, Dialogue Design Guidelines, in *Human Factors of Information Technology in the Office*, B. Christie ed., pp. 212-241, Wiley Inc.

[Dix 1996] A.J. Dix, R. Mancini and S. Levialdi, Alas I have undone- Reducing the risk of interaction?, in *Ancillary Proceedings of HCI'96*, A. Blandford and H. Thimbleby eds., pp. 51-56, The British HCI Group.

[Mancini 1996] R. Mancini, A. Dix and S. Levialdi, Formal and Informal Definitions of Undo, Research Report RR9611, School of Computing and Mathematics, University of Huddersfield, UK, 1996.

[Mancini 1997] Modelling Interactive Computing by exploiting the Undo, PhD thesis IX-97-5, Università degli Studi di Roma "La Sapienza", 1997.

[Yang 1988] Y. Yang, Undo support models, *International Journal of Man-Machine Studies*, N° 28, pp. 457-481, 1988.