

Mundane intelligence in interaction design: A study of widely used adaptive user interfaces

ABSTRACT

In this paper we describe an ethnographic study of *Microsoft Office* users. The objective was to see how use of so-called intelligent user interfaces, exemplified by the tailorability and adaptive aspects of this software package, unfolded in practice. We discovered that designers' intuition about which aspects of "intelligence" users might benefit from, often fails. The paper suggests some relatively simple and specific real-time adaptations to concrete use patterns, which may increase the "perceived intelligence" of the user interface. The change of users' work mode induced by the system itself serves as a criteria to determine when such adaptation is needed.

INTRODUCTION

We know that much software offers an abundance of functionality, a lot of which is never used. For every new version, more functionality is added. Software vendors see this as necessary in order to keep abreast with the competition. On the other hand, learning the software becomes harder than before [6] and relatively less of the functionality is used. Microsoft have found that 95% of their 400 million registered users used less than 5% of the functionality¹. As a response to this situation, even more functionality is added to enhance the user interface, making it more easily understood and enjoyed by users. It is provided with "intelligence" [3]. The problem might be that this intelligence is not recognized and adopted, either.

The topic of intelligent interfaces has received a lot of attention [1-5]. The definition used in this paper is that the system itself shall adapt and optimize its interface so that it matches the current work processes and preferences of users. We are not going to be able to cover pertaining literature here, but it does not seem to be comprising any ethnographic studies of so-called intelligent user interfaces. Thus, this paper is a novel contribution. It shows in practice how users deal with intelligence in the user interface of common office applications. Our findings

can be used to guide new research to improve user interface design, intelligent or not. Our point of departure is mundane, trivial, everyday usage of Microsoft Office products, which implements quite a lot of functionality which we see as fulfilling the definition of intelligent user interfaces. Currently Microsoft aim to incorporate even more mechanisms that will be sensitive to concrete and individual use contexts than previously has been the case. Therefore, it is a standard software package that is readily the best place to do empirical studies of intelligent software.

RESEARCH METHOD AND SITE

We have chosen to base our studies on observations, and interpret them within a framework of ethnography [7]. The objective of our observations was to describe a representative amount of in-context user interaction with standard office applications. We wanted to be able to separate extraneous interruptions and delays from those eventually imposed by interaction with adaptation mechanisms. Moreover, we wanted to map the role and character of intelligent action by the interface between discrete application packages, vs. the articulation work thus carried out by the user.

OBSERVATIONS

We observed Microsoft Office usage in two companies. One has 280 employees. They manage patent- and intellectual property rights for various stakeholders. The department we worked with was responsible for the preparation of presentations as well as process descriptions. They relied exclusively on the Microsoft Windows/Office 2000 combination. The second is a private insurance company with approximately 600 employees worldwide. The department of our observations had about 30 people, most of which were working with the Microsoft Windows 2000/Office 2003 combination.

This is one excerpt of one user (K26) working interchangeably with Word, Excel, PowerPoint, Internet Explorer and Novell Groupwise. They were all maximized at the same time, and she used the mouse to click on the task-bar (rather than *shift-<tab>*, to alternative between the applications. K26 was working on

¹<http://www.microsoft.com/products/ceip/en-us/default.aspx>

a document together with a group of colleagues. The document consisted of a mixture of text with headings, and a table with three columns and about 20-30 rows. The data in the table represented workflows and meetings, which had been or were to be held. K26 gathered data from emails that she had received or from the web, and pasted them into the pertaining columns in the table.

Internet Explorer, which was used as the browser, as well as the email software, supports text styles, which meant that almost every time she copied text over to the target document, the text formatting properties of the original document were incorrect. In one instance, when this happened (again), it became clear that she was not surprised, only annoyed. She “knew this was going to happen.” The pasted text had the wrong font; one word kept its *italics* and it was the wrong size.

The problem was solved in the following fashion:

1. First she used the mouse to select the wrongly formatted paragraph.
2. Next, she changed the font size. Again, the mouse was used to pick the correct size from the drop-down menu on the toolbar.
3. She selected and deselected all three font formatting options on the toolbar.
4. Finally she changed everything to the right font. Again, she used the access option via the toolbar.

All of this was carried out in a smooth and efficient fashion. However, a lot of this layout work thus was redundant and sometimes the internalized routine of performing the procedure in the same way every time made her miss that the text was accidentally de-selected, e.g., when she dislodged the mouse.

During our observation, this participant relied on four different date formats in the same document²:

1. September, 07
2. 9/21/07
3. 21-Sep-07
4. 21 September 2007

In those cells of the table that had had much text, date format 2 and 3 was used. She only used date format 4 once, to register the creation date of the document. Date format 1 was used in some cells, which had very little text

²These samples illustrate the used formats, but are translations into international date formats. The original format were, e.g., 02-februar-2006, and this is also the correct date of this particular excerpts from the fieldwork.

in them from before. Every time she typed the dates in any of the formats 1, 2 or 4, Microsoft Word displayed an auto text suggestion, which was on the format <Year-Month-Day>. This was ignored, every time. The proposed format is the inverse of the common national format, which may explain why the auto text was not accepted.

A little later, the participant typed the name of a country into the same cell, but she did not want it starting with an uppercase letter (for some reason). Every time, Word first corrected it to start with uppercase, until she deleted the word again and retyped it. “Why is this happening,” she wondered. “I do not understand why this happens with the name of the country. When I type the name of the capital without an uppercase first letter, then it leaves it like I want to.” The reason was that the country name was entered in the autocorrect list of words, which keeps track of the words that Word shall amend as you type. This is an option that can be toggled on or off. The name of the capital clearly was not in the list. The user seems to interpret this as a semantic capability, on the other hand.

The next user that we observed was a male of 31 (M31), who did not use any short cuts or function keys. All text copying was invoked clicking the right mouse button. Since he kept browsing menus and relied heavily on tool-tips, we assume that he was not a proficient user. One very simple type of “intelligent” user interface, which is the expanding menus, seemed to confuse this user and he often did not find the functionality he was looking for. This participant was working on many different types of documents, and the excerpt below is from the creation of a new version of a PowerPoint presentation for a seminar. The data had been typed in previously. Similar seminars had been conducted earlier as well, and several versions of this presentation had been prepared for them. The current document could therefore fetch text from three different sources:

1. The agenda plus other practical information was in an email.
2. The authors and title/abstract for the talks were published on a website.
3. Earlier versions of the presentation, as a reference for the layout work

The user started by opening a presentation from the server (3). It was the corresponding presentation from the year before. He read quickly through it and started editing the text. He changed the first heading, but he decided that amending the previous version to this year’s seminar would take too long. The new presentation was supposed to be short (around 5 pages) while the old one was long (18 pages). He started a new PowerPoint instance (rather than creating a new presentation within the current one)

and then noticed that it came up with an empty document, which was based on the default template. Then he looked at the previous one again, before he opened the pertaining email with the title of the entire seminar and the agenda.

The front page of last year's presentation had a picture covering the entire background. It had a company logo in the top left corner, and the rest was a graded dark blue. The front-page text was white on blue background. There was a large textbox in the middle, for the title of the seminar, and a smaller one below for customer/occasion, place and date. When the user selected the large textbox to change the title of the seminar, the text was inverted, but since it was white on blue to start with it now became white on white. The user seemed un-surprised by suddenly having to work with invisible text. He finished amending the title and then clicked on a different element on the page to get focus away from the one he had been working on. There were, as one could expect, some typos in the text, and the user repeated the process of typing invisible text twice before he was satisfied.

The user next tried to copy this slide from the old presentation onto the new one, but he lost the background picture and text formatting since it was based on the default template. Therefore, he continued to work on the old one. From the File-menu he chose *Save as* and overwrote the new presentation. He then selected all the slides in the renamed old presentation in the outline view, except the title page and deleted these. Then he changed the date on the title page.

He went back to the email, which contained the agenda. It was supposed to go into the presentation on the slide after the title page. From the right-mouse-button menu, he picked *New slide*. It came up based on the layout from the title page, which was not what he wanted. He decided to try writing it again on top of the right layout, which he had found in the old presentation. He opened the original again and copied the text that he had in the old document over to the new working version. He closed the working document and saved the old one. The new was re-opened and he turned to slide number two. It now had the right layout, with a title line and a larger text box, implying an itemized list. He changed the title and then copied the new agenda from the email and pasted it on top of the old text, which was now in the new document version.

The text formatting properties from the email, however, were also pasted with the text. It now had a font, size and color, which deviated from the front-page. The user noticed and immediately amended the text by the following steps:

1. Selecting the desired font size from the toolbar
2. Selecting the desired font color from the toolbar

3. Selecting the font type from the Format → Font menu, since it was not available from the toolbar

He then selected the text, which was still not laid out correctly and chose *Itemize* from the right-mouse-button menu. It was now a list of equally formatted bullet points. The user started editing the text and wanted to move line 2 so that it was aligned with the second column on the first line, but without any bullet points.

Moving the text marker to line two and hitting <tab> resulted in a slide that was correctly aligned, but had the unwanted bullet point. It had also changed to *italics* and the font size was smaller. The user therefore selected the line using the mouse, chose Format → Font from the menu and changed the size and removed the italics. The text now had formatting like line 1, but it was still not aligned where it should be. The user thus deleted blanks using <backspace> until the bullet point disappeared and then inserted blanks again using the spacebar until the text was back where it belonged.

The remainder of the items on the agenda was supposed to have a different formatting. The user hit <enter> to make a new line. PowerPoint did not insert a bullet point. He therefore clicked the *itemize* icon on the toolbar and then typed in the rest of the agenda without further adjustments; the items appeared automatically just as expected.

Deciding that he wanted the entire text of the document shifted one more tab length to the right, the user selected all the text and hit <tab>. This moved all the text, like he wanted, but it changed at the same time the font size, type and bullet-type. This was because PowerPoint changes the style of the bullet points with usage of <tab> like this.

This was definitely not what he wanted and he chose *undo* from the toolbar. He then performed exactly the same sequence of actions again, but to no avail: The result was exactly the same, even after the third attempt. The user "gave up" and left the text like this, selected all of it and changed the font type and removed the italics from the Format → Font menu. He then clicked the *itemize* icon on the toolbar to get the right bullet points back, but it removed all of them instead. He clicked it again, and that brought the bullets back. From the Format → Bullets and numbering menu, he set the bullet points back to the style that he originally had.

The agenda was now finished. The entire process had taken 1 hour and 23 minutes in order to finalize the two pages (front page and agenda). The user was then supposed to make a page with an overview of the talks. It needed to comprise the following items: Name of the talk, name of the lecturer and the abstract. He marked the text on the slide with the agenda to select it, copied it and pasted it

into a new document using the right-button menu. He changed the heading and continued writing white text on white background, as he did it the first time. This time he had to do twice to get it right.

Afterwards, the abstracts describing the talks were copied from the homepage of the seminar and pasted into the presentation. The formatting of paragraphs looked all right, but the text was not boldface as desired. The user selected the text and clicked the boldface-icon from the toolbar. The next sub item was selected, and he picked *italics* from the toolbar. He then used the <tab> to move the entire sub item-paragraph one tabulator length inward. The paragraph was then divided into individual lines using <enter> and each line tabbed into place one by one. Now the result was as desired: Boldface titles with one tab aligned italics abstract underneath.

Having looked through the document, slide by slide, the user also chose to watch the whole series as a slide show. He did not find any errors. He then printed it using the File → Print menu, to double check against the content of the emails and the webpage.

DISCUSSION

We have described how users systematically and methodologically relate to the intelligence of the user interface, but perhaps not in the ways intended by the designer. Users fall back on their own practical intelligence, when the functionality offered by the system does not match *their* intentions. This happens often, since the intelligence of the systems is pre-planned and static, whilst users form highly individual tasks. These tasks are, however, recurring and stable, and easily broken down into discrete steps, which the application *can* take into account. One consequence of the interface that the users routinely dealt with in this way was the formatting problems caused by the “smart” copy-paste operations of MS Office. Finally, the interruptions and shift of work orientation that such problems induce, seems to be a real candidate for a metric that can be used to detect *when* the system should start acting as an intelligent agent for the user.

The ethnographic studies presented in this paper show that users fall back on well-known and transparent routines, rather than efficient problem-solving strategies, when they encounter problems related to intelligent user interfaces. For MS Office, many such problems are associated with the “smart copy-paste”- functionality, which (for Word) copies paragraph formatting across. It can (for PowerPoint) copy either a slide or the text in a slide, but master slides do not change even if the local formatting changes. Also, the different approaches to changing the formatting of text, either by amending the visual layout for paragraphs individually or re-specifying

the style, which they are an instance of, confuse users. The generic approach of working with styles (“classes”) instead of paragraphs (“objects”) represent a different way of thinking, and shows clearly that it is the “intelligence” of designers (who program this way) or typographers (who are professionally trained to think about the layout in general terms, rather than form-as-content for specific paragraphs) that is represented in the systems. The document metaphor breaks down since the designers’ ambition is actually to make the tool act as an editing agent rather than instantiate electronically a piece of paper, which is what it “looks like” to the users.

This study shows that no amount of “designer intelligence” can be expected to work well in different use context. It is absolutely necessary for intelligence in interaction design to be based on the action of individual users. Sometimes then, an intelligent user interface strategy can be really simple, such as giving users easy access to their local history, so that they can do again what they just did, with less effort. Intelligence is not always complicated, and this is something that we need to take into account when we wish to design and deploy a more ambitious interaction design.

REFERENCES

- [1] A. Jameson, "Adaptive interfaces and agents," in *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*: Lawrence Erlbaum Associates, Inc., 2003, pp. 305-330.
- [2] M. Maybury, "Intelligent user interfaces: an introduction," in *Proceedings of the 4th international conference on Intelligent user interfaces* Los Angeles, California, United States: ACM Press, 1999, pp. 3-4.
- [3] M. T. Maybury and W. Wahlster, "Readings in intelligent user interfaces," Morgan Kaufmann Publishers Inc., 1998, p. 648.
- [4] C. A. H. Miller, "The Rotorcraft Pilot's Associate: design and evaluation of an intelligent user interface for cockpit information management," vol. 12, p. 443, 1999.
- [5] J. R. Miller, J. W. Sullivan, and S. W. Tyler, "Introduction," in *Intelligent user interfaces*, J. R. Miller, J. W. Sullivan, and S. W. Tyler, Eds. New York: ACM Press, 1991, pp. 1-10.
- [6] D. A. Norman, *The Invisible Computer: Why Good Products Can Fail, the Personal Computer Is So Complex, and Information Appliances Are the Solution*. Boston, Mass., USA: The MIT Press, 1998.
- [7] D. Randall, R. Harper, and M. Rouncefield, *Fieldwork for Design Theory and Practice*: Springer, 2007.