Visualising the hidden

Lecture Notes 3 for INF4280 – development of IT competence in organisations.

Jens Kaasbøll, Department of Informatics, 2009

Computer users who understand concepts and principles learn new applications more easily. When using an application, some features and principles are easily recognisable at the interface, for example that the cells in a spread sheet are organised in a grid, and that the text in a document has a specific layout. The sequence of operations, typically whether to choose data before operation or vice versa, may not be displayed, but they are experienced through the users' actions, so we obtain an immediate impression.

Other features are less prominent. Examples of hidden features are that the caption of an illustration in a text document does not belong to the main text flow, and that behind a number in a spreadsheet cell could be a formula which refers to many other cells. In the latter case, one can click at the cell and see the code of the formula, but there is no graphical representation of the functional dependency between the cells. In the word processor, there is no intuitive way to see where one text flow starts and another one ends. It might be possible to view the non-printing characters, but these do not tell us about the text flows or many other properties of the document, like the paragraph and character styles.

When there are no ways to show the hidden features, they remain abstract until made explicit. The written text is a one-dimensional sequence, while structures in the computer often are also of other kinds. Since many hidden aspects are structural, a combination of language and graphics would normally be a better option than just one of them.

Creating useful graphical models is partly arts & crafts, but there are also principles to consider. The books by Edward Tufte constitute a comprehensive introduction to the area (Tufte, 1990; Tufte, 2009). Marti Hearst (2003) has made a tutorial on graphical elements and how people experience them, while Rosling (2006) provides a video of visualisation of numbers and statistics. In the following, specific considerations for visualising the interior functioning and structure of software is presented.

Recognisable elements

Any explanation of what goes on in the interior of the computer should be based on the current competence of the users, including the users' understanding of concepts, experience with operating the software and their background for understanding the notation used.

In order to aid understanding, and not making it more difficult, graphical representations need to be

- simple, in the sense that they contain a small (7±2) elements
- recognisable, so that each element provides immediate meaning

The model of the file system in Illustration 1 is simple, but is made with a notation which is not recognisable by most users. On the other hand, maximum recognisability is sought in Illustration 2, while this illustration does aim at providing a general presentation of the structure of the file system. Illustration 2 also uses examples instead of the general categories in Illustration 1, bringing it closer to user experience but making the illustration larger and less simple. There is often a trade-off between simplicity and recognisability.

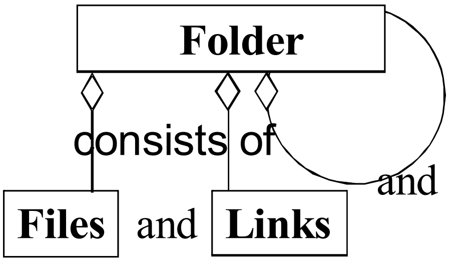
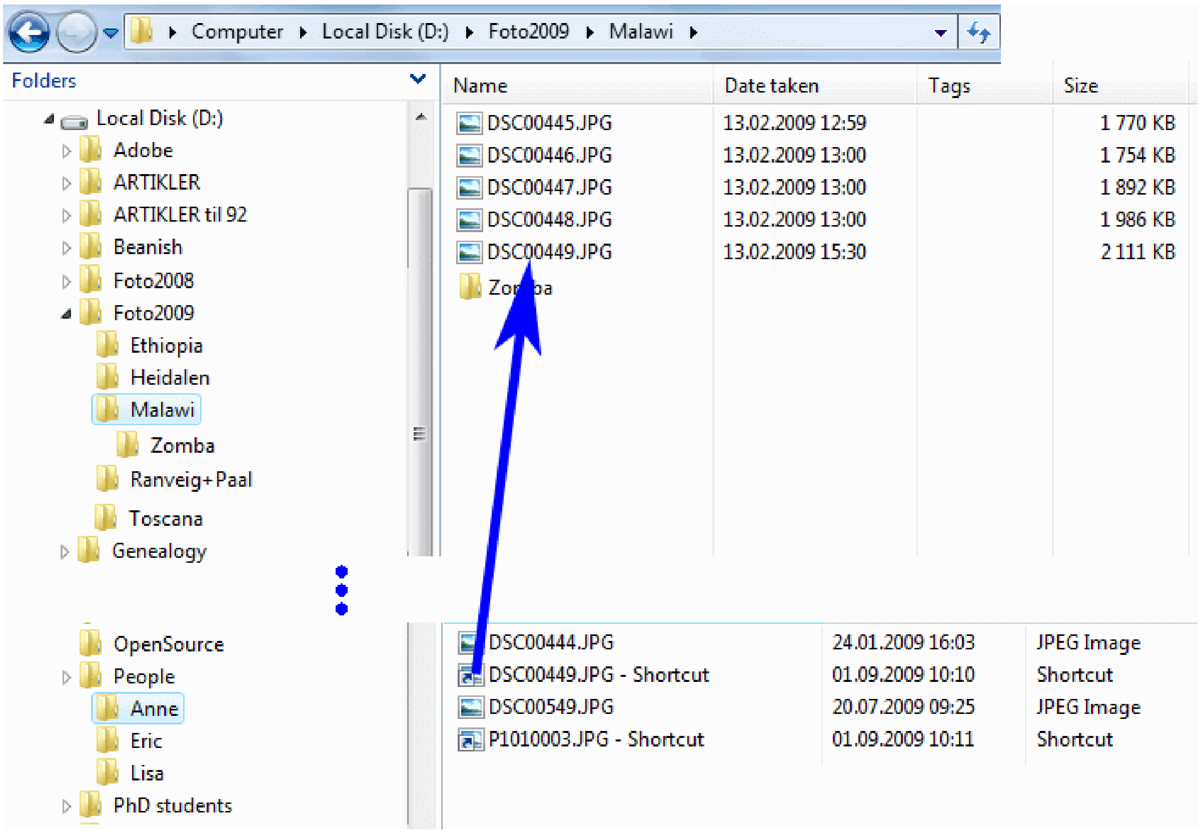


Illustration 1: Abstract model of the file system.



Name	Date taken	Tags	Size
DSC00445.JPG	13.02.2009 12:59		1 770 KB
DSC00446.JPG	13.02.2009 13:00		1 754 KB
DSC00447.JPG	13.02.2009 13:00		1 892 KB
DSC00448.JPG	13.02.2009 13:00		1 986 KB
DSC00449.JPG	13.02.2009 15:30		2 111 KB
Zomba			

DSC00444.JPG	24.01.2009 16:03	JPEG Image
DSC00449.JPG - Shortcut	01.09.2009 10:10	Shortcut
DSC00549.JPG	20.07.2009 09:25	JPEG Image
P1010003.JPG - Shortcut	01.09.2009 10:11	Shortcut

Illustration 2: Recognisable model of the file system.

Illustration 1 is a model of how the file system can be conceived under the surface, while Illustration 2 is mainly a surface model with an additional graphical element for showing the under-the-surface connection. The user interface of the Windows file system provides a reasonably good view of the data structure when viewed in the Explore mode shown here.

Visible impression versus under-the-surface structures

For applications which aims at What You See Is What You Get (WYSIWYG) principle, the data shown at the interface is supposed to mimic the printed copy, and then there is little room for also showing the underlying structures at the interface. A word processor file can include a spread sheet table, and such imports of other data types into a text file can often be done either by making a copy of the spreadsheet or by setting up a reference to the spreadsheet file. The visual result in the text file will be the same, but the underlying data structure will differ. In such situations, the illustration should depict the two data structures in the hard disk of the computer, and that they appear in the same way at the interface, see Illustration 3 and 4.

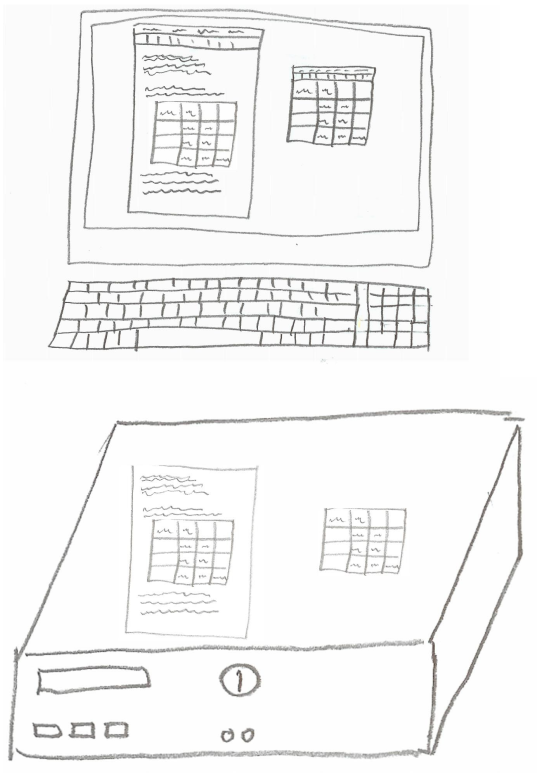


Illustration 4: Imporing a spread sheet into a text file by making a copy.

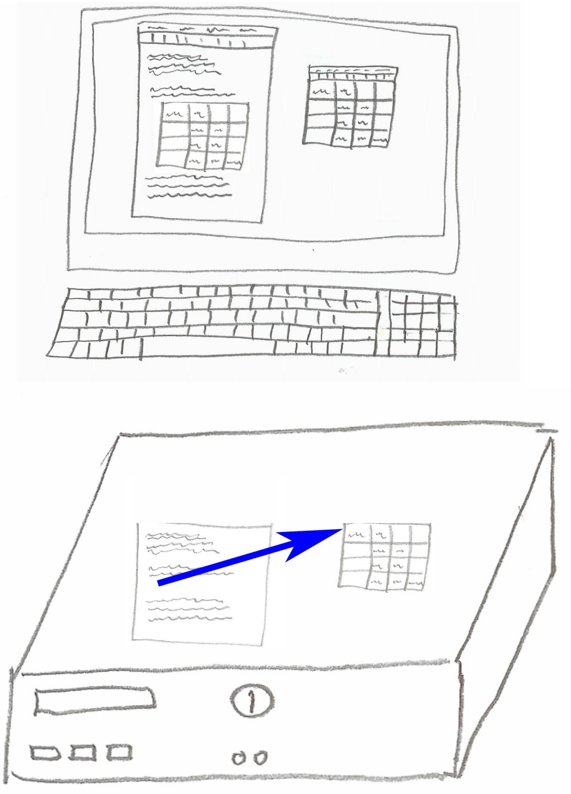


Illustration 3: Importing a spread sheet into a text file by setting up a reference.

The elements of Illustration 3 and 4 exploits the everyday experience of the layered architecture of the computer. The hardware, the data within the computer, and the visible copy of these data in way which the screen allows. In order to strengthen the impression of the data inside the computer as something which is different from what is seen on the screen, this data is visualised with a lighter shade of gray.

Several relationship types

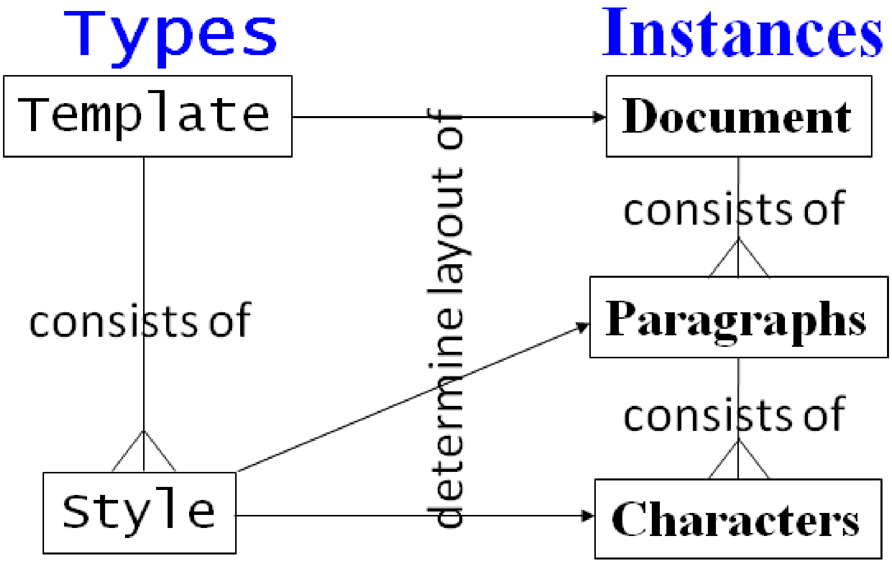
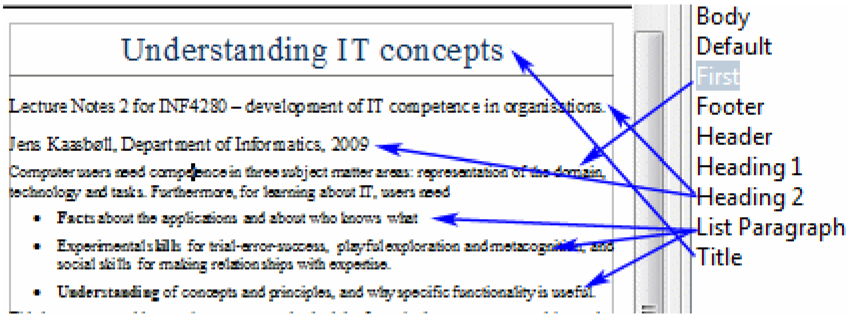


Illustration 5: Two types of relationships in one model.

In the previous illustrations, blue arrows are used for denoting reference, link, pointer, shortcut, or whatever particular name is used for the mechanism for achieving functional dependency. Often, there is more than one type of entity and relationship to illustrate, and then the corresponding symbols have to be differentiated. Illustration 5 shows how the character fonts are used for separating the two kinds of entities in the model. Also, the two kinds of relationships are differentiated, and since the arrow and the crowfoot could mean anything, the relationships are also named. Even with these names, the model is at a high abstraction level which might be better suited for teachers of computing than for learners.

Exploiting the more easily recognisable interface and adding the type-instance relations at an example may produce a visualisation like Illustration 6. Adding the consists-of relationships here would have made the illustration far from simple.



Understanding IT concepts

Lecture Notes 2 for INF4280 – development of IT competence in organisations.

Jens Kaasbøll, Department of Informatics, 2009

Computer users need competence in three subject matter areas: representation of the domain, technology and tasks. Furthermore, for learning about IT, users need

- Facts about the applications and about who knows what
- Experimental skills for trial-error-success, playful exploration and metacognition, and social skills for making relationships with expertise.
- Understanding of concepts and principles, and why specific functionality is useful.

Illustration 6: Style-paragraph relations.

Instead, one more illustration would be needed to capture also other aspects of the model in Illustration 5 if interface copies are used.

Dynamics

A learning process for concepts and principles divided the stages of competence into

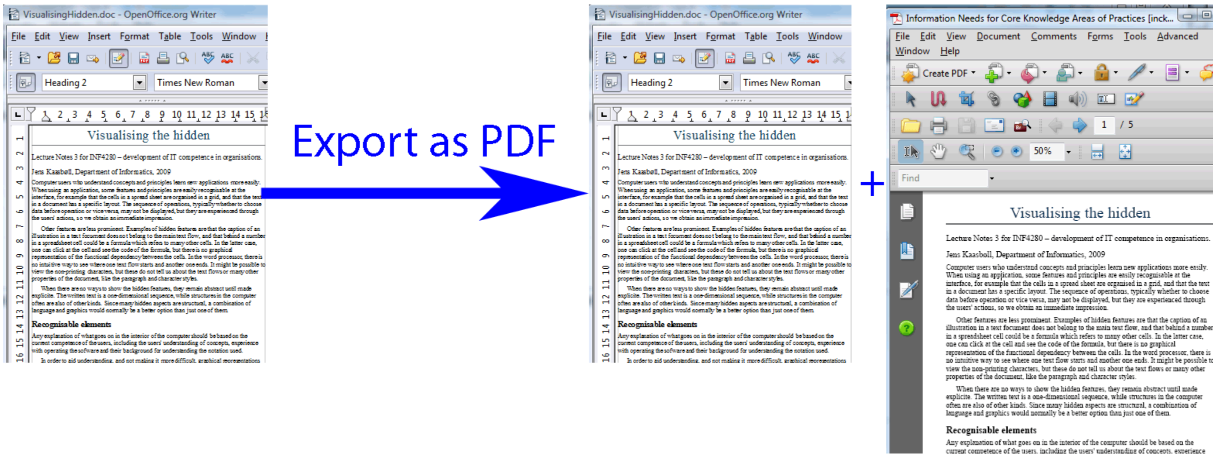
Action, where the user carried out opera

Process, where users tell about the effect of the action, that is transforming input to output.

Object, where the users talk about the principles or concept as an object of its own, and relate it to other topics.

The action stage would entail carrying out a sequence of operations, and having documentation on which buttons to push will ease this task. However, this will be a description of the visible clues at the interface, so it does not pose a challenge for visualising hidden data or structures.

In the process stage, however, there is a need to understand the status before and after, and this could require some models of the hidden. Illustration 2, 3, 4 and 6 all depict status before or after an operation. For example, users may be unsure of the result of Save As or Export operations, by not knowing whether the original file also will remain. Illustration 7 Shows an example of how to depict the pre and post status of an operation.


VisualisingHidden.doc - OpenOffice.org Writer
File Edit View Insert Format Table Tools Window
Heading 2
Times New Roman
Export as PDF
+
Information Needs for Core Knowledge Areas of Practices [inck...
File Edit View Document Comments Forms Tools Advanced Window Help
Create PDF
1 / 5
50%
Find

Visualising the hidden

Lecture Notes 3 for INF4280 – development of IT competence in organisations.

Jens Kaasbøll, Department of Informatics, 2009

Computer users who understand concepts and principles learn new applications more easily. When using an application, some features and principles are easily recognisable at the interface, for example that the cells in a spread sheet are organised in a grid, and that the text in a document has a specific layout. The sequence of operations, typically whether to choose data before operation or vice versa, may not be displayed, but they are experienced through the users' actions, so we obtain an immediate impression.

Other features are less prominent. Examples of hidden features are that the caption of an illustration in a text focument does not belong to the main text flow, and that behind a number in a spreadsheet cell could be a formula which refers to many other cells. In the latter case, one can click at the cell and see the code of the formula, but there is no graphical representation of the functional dependency between the cells. In the word processor, there is no intuitive way to see where one text flow starts and another one ends. It might be possible to view the non-printing characters, but these do not tell us about the text flows or many other properties of the document, like the paragraph and character styles.

When there are no ways to show the hidden features, they remain abstract until made explicite. The written text is a one-dimensional sequence, while structures in the computer often are also of other kinds. Since many hidden aspects are structural, a combination of language and graphics would normally be a better option than just one of them.

Recognisable elements

Any explanation of what goes on in the interior of the computer should be based on the current competence of the users, including the users' understanding of concepts, experience

Illustration 7: The input and output of the Export operation.

References

Hearst, M. (2003) *Information Visualization: Principles, Promise, and Pragmatics*. CHI 2003 tutorial. Downloaded from http://bailando.sims.berkeley.edu/talks/chi03-tutorial.ppt 2 Sep 2009

Rosling, H. (2006) *Hans Rosling shows the best stats you've ever seen.* Downloaded from http://www.ted.com/talks/hans_rosling_shows_the_best_stats_you_ve_ever_seen.html

Tufte, E. (1990) *Envisioning information*. Cheshire, Conn.: Graphics Press.

Tufte (2009) *The work of Edward Tufte and Graphics Press.* Downloaded from http://www.edwardtufte.com/tufte/index 2 Sep 2009