

The Design Sketching Process

Rikke Orngreen

Copenhagen Business School, Denmark, orngreen@cbs.dk

Abstract. Sketching the *what-if* situation provided insight into alternative design possibilities, and gave an idea of how the sketched design will work in the context at hand. Though software provided possibilities of creating a high fidelity executable prototype at a very fast pace, here sketches that were non-executable drawings of the envisioned system, are investigated. The sketches were used in various forms within the development team as well as in collaboration with external experts and users. Where issues of usability and technical requirements are often dealt with separately, this approach suggest simulation through drawings of the context of use and the different users' needs. This turned the design sketching period into both an analysis and interaction design of the new envisioned work processes, as well as gave input to the future production process. The case study illustrating this design sketching process from rough drawings of conceptualisations and detailed storyboards of functionalities was the development of an e-learning platform for case-based learning.

1 Introduction

Sketching provided artefacts that may be used for collaborative exploration of the system being developed, by reading and interpreting the drawings and concurrently (re)write and (re)draw new ideas to the design. As such it created mutual reference points, similarly to mapping models made by users and developers like the rich picture in Checklands soft system methodology [1] and Avison & Wood-Harpers Multiview [2]. In this paper it is illustrated how the process of sketching thus became similar to work analysis, by observing and discussing the sketch and interpreting it, but it moved beyond the present situation, and took the future system's work context into consideration. The challenge was to allow for the design sketch and the design idea it represent to be adjusted easily or even more so, disposed of easily. This process allow for alternatives to be found, as opposed to refining existing ideas, but only if the team makes it an established way of working,

Please use the following format when citing this chapter:

Orngreen, R., 2006, in IFIP International Federation for Information Processing, Volume 221, Human Work Interaction Design: Designing for Human Work, ed. T. Clemmensen, Campos, P., Orngreen, R., Petjersen, Al., and Wong, W., (Boston: Springer), pp. 185–202.

As design sketches were made, worked on, and then either eliminated or chosen, the design of the system was shaped. The transformation from representations of concepts and ideas to representation of functionalities and contents began. This meant a shift from hand drawings of alternative ideas to digital storyboard-like representations of functional requirements, as is discussed in this paper. The creation of such detailed sketches is always a complex and resource demanding task (time, adequate competences and insight into the system are needed). The detailed sketches provide vital visual documentation that can be accessed and refined, and used in the production process. Here, the critical process of trying to anticipate the system being designed, envisioning the future work context, was continued. The difficult part was to recognise when the sketch was a design suggestion (a drawing of an idea) and when it was a visualised specification (a detailed storyboard) similar to a written requirement specification or design documentation.

The case study in this paper was the development of a case-based e-learning platform named CaseMaker. The case-based learning method and pedagogy is often practiced at university level courses. Through the development of CaseMaker it should be possible to develop digital online cases, use them in online and blended teaching situations, and for students to analyse the cases online. As an example, cases in business schools often consist of company descriptions and material (as interviews, balance sheets, project or product presentations etc). Teachers use cases in courses to e.g. show how the different methods or theories work in practice. Students thus discuss the cases using these theories as well as experience from their own work situations, similar companies etc. Like many e-learning platforms, CaseMaker needs to support many roles (that of developers, teachers and students as primary user-roles), and thus also many ways of working with the system.³

Design sketching provided the CaseMaker team with design insights about how to work with possibilities from present case-practice. Sketching was also used when trying to get inspiration from other sources and design ideas. This eliminated some of the “childhood” problems that many e-learning platforms were facing. The sketches were used in various forms within the development team as well as in collaboration with external experts and users. Other tools and techniques were also used within the project and in relation to the design, but are not described or illustrated here. E.g. scenarios were used in the very early stages of the project, use-cases and other flow-models in the later design phases, and so forth. The focus of this case study is thus not to describe the big picture of the CaseMaker development process, or to present case-based development, teaching and learning, but rather to

³ CaseMaker was funded by the e-learning fund at the Copenhagen Business School (CBS). It was and is still being developed at CBS with Rikke Orngreen as project leader and active member of the development team. Primary development team members came from the CBS Learning Lab and included a developer and 1-2 senior consultants who were present at almost every design meeting. During development users and experts were invited to panels, discussing the design, and user workshops were held. Third party consultants were hired, and external programmers worked on the actual production together with an employee from CBS Learning Lab, particularly in the first half year of 2005. During summer 2005 the first version of CaseMaker were finished, but not yet implemented for public use. For more information about the motivation and functionality of CaseMaker, see [3] and [4].

point out specific perspectives on the various forms design sketches may have, and the variety of processes involved.

This paper starts out by illustrating the various forms sketching may take, and the different mindsets, implicit to the way they are used. This presentation (in paragraph 2) is based on very different types of literature, from the semi-abstract to the extremely concrete, and are all discussed with a very practical and specific focus. Afterwards the CaseMaker project is used (in paragraph 3) to illustrate how the design sketching process can use not only one type of sketch and sketching process, but rather an array of sketches that vary through the phases of development.

2 Sketches and their relation to.....

To define a sketch may not seem that hard. The Wikipedia defines (for the time being) a sketch as: “*A sketch is a freehand drawing or other composition that is not intended as a finished work. Sketches usually serve to store ideas for later use...*”⁴ There might be slight variations, and certainly the concept of use is broad and varied. It is the overlap with other techniques and the processes involved during sketching that seem to be worth while dwelling on for a moment.

A sketch, a storyboard and a prototype are artefacts that can all be used in relation to the design representation of a software product under development. These three forms of representations have a lot of overlap, but also signify the various stages of the software product – from representations of an idea or concept in a sketch, to functionalities and contents in a storyboard, to detailed design in a possibly executable prototype. They are concepts stemming from different traditions (the creative design and art traditions, the film and multimedia world, and engineering), but can as just described be viewed as part of an incremental and iterative software development process. But just to make the confusion complete; prototyping as a complete software development method (as Rapid Prototyping opposed to the Waterfall Model), both within software engineering, human interaction and software management, almost always contains a sketching and/or a storyboarding period. I.e. sketching then becomes an activity in a prototyping process. Prototyping is often seen as a method that bridges the gap between the various phases of analysis, design and production [5].

Thomas Snitker, a Danish usability practitioner, writes briefly in his newest edition of a book on user-centred design, about advantages and disadvantages of paper prototypes under the heading: “*this napkin is a prototype of my new website.*” [6, page 75]. Now, according to the description of sketches so far, this may lead to the conclusion, that if the napkin contains hand drawn images, it is not a prototype, but a sketch. Both interpretations might be correct, and thus states more about the methodology behind the things drawn than the drawing it self.

⁴ The wording: for the time being, is used here to indicate that information found on Wikipedia is not permanent, and is not meant to be, but is rather dynamic in its nature – accessed the November 2005, http://en.wikipedia.org/wiki/Sketch_%28drawing%29

2.1 A design perspective on a sketch

Prototyping, when seen as a life cycle development method that the development team uses from the very first initialization (to begin the development and work on a new idea, a problem or opportunity, or a re-design task) to the test and evaluation of the finished product, might include a sketching period. But to make a sketch is mentally different from making a prototype. The difference was amply illustrated by William Buxton in his keynote at the Interact 2005 conference. Here he presented the difference in paradigmatic views of a sketch and a prototype, which he summarised in a table longer, but similar to table 1.

Table 1: Buxton's table on differences between a sketch and a storyboard, highlights by author⁵

Sketch	Prototype
Invite	Attend
Suggest	Describe
Explore	Retire
Questions	Answer
Propose	Test
Destructive	Constructive

That a prototype is constructive in nature, and thus keeps on building on the same ideas and concepts, may not be surprising, but that a sketch is destructive is a somewhat blunt statement (see table 1). The perspective is that many ideas and thoughts may seem good until worked on for a while, and when prototyping the developers may not reach the difficult point of being able to “kill ones darlings”. Sketching makes this easier. It is easier to draw something quick on paper, and then realise 30 minutes later the idea was not something meant to be, than using half a day or a week on a detailed storyboard or executable prototype, and then throw it away. The more the idea has been worked on, the more difficult it is to dismiss.

The interesting notion of a sketch as inviting further ideas and suggestions, as concept and thoughts, and not as a product to be further developed, is a fruitful contribution to the design sketch discussion. It first of all fosters the idea that many sketches can and should be made, with some, the majority, being immediately discarded. At the same time, as sketches flow from the developers, the mere process of sketching always seem to open up, rather than close down. Secondly, before a sketch is able provide input to the design process, it also has to be looked at, it has to be read, understood (even misunderstood) and interpreted, before it leads to exploration of alternatives.

At the Interact conference Buxton spoke highly of hand-drawn sketches on paper, but also recognised working processes emerging, which would lead to hand-drawn sketches that are digital. Here he made a point out of presenting sketches that were made by graphical designers and industrial designers that know how to make

⁵ Based on notes from the keynote by William Buxton: “*Sketching and Experience Design*”, given at Interact 2005, the 14th of September 2005, Rome, Italy. See also Buxton's homepage for more references: <http://www.billbuxton.com>

beautiful hand-drawn digital images, but also showed how they were “dressed-down” to look like sketches, and to be interpreted as draft suggestions and ideas.

2.2 A storyboard – from movie to multimedia production⁶

Faulkner determines that: "*The initial design for the system can most conveniently be presented in the form of a storyboard*" [8, page 103]. Storyboarding is a method originating in movie production. A storyboard phase typically begins, when a decision to produce a movie is taken. It contains two elements: drawings of the appropriate appearance of the screen, and comments about everything from the set, the actor(s) mood, what should be said, foreground and background action etc. [9,10]

Storyboarding is a method, which is useful in designs that use multimedia, and which have many forms of interaction, because of its ability to visualise, what should take place in the system. Multimedia storyboards typically also contain information about timing requirements, hyperlinks, animations, text etc. [11,12]. A storyboard can serve as a tool, which supports the design process in conjunction with scenarios [13,14]. It can also serve as a way of documenting the decisions taken, and then later be used when the production of scripts, media production and authoring (programming) begin [7,15,16]. Figure 1 shows an example from Hofstetter's book on multimedia literacy, and even though this book only gives an extremely short introduction to storyboards, Hofstetter is one of the few who show an example of a format that can be used for documentation [15].

In Orngreen and Pries-Heje 1999 the storyboards from various projects are also depicted and discussed. A resulting framework for contemplating the degree to which storyboards are continued to be used throughout the development process is given. The idea is not to promote the use of a one and only storyboard standard, but rather to contemplate the need for storyboards based on 4 variables of the system under development: complexity, size, amount of re-use of material, and media usage. The need to make detailed storyboards for every part of the system grows with: high complexity, large systems, large amount of material (contents, navigation etc.) being made from scratch, and high degree of media usage. Whereas simple small systems, with a lot of reuse and little media usage just need conceptualisation of the general idea of the system [12]. This is to ensure both an economically sound approach, and a user-centred approach that matches the envisioned work context.

⁶ Large parts of this paragraph draws on previous analysis and work on storyboards found in Rikke Orngreen's phd. dissertation on the development of multimedia teaching cases [10]

Module: _____ Strand: _____

Filename: _____

Screen No. _____ of _____

Images: _____

Audio: _____

Video: _____

NAVIGATION

Next: _____

Back: _____

Menu: _____

Help: _____

Notes: _____

Figure 1: Storyboards, design and documentation tool. Source: [15, Figure 45-6, p. 379]

2.3 Sketching in user- centred and participatory design

Sketching is also used to understand present practice with a strict analytical focus, rather than the more analytical creative / design focus of the two previous paragraphs. Visual mapping of the present context, rich pictures, geographical models etc. are examples of these [1,2,23]. They are either drawn by the developers during or after field studies or the users themselves in a more participatory oriented development process. E.g. the more detailed a storyboard becomes, the closer it resembles the prototype as a product, and the farther away it moves from the Buxton idea of a sketch. However, both a storyboard and a sketch in Buxton’s notion have common perspective in that the sketching of ideas and functions are primarily done as part of the developers work. In user-centred design the approach is to involve users and clients in the sketching work of present practice and the vision. [13,17,18,19]

Adaptations of storyboards are used to interpret the current or past situations. Movie analysts use them when interpreting scenes from movies, describing and showing areas that are not immediately in the frame of the camera, but vital to the story. (As something physical or a sense of atmosphere). In IT development, drawing flows of frames can be used with annotation strategies from dance notation and video analysis literature to note important contextual features, feelings, and dynamic behaviour (user actions) that the system under development should adhere to. [7].

The vision phase is one of 4 phases in the participatory framework for pre-investigations, named the MUST method by Boedker, Kensing and Simonsen. The techniques used in this phase of exploring the vision of the system to be developed are: virtual mapping, drawings, collage of sketches/ photo montage and scenarios. In the presentation of the MUST-method, the questions raised by the user-groups when applying these techniques are of special interest. Questions as: what if there is a new system which is able to...; what if, we organise ourselves differently etc. [17]. These are questions and answers that will be reflected in the sketches. According to Boedker, Kensing and Simonsen, a pre-investigation workshops that make hand-drawings, collage, design sketches and data models can use these tools and techniques for dialogue on abstract knowledge of present and new IT work practice. [17, figure 8.2 page 216 and 270-319]. Their starting points are very oriented towards “problems” in the present work practice and “solving problems” with IT support, and not so much towards opportunities and exploring, and experimenting with innovations. It is however, interesting to see, how this starting point joined with these tools, provide a sum of artefacts of various kinds of sketches that creates a solid interpretation of the present as well as envision of the new. This is a solid basis for the development team to work with after the pre-investigation.

Similarly development teams with roots in user-centred design and human computer interaction often invite users and clients to see various representations of the teams understanding of the context and the teams design suggestions. E.g. Buur and Boedker describes the design collaboratorium, where users are invited to workshops contributing with reflections on the teams design suggestions (often prototypes, but also collages of other users’ statements, sketches etc..) [19].

2.4 Sketching software tools

Newman, Lin, Hong & Landay identifies the design process as consisting of phases of: Discovery (where hand-written notes are primarily used); Exploration (where hand-written site maps and storyboards are used), Refinement (where digitised schematic and mock-ups are used); and Production (where digital prototypes are developed) [14, see for example figure 4, page 274]. In their wording site maps and storyboards are both visual representation of the system structure and flow / navigation, as opposed to also having storyboards that contain information about contents and functionalities, as discussed earlier.

With the objective to develop a tool, named DENIM (Design Environment for Navigation and Information) that assist web-designers, Newman et al carries out a thorough qualitative investigation of how 11 web-designers work [14]. They find that all of them use “*low-fidelity sketches*” on paper, which are important for the design process focusing on structure and navigation and are means of communication (between developers). However, they also find that designers are reluctant to show them their drawings, and that they would never show clients these rough sketches, only to co-designers, who provide valuable feedback early in the design process. If a sketch idea is presented to clients, digitalised “nice” version of the same idea is made (in Photoshop or Illustrator) [14]. Similar conclusions and objective of the two tools (FreeForm and SUMLOW) are discussed in Plimmer &

Grundy 2005. They look at how hand-drawn sketches made with stylus on tablet PC's can be automatically "*beautified*" into resembling real data models, database entry forms etc [20].

3. The CaseMaker design sketching process

Numerous examples of sketches from the CaseMaker project could be presented here. From visual maps of the existing context, over heaps of paper sketches of design suggestions, to many, many PowerPoint files of storyboards. Three examples have been picked out to show the stages of the design sketching process and to highlight some of the problems and opportunities the tools encountered, both on a specific level for this project and as general lessons.

3.1 A case of a technical function that initiated an overall design strategy

In the very first "whiteboard discussions" in the core development team (November 2003, just after receiving funding for the project) the team found that the experience with not only CBS' e-learning platform (which is Sitescape), but also experiences gained as invited teachers at other universities or corporate business universities, all pointed to the need for more flexible user / role assignments. In many platforms the administrator has to (semi)manually allow for users to enter the system, by drawing on information from the student and employee registration databases (known as LDAB). The registration database is updated, but not nearly fast enough, and even the guest-ldab gives problems. In this way the first whiteboard sketches served as a way of reflecting on present practice. These were very technically oriented problems, that are usually not dealt with at this stage, but which have severe influence on the users' interaction with current e-learning systems, and consequently the users' experience with and feelings for the e-learning system. It led to envisioning new ideas that were very different from present practice, and pointed towards more open strategies of allowing users to use the system if they wanted to, whether they were officially CBS students or not. This brought the context back into play, with questions like: "if we make it very flexible and user-controlled, what about security, and will the CBS Management approve of such an approach?"

The whiteboard flow-sketches of signing in took many forms. Figure 2 shows a simple tree-structure, which at this time of the discussion was enough to trigger a very long dialogue and many more sketches and other tree-structures. Together with issues about encouraging developers to share material and cases, the above issue eventually led to the decision to go for a very open standard system. It could later even be rolled out as an open source project. One might say that the discussion on the log-in procedure was the initiator to the design paradigm and mindset which was applied throughout the development period.

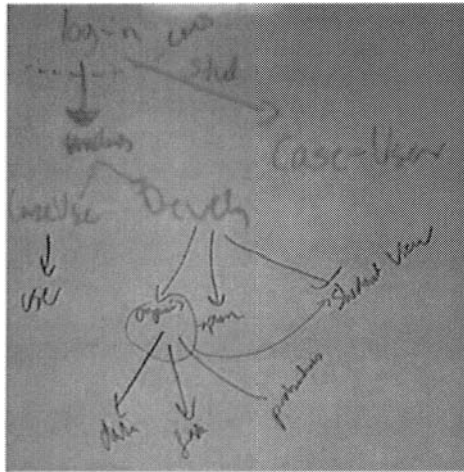


Figure 2: Whiteboard flow-sketch, log-on procedure led to open standard decision (from November 2003)

3.2 Trying to escape the prototyping cravings and software tools that came too early in the process

A year later the more detailed process of getting the structure and contents of the whole system in place began. At this point the contents and functionalities were clear, but the design of the structure, how to interact with the system, and which paths to follow had not been discussed yet. It is noteworthy that there were large breaks in the development process, and that the time difference of 1 year reflects these breaks. Probably approximately 2-3 months would have gone by, had the development team worked full-time. The objective now was to consider, how the structure should be, so that it supported a natural work process for the users. Though sketching was done in more details with other aspect of the system, taking point of departure in the log-on discussion from before gives basis for comparison.

A member of the team and an e-learning and developer consultant wanted to apply a software tool for sketching. At first the DENIM sketching tool mentioned earlier was used, but it sort of put the developers in a too rigorous and detailing state of mind, compared to the kind of switch between reflection and designing needed (see figure 3). When using DENIM (which is a newer edition of the first sketching tool by primarily Landay and Myers named SILK [21]) sketches often became low-fidelity prototypes with focus on structure. I.e. even though it visually were rough designs (figure 3), as developers we tended to treat these as design suggestion to be critiqued and commented on in a way that nourish further development, and not so much to rephrase the idea all together. As a consequence, DENIM was abandoned.

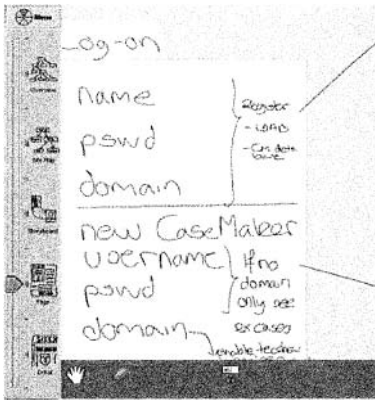


Figure 3: DENIM flow-sketch that was newer finished (from November 2004)

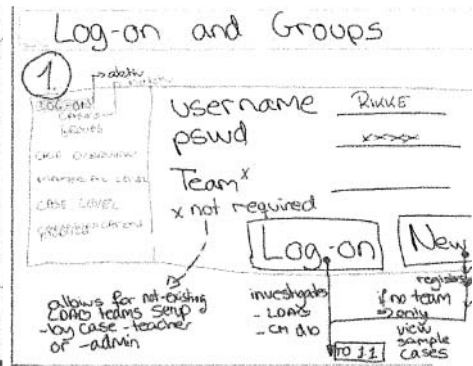


Figure 4: Journal note sketch that was transformed to a similar paper process (from November 2004 – cropped slightly)

Then, there was a brief attempt to still use the tablet PC as “paper” and the Microsoft windows journal software with a complete blank page as starting point (figure 4). However, again the idea of being fast at sketching an idea, read and interpret it, then quickly thereafter present the next idea, and then make the next etc.... seem to come second to the notion of getting more and more details into the picture, once the software comes into play. This is fine if the development team is aware of this “hook” to detail that the software gives. As with the DENIM attempt the Journal sessions only lasted two days with a couple of hours work each day, and were then quickly discarded as not usable at this stage in the process – at least for this kind of system. The development team members had had similar experiences on previous occasions.⁷ The experience is that the software is very appropriate for late detailed design and refinement stages of a design process, but not for contemplating the surrounding context of users and for getting into the “sketching-mode”.

3.3 Getting a user-centred and collaborative approach, by meeting and simulating context to eliminate and refine choices

The design sketching process is also about willingness to view design from the system or the user perspective. Where system developers in the software engineering tradition do have the users in mind, it is often the needs of the common or average use(r). In the rational unified process for example, use-cases are applied to show the users a way through the system. This creates a basis for defining system requirements [22]. These user sketches and descriptions often become self-fulfilling prophecies, as the user acts in ways that are intended, illustrating how useful the system will be.

⁷ The experience with DENIM in particular and using PC (tablets and ordinary) in general, stems from both the CaseMaker project during 2004 and 2005, but also from several student projects, where DENIM was used on real companies’ design and re-design tasks given to students.

In a user-centred approach, it is the various and individual contexts of use that are in focus, and the often contradicting ways users act [23]. Sketches and descriptions thus have to take point of departure in: Why and how can users use the system? Rather than only contemplating how the system can support the user? Rephrasing this to the CaseMaker design process: the software engineering approach (to put it a bit squarely) deals with: how can CaseMaker provide cases to case developers, teachers and students? And the user-centred approach deals with: how would different users work with CaseMaker (not only a developer, a teacher and a student, but various developers, teachers and students).

In the beginning the CaseMaker project team concentrated on making storyboards fulfilling needs that had been observed in case situations. E.g. the two sketches in figure 5 are analytical sketches or storyboards (see 2.3) of students working with traditional and multimedia cases. (From Orngreen's previous work on teaching cases [7]). The sketches are part of an analysis of a video recording. In addition, the sketches are supplemented with a larger written interpretation). I.e. first the sketches are made, and afterwards they are read, providing reflections that are documented in the written interpretations (the written interpretation is not shown here.⁸ The in- and out-of-frame concepts are seen on the sketch, where the square denotes the frame of the video screen and the drawings out-side of the frame documents what is not visible to the eye.

Such knowledge of existing case situations supplied input to discussions on e.g. types of cases that users would prefer to work with. Some case developers have material for cases that are appropriately presented to the students in a very linear fashion, others in a more networked fashioned, others again in a hierarchical tree structure. The teachers may want to give the student some material at the beginning of a case teaching event and additional material later on. Students would most probably like to access material at different times, but still be able to work collaboratively, leave notes to each other etc.

⁸ In some analysis the hand-drawn analysis was made on top of screen-dumps of the video recordings. However, in this situation the students wanted to be anonymous. Also, similar sketches are sometimes made during the observation not using video recordings, as video-observation and analysis is a very time consuming, and thus expensive process to carry out.

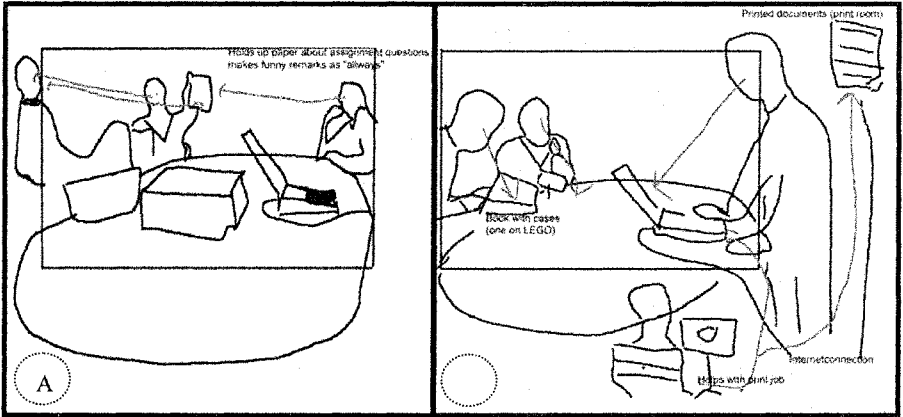
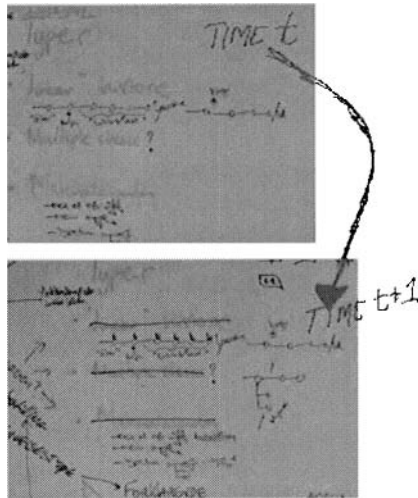


Figure 5: An analytical sketch of students working with cases, from [7, the studies were conducted during 1999-2002].

The start-up meeting in July 2003, mentioned earlier, was a two day workshop just for the development team (4 persons on this occasion). Here we also used the whiteboard-sketching sessions, to draw on each other's sketches. As seen on figure 6 there were several ideas and discussions about the types of cases at time t . As the discussion went on, some were eliminated at time $t+1$. The different words used to illustrate different ways through the case material were not easy to understand between the members, and more diagrammatic notations were then used. The diagrams led to an understanding that several types could be encompassed in the same notion of a case (se figure 6).

Figure 6: Whiteboard-sketch – picture from two subsequent periods (July 2003).



At some point we contacted experts and potential users (teachers and students). Panels were held (with presentation of ideas and storyboards and following discussion). Later vision workshops [17] and interviews about CaseMaker were also conducted. In these panels and workshops present practice, visions of CaseMaker and specific functionalities were either reinforced or dismissed. E.g. the idea of letting CaseMaker have cases that were sort of simulated story lines with multiple choice branching (i.e. the story would go in different directions depending on the choices) were eliminated. This was done because though simulation-based learning is interesting, it was not seen as part of case-based learning, but quite different from it. It would require a complete different mind-set compared to the other features included in CaseMaker, and might therefore seem more distracting and off-key rather than part of a coherent platform.

As a result storyboards as seen in figure 7 were made to provide estimates of the necessary time and resources needed for the programming phase, as a means for considering choice of coding platform and structure, as well as making the more detailed user evaluations (the user workshops as opposed to the user and experts panels). The example is still the case types, but now instead of referring to case types, the storyboard refer to case structures, and allow the user – the case developer – to structure the case material into headings at the same time.

In these storyboards (figure 7) CaseMaker was always shown as having a pane-menu, though the format had never been discussed. Just prior to the beginning of the programming process, it turned out that the members in the development group had different thoughts on why the pane-menu was used. Some thought the format was chosen to show overview of where we were in the system when dealing with a certain function. That it was not at all part of the design chosen and this was still to be determined. Others saw the storyboard as representing the final interaction design with respect to navigation. (Not with regard to colours, font etc., which were deliberately made ugly and dummy-like to show the state of the system, i.e. that there was room for re-design).

The team decided to investigate appropriate navigation options, by sketching and simulating use on a more detailed level. The team would imagine different paths through the system depending on different use objectives and preferences (see figure 8). Also, small pieces of papers with various input and contents to cases were made as to simulate different work processes, similar to the methods described in the many articles on evaluating paper mock-ups. (Including methods dating back to the often quoted “design at work” [24] and newer text books on interaction design [13] etc.)

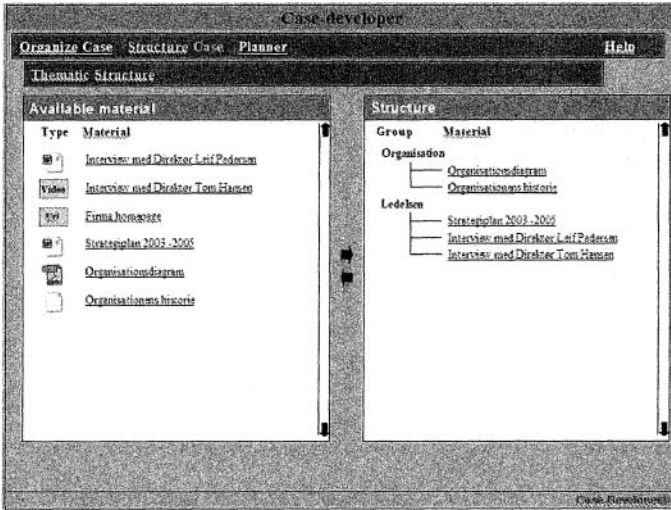


Figure 7: PowerPoint storyboard, case types in the case-develop module, from February 2004

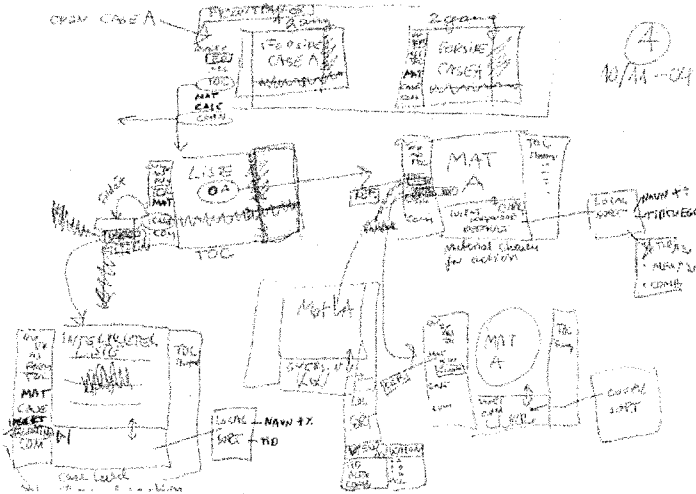


Figure 8: Paper sketches, trying out various work processes, November 2004

Up till then our system in the storyboards resembled a stand-alone system, and not an open flexible web-platform. Though the functional possibilities may be the same, the mere “look and feel” of a system, of the user’s perception of it, also have great influence. Furthermore, the pane-menu made an overview of the features in the system difficult, and created a laborious work process. The shift in interaction form only resulted in some extra hours of dialogue and evaluation with users. It would have been difficult to change, had the system been implemented that way.

Figure 9 shows the appearance of the first version of CaseMaker. The example chosen are the case-developer module and the case type's possibilities, now referred to as different case structures. The figure shows that the pane-menu was abandoned (figure 9).

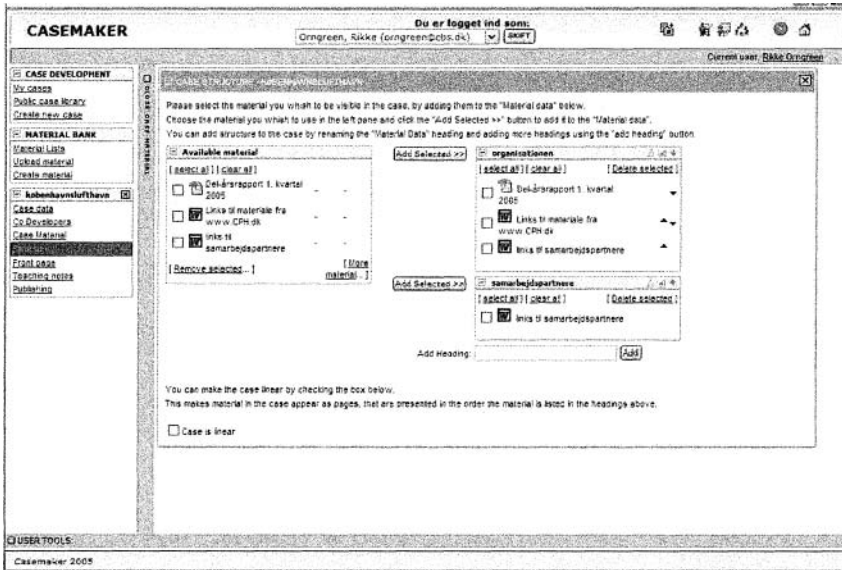


Figure 9: the first version of CaseMaker from summer 2005

5 Learning from the design sketching processes

Sketching is often seen as a very individual creative process of an artist and designer making a piece of art, a movie or a new product. The sketch becomes a medium for an internal dialog of suggesting design, and getting to new design suggestions by investigating their application (reading and interpreting the sketch), and by using the sketch to reflect on advantages and disadvantages. This process is also fully convertible to a collaborative process. However, the individual creative process must not be underestimated, as the collaborative design sketching process can easily lead to overruling of ideas that might have been great (but could be too quickly eliminated by other members of the team). As in the CaseMaker project marked by long periods of only periodical work on the project, sketching is a quick way of getting into the system “idea” again or introducing new people to the concept, by showing old sketches, and drawing new with them as you speak.

Getting people to sketch is not always easy, and it can be difficult to escape the “prototyping mode” of wanting to construct rather than being open and inviting at the same time as being destructive (table 1). In the project team many were used to a process of getting ideas, programming them, and then refine, rather than

simultaneously looking at possible alternatives. It is a mental shift to adhere to the design sketching process and this is not an easy shift to make.

Both the CaseMaker case study and literature show it can be difficult to “kill ones darlings”. The perspective here is that recognizing what the system is *not* about, is also a step forward, particular if this finding can be made early in the design process.

The CaseMaker design team also wanted inspiration from other solutions to come forward very early in the process. E.g. research on existing e-learning platforms demonstrates the difficulties in getting an overview of the many, many postings students make during a session (a course, an assignment or now a case). For teachers and students alike it becomes difficult to get an overview of the learning process, and it is not a matter of applying simple search-functions, but rather to get an indication of: who always begins a discussion, who makes summaries, which topics from the curricular are prioritised etc. Sketching on basis of other types of applications addressing this difficulty became a way forward. Particular qualitative data analysis applications were inspirational as they use visual representations. Sketching these features in a CaseMaker setting has lead to design suggestions with contents related visual overviews of weighted, dynamic and annotated relations in various spatial and time dimensions. These features also inspired general consideration on what is now called proactive teacher tools [4]

The design sketching process as discussed and carried out in the CaseMaker project is much about user-centred design and human computer interaction, but at the same time the CaseMaker development team had the objective to allow for the more technically oriented perspectives to be discussed. By only using written documentation and very formal modelling and drawing techniques (data models, flow charts etc.) the interaction with the system and the intended feeling of working with the system, which are a vital part of the work context, become blurred. Sketching created a basis for a dialogue, and also provided means for simulating the work context in the design process.

Acknowledgement

The CaseMaker project is funded by CBS, through the E-learning Fund. The first version of CaseMaker was developed at CBS as a joint venture between the author at Department of Informatics and the CBS Learning Lab, in particular Kenneth Andersen and Claus Nygaard. Several external partners, consultants and programmers have worked on the system, and Mads Syrak, from Syrak data as well as Karin Levinsen, a practitioner and researcher from the department of Informatics have both played vital roles as well. Thanks to all and everyone!

References

1. Checkland & Scholes (1990): *Soft Systems Methodology in Action*, John Wiley and Sons.
2. Avison & Wood-Harper (1990): “*Multiview*”, Blackwell

3. Orngreen, Rikke (2004): "CaseMaker – An Environment for Case-based e-Learning" in *Electronic Journal of eLearning, EJEL*, vol 2, nr. 1, p. 167-180
4. Orngreen & Levensen (2005): "Proactive teacher tools for online teachers", in *The 8th IFIP World Conference on Computers in Education (WCCE2005)*, 4-7 July 2005, University of Stellenbosch, Capetown South Africa, 2005, 10 pages
5. Sommerville, I. (1992): *Software Engineering*, Fourth edition, Addison Wesley Publishing Company, USA
6. Snitker, T. (2004): *Breaking Through to the Other Side*, Nyt Teknisk Forlag, Denmark
7. Orngreens, Rikke phd. dissertation on the development of multimedia teaching cases
8. Faulkner, C. (1998): *The essence of Human-Computer Interaction*, Prentice Hall.
9. Bordwell D. and Thompson K. (1997): *Film Art, an Introduction*, McGraw-Hill, 5. edition
10. Hart J. (1999): *The Art of the Storyboard - Storyboard for Film, TV and Animation*, Focal Press
11. Landay, J. (1996): Interactive Sketching for the Early Stages of User Interface Design, Ph.D. Thesis, School of Computer Science, Computer Science Division, Carnegie Mellon University, Pittsburgh, PA December 19, 1996, CMU-CS-96-201 Also appears as CMU-HCII-96-105.
12. Orngreen, R. & Pries-Heje, J (1999): "Designing the design - need-based storyboard adaptation for multimedia systems design" from Proceedings of the 22nd Information Systems Research seminar in Scandinavia (*IRIS22*): Enterprise architectures for virtual organisations, 7-10 August, 1999, Keuruu, Finland. Vol. 3
13. Preece, J.; Rogers, Y. and Sharp, H. (2002): *Interaction Design – beyond human computer interaction*. John Wiley & Sons, Inc., New York, New York, USA
14. Newman, Lin, Hong & Landay (2003): "DENIM: An Informal Web Site Design Tool Inspired by Observations of Practice", in *Human-Computer Interaction*, Lawrence Erlbaum Associates, Inc., 2003, Volume 18, pp. 259–324
15. Backer, D. (1994): "Multimedia Presentation and Authoring", in Buford, J.K: *Multimedia Systems*, Addison Wesley, Publishing Company, ACM Press, SIGGRAPH series, New York, New York, USA, pp. 285-303
16. Hofstetter, Fred (1997): *Multimedia Literacy*, second edition, The McGraw-Hill Companies, Inc.
17. Simonsen, Kensing & Bødker (2000), *Professionel IT-forundersøgelse* (ENG: professional IT pre-investigations), 1. udgave, Samfundslitteratur.
18. Holtzblatt, K. and H. Beyer (1996). "Contextual Design: Principles and Practice." in *Field Methods Casebook for Software Design*. D. Wixon and R. J. New York, Wiley & Sons: p. 301-333.
19. Buur & Bødker (2000): "From usability lab to "design collaboratorium" - designing with interaction styles", in *proceedings of Designing Interactive Systems, DIS2000*, New York 2000
20. Plimmer & Grundy (2005): "Beautifying Sketching-based Design Tool Content: Issues and Experiences" in *6th Australasian UserInterface Conference (AUIC2005)*, Newcastle, Vol. 40. M. Billinghamurst and A. Cockburn (Eds.), 8 pages.
21. Landay & Myers (2001): Sketching interfaces: Toward more human interface design." In *IEEE Computer*, 34(3), 56–64.
22. Conallen, J. (2000): *Building Web Applications with UML*, Addison-Wesley Longman, Inc., reading, Massachusetts, USA

23. Nielsen, L (2004) Engaging Personas and Narrative Scenarios - A study on how user-centered approach influenced the perception of the design process in the e-business group at AstraZeneca-, CBS Ph.D.serie, nr.2004-17
24. Greenbaum, J. and Kyng, M. (editors) (1991): *Design at work - cooperative design of computer systems*, Lawrence Erlbaum associates publishers, New Jersey, USA.