Position Paper for the Workshop TeaDIS - Teaching Design of Interactive Systems

Schaerding, Austria, 20 - 23 May 1997

Teaching Design of Complex Interactive Systems
Learning by Interacting

Gerrit C. van der Veer

Department Computer Science, Vrije Universiteit, Amsterdam;
Department Ergonomics, Twente University of Technology, Enschede, Nederland
Email: gerrit@cs.vu.nl

Michele Mariani

Department Communication Science, University of Siena, Italy;
currently Department Computer Science, Vrije Universiteit, Amsterdam, Nederland

Summary

In this paper we document our experiences in developing and teaching design classes. The courses we teach, and that we still try to improve and try to keep up with state of the art design approaches, originaly developed in close cooperation with colleagues like Michael Tauber and Steve Guest. Only in a way of cooperation we will be able to improve and to produce state of the art education.

Teaching interactive systems design in our situation means teaching various groups of university students, as well as groups of experienced practitioners, in most cases stemming from a variety of disciplines like software engineering, electrical engineering, cognitive psychology, and AI.

Our current design classes are organised in such a way that students are forming a design team with subteams for different specialist design methods like task analysis, formal modeling, prototyping, usability evaluation, and requirements analysis. The team collaborates in an iterative manner, starting from an initial statement from a real client and ending with the presentation of a complete design (including design rationales, working prototype that is evaluated, but also including organisational re-design and possible video scenarios.

1. Introduction

complex interactive systems

In the early years of human-computer interaction, interactive systems could still be described as single stand-alone PCs used by a single user for performing certain well-defined tasks like producing documents, performing calculations, or searching personal databases. Gradualy, computers became part of networks, and tasks for which information technology is applied started to include complex work activities that can be characterised by elements like cooperation, communication, management of work, group decision making.

In our approach towards design we consider human-computer systems that feature in situations of work (including education) where people work in groups, organisations, and structures of interdependence (e.g. market situations with several parties). Work activities and tasks in these cases include communication between people, coordination, and actions of several perons on shared objects and in shared work spaces. Information technology, in this environment, is used bu many partners for a variety of tasks. The information technology will often be distributed among several or many machines, and will show an architecture of complex functionalities and complex data structures. This situation tends to be the "normal" situation of human-computer interaction in the future, and this is why we focus our education of design on "complex interactive systems"

design methods

In our approach we need to make a clear destinction between the concepts of method, technique, and tool, and, moreover, in teaching we feel the need to make choices. Design methods, in our terminology, specify how the total of all design activities are organised and what kind of people are participating in it (the "stakeholders" of design).

In a design method a preliminary choice is made for the type of techniques that may be used. The relation between the techniques is established, both in time, and in information transfer between the techniques applied. A design method for interactive systems may be based on the gradual development of a prototype towards the final implementation for actual use, like in "rapid application development" (Martin, 1991). In that case, techniques will be needed that provide reusable code, even in early phases of prototyping, and testing needs to focus from the start on both the functionality of the final system, and on the system's image towards the user. Other methods focus on a fixed sequence of design phases ("waterfall approaches"), or, contrary to this, on a continuous interation between phases of analysis, specification, and validation.

Another source of difference between design methods concerns the stakeholders. The "Scandinavian school" advocates user participation, which, in certain situations, includes user involvement in major design decisions. Some methods require the involvement of Ergonomists as managers of interactive systems design in general (McClelland, 1993), other advocate a core role for ethnography (Jordan, 1994).

The method we teach features desing as an interative structure of techniques. The central phases of analysis, specification and detail design will iterate, based on the outcome of evaluation techniques applied during these phases. Also, the input of certain sources like the client of design will be subject to subsequent revision leading to the need to negotiate again on requirement specifications. Our method is based on the input of a considerable set of disciplines, both stamming from the domain of humanities (cognitive psychology, antropology and ethnography), and several domains of engineering and design (e.g. organisational design, software engineering, graphical design). In this respect we do not intend to prescribe the complete set of disciplines, since, in our view, the actual nature and the situation of the interactive system to be designed will influence the choice of the stakeholders needed. regarding user involvement, our method starts with analysing users and their task situation in the current situation (if there is any to be identified), analysing future use and future users, and involving both future users, staksholders of the system to be designed, and clients of the design (people or organisations who commit the design or pay for it).

techniques, and tools

In a design method like we are dealing with, many different techniques may be applied, some of them depending on the actual situation: collecting information on the current task situation, analysing expert knowledge, several techniques of global specification and detail design (ranging from design space analysis, to graphical design and formal specification of artifact functionality), different variants of early representation of the system under development (scenario, simulation, early prototype), and many forms of evaluation and usability testing (subjective evaluation by prospective users, cognitive walk-through by human factors experts, formal evaluation, application of norms and standards).

Applying many different techniques, used by specialists from different disciplines, brings the need for a management structure and a method to control the design enterprise. Decisions need to recorded, specifications and feed-back needs to be formalised in order to communicate between disciplines and between viewpoints. Complex design projects of this type will frequently ask for decisions where different viewpoints result in conflicting criteria. Each decision will need the identification of an "owner" of the decision, and a well defined structure of responsabilities communication of decisions, as well as the continuous documentation of teh rationale and the details of specifications handed over between design phases.

Complex design of this kind requires tools for management, communication, and documentation, developed for collaborative design by an interdisciplinary team of design experts who frequently need to interact with current or prospective users, clients, and stakeholders of the system, on specifications and evaluation matters.

Complex systems design in the sense of our approach and method requires the education of designers, in respect on how to deal with a variety of disciplines, a variety of viewpoints, and the continuous iteration between analysis, specification, and evaluation. Design of this type can only be learned by doing, i.e., by participating in actual design experiences of real life size, in an interdisciplinary setting.

2. A background for gaining design experiences

Learning how to design is more than learning about methods. It is learning how to choose which is the most appropriate technique to cope with a specific design problem in a specific design phase. In this respect design is a situated activity, that is to say that it cannot be planned and fully specified in advance. In such a context the capabilities to play a role inside the team and to communicate in a suitable way are much more critical than the "how-to-do-it" kind of knowledge. Such abilities are even more important when making the design user-centered, because of the many activities that require the users' participation.

situated design

Complex interactive systems are strongly situated, and so is any actual design of them. E.g., in some cases there is no possibility to approach prospective users, in other situations, certain decisions are prohibited by constraints from the client. Information technology and the models of application in complex systems develops so fast that it is impossible to define a standard set of techniques. The only sensible way to proceed is to try to be aware of the state of the art of techniques as well as their applicability in actual design processes, and to constantly be aware of the restricted validity of this knowledge.

Recent publications of the assessment of design techniques and methods should be utilised to identify possibly relevant candidates as well as the related problems and the utility of the results. Consequently, designers of complex interactive systems should permanently maintain their knowledge by updating their awareness of literature, identifying the value of each account regarding the relation towards theory of design as well as practice of the technique, assessing the value of each technique both in relation to the product of design and to the process of designing, always keeping in mind the contribution of the actual situation of design towards the value of the technique. In this respect, case studies of design as a process may be of more value than scolarly accounts of a method and its theoretical basis.

Based on insight in various design processes and their situational factors, one needs to learn to choose among available techniques, both in the domain of ones own discipline, and in multidisciplinary collaboration, where the result of design activities are constantly being handed over between different expertises, and where the aplication of any technique is dependent on the availability of information and specifications stemming from others' activities. This implies that one needs both to be able to compare available techniques from ones own domain of expertise, and to specify requirements for the choice and application of techniques by other members of the design team.

A choice for state of the art techniques often is accompanied by the need of mastering new tools. A designer of complex interactive systems needs to accept the burden of permanent education and training.

collaborative design

Participating in the design of complex interactive systems presupposes that one is aware of the specific characteristics of working in a multidisciplinary team, where different, and even contradictory, viewpoints may be valid based on the nature of the expertises involved in a certain decision. Striving for unanimity often does not make sense, but, on the other hand, consistency of specifications needs to be a final goal. Complex decision making will require the ability and willingness of considering the nature and boundaries of the design space. Various domain experts need to contribute their own views on possible actions and options for specifications, as well as need to be open to consider those contributed by other disciplines and competences. In the same way, the criteria for assessing the decision will often need to be collected from various stakeholders in the design process. An openness for "alien" options and criteria needs a basic trust in the competences of all disciplines involved in the design, a trust that can only be developed by having experienced at least a basic insight in the goals, viewpoints, methodology and vocabulary of the various expertises at stake. The other side of this is the willingness to provide suffient insight in ones own expertise and approaches to the extend of explaining the viewpoints and related techniques and of accepting to be questioned about ones rationale and professional values.

Apart from understanding the ever developing fields of the different disciplines involved in design, one needs to be able to collaborate. Basically, this requires the flexibility to communicate, at all levels of detail that are relevant in the design process, between disciplines, as well as to document all design rationales and resulting specifications for different audiences.

a need for basic knowledge

Learning to design complex interactive systems requires some insight in basic disciplines like humanities (at least cognitive psychology and antropology) and engineering sciences. No introductory course on interactive systems can do without this. Apart from some theoretical insight, some skills from the various domains have to be mastered, like the basics of experimental design and empirical data collection and the related application of relevant statistical methods, the use of formal representations, and some experience in high level programming.

In order to understand the different approaches and methods that feature in design, it it needed to collect understanding about the different "schools" of HCI and CSCW. In this respect students should be aware of at least some of the major developments in cultures where English is not the first languages (and, hence, where publications are often not available from the "main" journals and conference proceedings). E.g., developments in the French and German speaking parts of Europe may need some extra effort to identify. The same is the case for relevant developments in South-east Asia.

Most relevant developments may be identified through the important journals, conference series, and series some publishers maintain. The Web has developed into a main source, with important collections of references like Perlman (1997). Some of the major conferences have developed into a source for permanent education, with tutorials at various levels, where students may register at strongly reduced prices, or from which tutorial notes may be bought separately. Develping a map if sources of information and education has to be a major goal of introductory teaching in HCI. But even here, it is not so much the "knowledge" of relevant pointers, as the way to keep up to date with developments, that has to be mastered.

how to get started

Still, in order to start participating in real design experiences, students need a starting kit. Introductory education needs to provide an introductory "encyclopedia" of techniques and tools for design of interative systems. Some of the European projects in the domain of design of interactive systems did produce products of this character (Esprit projects like Amodeus, Race project ISSUE), as did teams at institutes like Virginia Tech and Philips Design. As long as such collections and accounts of methods, techniques and tools are not presented as permanent state of the art accounts, they provide valuable sources for techniques that may be of relevance depending on the actual situation of design. Other sources of knowledge of techniques are to be found in readings that are devoted to single phases in design, like task analysis, (Kirwan and Ainsworth, 1992; Diaper, 1989) graphical representation design, or evaluation or selected aspects of design methodology like management of design (Lim and Long, 1994; Moran and Carroll, 1996; Newman and Lamming, 1995) or usability engineering (Nielsen, 1993; Shackel and Richardson, 1991)

Teaching this "knowledge" or providing the educational material for introductory learning on interactive systems design requires a strong focus on the relativite validity and applicability of methods, techniques and tools. Technology changes rapidly, and so does the scene of design, and, moreover, design is strongly situated. There are now "best" techniques, and what is a good choice today may be below optimal tomorrow. Any introductory education or learning material should convey this message from the start: designing interactive systems requires permanent self education and permanent awareness of the changing world and the changing map of sources of state of the art information.

3. A structure of design activities

In this section we will briefly describe our "method", not so much to advocate this approach as the best, but to illustrate furtheron that the choice of a method may provide the framework for learning to design for a varyity of situations. A method is needed as a framework for relating situational factors and techniques to be applied in complex design activities, not as the goal for education. For a more detailed account on our approach, see van der Veer, van Vliet, and Lenting (1995).

modeling complex systems

Van der Veer (1994) and van der Veer, Lenting, and Bergevoets (1996) derive a structure of concepts to model the complex systems. The framework considers 3 viewpoints: (a) people, (b) work, and (c) situation.

a. people

Modeling people is based on specifying types of roles and their task-related attributes, specifying an actor type and its attributes (actors are individual users), and specifying the organisation, i.e., the relations between actors and roles in the task domain. The specification of these people related concepts will include relations to the concepts that are specified from the other two viewpoints.

b. work

Modeling work means specifying tasks and a task structure, actions that relate to tasks, and protocols and strategies (to be identified in the case of real world situations) or procedures (specified when task structures are designed) to indicate action structures related to situations and roles. Each of these types of specifications is done by coupling relations to the work concepts, and to the concepts regarding people and situations.

c. situation

Task delegation to information systems, from the users' points of view, should be considered "situated", i.e., the situation has to be modeled and designed as part of designing the user interface and functionality. Modeling the situation means specifying object types (in the meaning of "things" people manipulate in the course of performing or delegating tasks), the structure of objects (both the type hierarchy and the semantic relations between objects), and specifying the situation where certain tasks are performed, being a space where people act in relation to each other and on objects.

phases in modeling

If the system to be developed will feature in a complex task world, we need to model this, including the 3 viewpoints mentioned, i.e., the people (users and user groups), the work and task structure, and the situation. In our approach we further structure this activity into the development of two different categories of models, which we label task model 1 and task model 2.

Task model 1 represents the "current" task situation, i.e., the actual structure and organisation of activities including the artifacts and technologies that are available and used. In order to develop task model 1, several methods of data collection collection may have to be applied. At least we need to make a destinction between expert knowledge (to be derived by knowledge acquisition methods for which we need acces to expert users in the current task world - these methods may vary from structured interviews to hermeneutic analysis of observed expert behavior), knowledge of documents (official business regulations and procedures, descriptions of the organisation, business goals; all of these may in some situations be found not to reflect actual practice), and experience in real life practice (for which ethnographic methods like interaction analysis need to be applied)

Task model 2 models the task domain of the future situation, where the system to be developed would be used, including changes in organisation of people and work procedures. The relation between task model 1 and 2 reflects the change in the structure and organisation of the task world as caused by the implementation of the system to be developed. As such, the difference is relevant both for the client of the development process and for the user. The development of task model 2 from task model 1 is an instance of specification, in this case based on knowledge of current inadequacies and problems concerning the existing task situation, needs for change as articulated by the clients, and insight in current technological development. As such the development of task model 2 is a case of compromises between different values, e.g., regarding available resourses from the point of view of the designer, regarding ease of user for the end user, regarding aspects of implementation and maintenance from the point of view of technological solutions.

The information system to be designed is specified in detail as a concequence of task model 2. Tauber (1988) and Van der Veer et al. (1985) indicate this as the UVM, the user's virtual machine, denoting all aspects of a system the user should be aware of during interaction (including planning and evaluation of interaction). It has to be modeled in all details that are relevant to the users including cooperation technology and user relevant system structure and network characteristics. In this perspective we need to specify the functionality (as modeled, e.g., by the semantic level of Moran (1981) or by the object structure and basic task structure of ETAG (Tauber, 1990).

Apart from the functionality of the UVM we need to specify the interaction between user and system, with its two directions of communication that we label the "language interface" (modeling the language in which the user expresses himself to the system, the style of dialogue) and the "presentation interface" (modeling the system's actions and representation of relevant information for the user). These 3 components of the UVM have to be distincted in the activities of design. The 3 types of design decisions resulting from this, each of which is a different class of models, obviously are strongly related in their consequences for the user, hence, the activity of maintaining consistency is a further important design activity in the phase of specifying the UVM.

The specification of the new task world (task model 2) and the new information system system (UVM) includes many design decisions that have to be considered in relation to prospective use. The activity of evaluaton will therefore be mandatory in parallel to each of these activities. In some cases of design decisions, guidelines might be of help, in other situations formal evaluation may be applied. Formal modeling tools like CCT, TAG, or ETAG may provide an indication of complexity of use or learning effort required (de Haan et al., 1991).

In many cases of design decisions, however, evaluation can best be done by confronting the relevant aspects of the intended system with the understanding, behavior, and feelings of the future user. Some kind of scenario or prototype will often be the best way to confront the client or the future user with what solution is proposed. A scenario enables representing the envisioned new task world and the new technology in sutiations of future use, including occasions of exception handling and disasters of varous kind (what if the plant is on fire ...). A prototype allows experimentation with selected elements of the different models that are developed for aspects of the UVM. It may enable imitation of (aspects of) the presentation interface, enable the user to express himself in (fragments of) the interaction language, and can be used to simulate some aspects of the functionality, including organisational and structural characteristics of the intended task structure. Hence, prototyping and related early evaluation is another activity that has to be explicitly distinguished.

design as team activity

For the past 7 years we have been performing design excercises that were intended to cover the whole set of activities mentioned in the previous section. Based on this experience, during courses in user interface design at a university and a technical university and in post academic courses for industrial system designers as well as on actual design experiences in industrial settings (e.g., Philips Design), we developed a view on structuring design teams in relation to the design activities mentioned.

Figure 1 illustrates our team model. The main types of design activities and the classes of models involved are grouped in relation to stages in design. It seems worthwile to make distinctions between major activity clusters and assign each to a separate (sub) team:

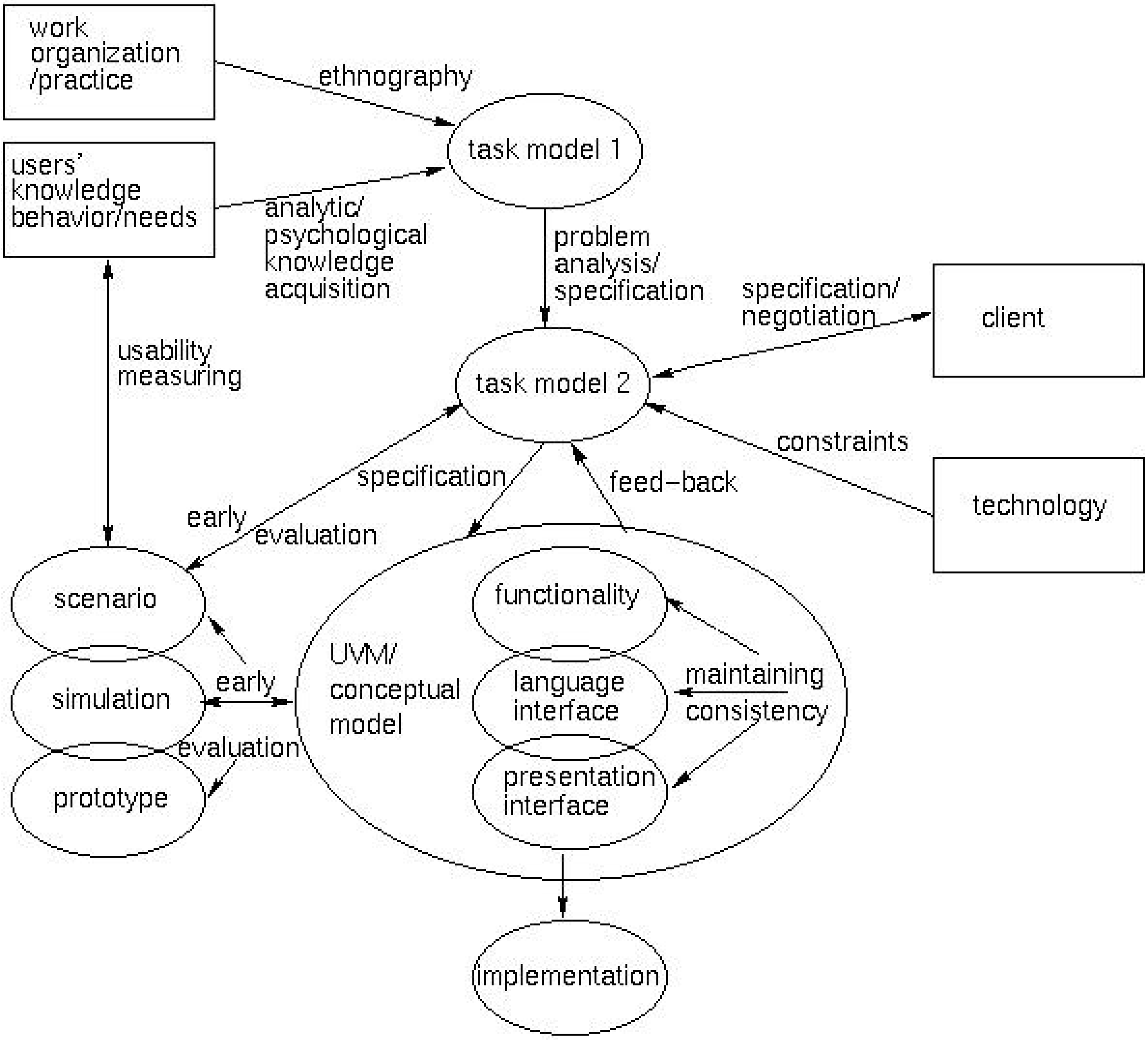

Figure 1. Design as a structure of activities

a. analysis

Task model 1 is developed on the basis of knowledge acquisition activities. Techniques that might apply may be related to knowledge ellicitation and psychological techniques, or be derived from antropology and ethnography. In cases of complex task situations, several different disciplines need to contribute. The final goal of this group of activities is to analyse, describe, and model the current task world.

Task model 2 is the result of analysing task model 1 on its values and problems for the user, confronted with the requirements from the client of the design process as well as with the technology availability and constraints. The aim of this set of activities is a specification and model of the envisioned new task world.

b. design

If there are different classes of users (different roles) and possibly stakeholders ("indirect" users of the system, i.e., groups of people who's work is influenced by the implementation of the new system) there will be a need to specify a separate UVM for each role. The specification of the UVM can be structured into modeling the functionality, which includes specifying the information system details as far as relevant to the user: what the system provides for this type of user - application design is strongly related to this part of UVM design;

modeling the language interface, which means defining the type(s) of dialogue style for interaction, influencing the kind of control the user feels to have over the system, as well as the amount of learning needed to operate the system - insight in human communication phenomena (psycho-linguistics) may be a worthwhile discipline;

modeling the presentation interface, including the look and feel of the technology, the specification of the graphical interface, and any representation of the system and its products, which strongly influences the ease of use and, again, learnability - graphical design is a specialism involved in this type of activity.

The development of the UVM is based on detailed specification of the "new system" on the basis of task model 2, and this will inevitably induce the need for feed back to task model 2. If design decisions force the specifications to deviate from the original global intentions, this has to be considered explicitly. Changes in the model of the information system may result in changes in other parts of the task world, like organisation of people's work, communication structure, and work procedures. If this would cause task model 2 to deviate from what is considered an adequate answer to the request for system development, this could lead to reconsidering the whole set of decisions taken sofar. Feedback to task model 2 is a design activity that has to be coordinated and continuously controled.

c. evaluation

Scenario development, simulation, and prototyping are based on specifications derived from specific or general evaluation needs that emerge during design decisions on task model 2 and the UVM. Analysing their claims and their indications on usability will return evaluation to the previously mentioned modeling activities.

Next to the specialist activities of developing these different representations of design specifications, evaluation may need a variety of testing techniques, from formal analysis and cognitive walkthrough to user testing, observations in usability labs, and the application of norms and standards.

Again, different specialisms will be needed in an evaluation phase, and a variety of expertises will be needed from ergonomists to hardware and software enginineers and programmers.

Implementation could be the task of another team, based on the specifications that eventualy result from the UVM when evaluation has been satisfactory. In fact, we never actualy performed this activity during our exercises.

d. managing the design

As already mentioned, in the case of distinguishing these activities a lot of "traffic" will have to go between the different teams that take responsibility for certain parts of the work. Consequently, another activity that could be identified, and that in practice turned out to be of major importance, is the management and coordination. Management includes controling feedback, guarding consistency between the different aspects of the UVM, monitoring evaluation, and controling specification as it takes place in various transfers between activity clusters.

4. Teaching design: an educational program

multidisciplinary design teams

Designing interactive systems needs collaboration between various disciplines. In most of our classes we are in the situation to practise this from the start. Our design classes most of the time contain a variety of students. Often about half of them are working on a major in Computer Science (where either Software Engineering, or Information Systems is the main topic), there are mostly some Psychology students (either focusing on Organisation and Labor Psychology or on Psychonomic Science) and some Artificial Intelligence Students. Apart from this, there may be expected students from Business Informatics, and sometimes from Electrical Engineering.

The size of classes varies between 10 and 30 students. If there are more than 30, we split the group. There are most of the time 2 teachers, who do not teach too much, though, but mainly act as consultants on various aspects of design techniques and on general design methods. In most cases, there are other experts around, like visiting scientists who work in the domain, and PhD students in related topics. During the course consulting these various experts is strongly encouraged. Many students are not used to this, and have to be convinced that using the available expertise is not a sign of weakness. On the other hand, students have to get used to the situation where the teacher is not willing to provide the "best" technique, but, on the other hand, is open to any suggestion for new approaches and creative combinations of techniques.

a "real life" design case

We try to provide a real life experience. To this end, in some cases we are able to offer a real design case from industry. In other situations we provide a realistic design request and a "client" who is able to act from his personal work situation. In those cases, the design excercise is often performed in a context where there is a serious intention to have a new situation or system installed in the near future. Finally, in some cases students come with their own design case, based on job situations some of them are involved with, or based on a need for (re)design of systems they have to interact with in their everyday life. In this way, there have been design cases for a public library, a photo-copy shop, a taxi and bus company, and a system for a steel factory, as well as design excercises for a book selling system, several university administrative systems and university facilities. In most of these cases, the class ended when a complete design was produced, prototyped, and tested on usability, though, in some cases, the design has been taken further outside of our educational setting.

In all cases, there was a "client" who was briefed to behave as realistic as possible. The expectations of this client were carefully monitored. Also, students were aware of the status of the client's responsabilities to his organisation, and of the status of prospective users of the system, and on how to behave towards any stakeholder of the design. The client, and, if relevant, prospective users were invited for the final presentation of the project, and received the full design documentation, as well as any demonstration and testing they required.

structure of the course

The structure and content of our classes presuppose insight in HCI approaches and in some of the basic related disciplines. In most cases, an introductory course is a requirement for the participation. Moreover lecture notes are provided (that, in most cases, are identical to the ones prepared for the introductory courses), that contain, among other things, a brief introduction to relevant aspects of cognitive psychology and ergonomy, a brief introduction to the most relevant "schools" in HCI and CSCW, pointers to literature on design, an account of a series of relevant techniques, and reprints of relevant readings to start with.

The first meeting of the course (re-)introduces our design approach as elaborated in section 3. The students are made aware of the purpose of the class: gaining experience with a real multidisciplinary design project for interactive systems, and are explained the constraints of participation and the way the results are evaluated. Also, the documents and deliverables from the total design team and from the various sub-teams are explained as well as their roles in the evaluation of student performance. Deliverables may include: a final design document, including design space analysis for important design decisions, and appendices containing formal specifications and representations of intermediate results; transcripts from interviews with prospective users, videos from ethnographic analysis and of scenarios acted out by designers and prospective users; prototypes and simulations; and an internal management report of the type that a design company would need to keep for resource management over different design projects.

A tentative set of milestones and an example timeline is suggested, but it is made clear that students are themselves responsible for the actual variants of these. It is made clear that planning depends on the actual design situation, including the requirements from the client and the results of the first negotations with him, as well as from the available resources in expertise among the team, and on their own interpretation and creativity. Finaly, the mutual responsabilities and dependencies are stressed. Often, there is a small number of students who does not formally participate but want to observe the process. They are welcome to do so, and to participate in any activity that the participants agree on.

The above mentioned aspects are the only formal teaching involved. After this, and after explaining the possible structure of a design team, we start to identify students who may take the role of managing the design team. As soon as we have identified 3 people who are eager and seem to be skilled enough (to the subjective impression of the teacher), we hand over the project to them. We still provide them with the design case and identify the client (as far as there is no proposal from the students), and suggest them to start recruiting their team from the students who committed to fully participate in the course. From here on the management of the team is in charge of the assignments. Some managers start to ask each student to provide a (written or email) statement of their background and expertise, and their wishes on what design phases to be involved in. Some other managers have interviews with all to enable them to do optimal resource management.

The next 3 to 5 months (we teach this course in different universities, with variations in the structure of the academic year) contain a series of about 10 formal meetings where the teachers are present. Apart from this, there are rooms available for group meetings, and a variety of machines for using the various tools for task analysis, formal specification, prototyping, for using the web and searching libraries, and for documentation and exchange of emails are made available.

For the formal meetings we request the management to provide the program, but we stress that we (teachers) at least expect presentations of the various sub-teams about the methods and techniques they are considering. Also, we provide comments on the techniques during these meetings, as well as on the actual course of the project.

Detailed questions on techniques to the teachers are not answered on these metings, but are refered to consult sessions. The results of these consults may be part of presentations of groups later on in the formal scheduled meetings.

The management will organise other meetings, both with the whole group, with single subteams, and in some cases with a single representative of each of the groups, whenever they feel the need for coordination and for decision making.

Evaluation of students' performance

From the start of the course it is made clear that the students (as far as they ask for this - there are often observers or participants who do not need any grade marks) are reviewed on 3 aspects of their performance during the course:

- the presentations on methods and techniques during the formal classes. This is mainly a collective product of small subgroups;
- the identifiable stake each subgroup has in the final design (where an account of the process and the design space analysis is more important than the actual resulting design decisions);
- the internal management review produced by the management group based on input from this subgroups: planning documents, progress reports, communications.

Consequently, the evaluations are mostly for a subteam as a collective group. Only when (from the internal management report) it is clear that single participants performed significantly better or less acceptable than their group, or when individuals were reallocated between groups or helped other groups out in difficult situations, the teacher will make a unique assessment for the individual concerned.

team structure

The team structure, as suggested in the first meeting, closely follows the structure of design activities as represented in figure 1. It is told that at least some analysis of current situations may be useful, and that this could be acomplished by various methods, where obvious candidates would be a (small scale) ethnographic analysis as well as a systematic interview of available experts. Consequently, this could lead to either a single, or several differently oriented, subteams for developing task model 1. The management team is free to decide whether this same group will also be responsible for specifying the future task situation (task model 2) or whether another subteam should be assigned for this job. Anyhow, the two activities can be both facilitated by using the same task modeling tool (van der Veer et al., 1996) where task model 2 may be developed based on (among other sources) a formal representation of task model 1.

The next cluster of activities relates to detailed design (the UVM part in figure 1), where a single team may perform all activities, though, alternatively, there could be a split in tasks where, e.g., functionality is specified by a separate group from sub-teams who consider dialogue design and graphic representations, and where, if needed, a special individual can be assigned the task to manage the consistency of the UVM design activities.

The final cluster concerns evaluation techniqeus, ranging from the development of special representations like scenarios and prototypes, to all kinds of testing, from cognitive walk through and formal evaluation to subjective usability measurement and the evaluation against norms and standards.

In actual classes it is the management, in collaboration with some of the subteams, who finds the optimal team structure. In our most recent class, the original idea was to have one single group for scenario and prototype construction and another one for all evaluation activities. After several discussions amont the students concerned and after these groups had given their presentations on methods and relations between different activities, they decided to re-allocate the tasks: one group specified scenarios and tested them during the development, another focused on prototyping which involved in this case a considerable programming effort, and a last group performed all types of evaluation on the prototypes only.

design as theory and practice

For each design activity the assignement to each group was for a three step process: first a presentation on the different theories about the topic, then a presentation on how a selected technique was applied in the case study. Finally the group had to provide a complete documentation that should detail at least the work done and the rationale for the decisions that were taken (e.g. on which criteria a decision was taken to choose an euristic evaluation instead of a formal one). Following such a path, learners have the possibility to view theory in front of practice.

Another effect of this structure is that, early in the life of the team, each subgroup shows its "expertise" and specialism to all other groups, developing the trust needed in multidisciplinary teams to accept arguments, options, and critics from others who seem to speak another "language". A certain basic insight in the various expertises of the "other" disciplines is needed to be able to realy collaborate, and, most of all, to be prepared to take decisions where different viewpoints have conflicting evaluations of outcomes, as is often the case in this type of complex design situations.

Finaly, there is an educational merit here: students turn out to be better educators than "expert" teachers can be, since they just mastered the material themselves and they still have a vivid memory of the realy new aspects of any technique they just discovered and are now investigating for possible application.

5. Some observations from a design class

communication means and problems

Along the design class people have to act as groups that must cooperate and cope with quite strict deadlines. As the class went on, the management group set up the means to cooperate and communicate. A management plan with deadlines was set up, a web page was built and a mailing list was made available for any communication necessity. This, of course, did not solved all the problems as the following excerpt from an e-mail message, sent by one of the groups, shows: "... have you ever heard of communication?...there is such a thing as speech and e-mail and the use of it for clearing up problems and keeping a tight knit on group activities when the group is rather large is common and considered polite by most of the human species. This is a working group in which the different parts of the organization should know what the other ones do, and in case of doubt step up to that organization and ask for explanation; that's for instance why all those presentations about theory are held, to make it obvious for everyone what the respective groups are doing and why ... ... could have been avoided if you had communicated about this with us instead of acting like you are the only person working in the project when in reality there are about 20 ...".

Interestingly enough, students autonomously highlighted the problems that the (relatively) new communication means can raise. Initially, e-mail was taken as a valid substitute of every other means (e.g. phone calls, face to face meetings). As the time passed, the same e-mail was labelled as and "indirect" means for communications, as opposed to "direct" ones (e.g. phone calls). Sending a message for committing someone to a particular task was discovered as very ineffective. Negotiation on committments were acknowledged as much more effective when direct, either by phone or by face-to-face meetings.

social processes and role playing

Giving the class the responsibility for a real case to be performed in a given time, students face with real life difficulties. It became apparent that they were no more performing an exercise. On the opposite, they experienced a full-immersion in their roles. They refer to themselves as groups: 'the language group urgently need to have a meeting with the presentation group to clarify the integration of our respective (and sometimes overlapping) tasks'. Indeed, as in everyday reality, the management group immediately got some distance from the other groups, and, of course, was 'hated' by everyone ('they are too rigid ... they just take care of deadlines ...'. In addition, each group showed a high level of internal cohesion, being very defensive of its own job. Each time some difficulties did arise, the team put the blame on the previous (in terms of project flow) one. Such an attitute sometimes seriously undermined the positive completion of the assignement, living rooms for oppositive rather than coooperative attitudes.

Feedback on methods

Too often lessons tend to be 'normative': there is one and only one well specified good way in which things have to be done, an you are supposed to learn about it. As an alternative, you can learn that there are a lot of different methods, each one with its own advantages and disadvantages and, again, you are supposed to learn about the different methods and their plus and minus. This second 'philosophy' is always preferable to the first one, but still less preferable than the one we are describing. By putting hands-on a real project, students discover by themselves the advantages and disadvantages of the different methods, which do not always overlap with the ones described in the books. The functionality group of the current year's class, for example, facing with the different methods for functionality specifications, decided to come up with a hybrid, new notation, that satisfied them more that the others already available. To quote Newman and Lamming (1996): "When none of the available notations is capable of supporting our design activities, we look for new methods. This is how all of the existing methods originated: solutions were devised that existing notations could not describe."

6. Conclusions

design as role-playing

Design implies the ability to play a role within a team. This is possible only if design is seen as a structured activity in which different groups hold different roles. Giving the class a framework to organize the design process, learners demonstrated the ability to be a real project group. In such a learning environment the teacher's role shifts to a consultant one while the learners hold the whole responsibility for the final result.

learning instead of teaching

Anyone who ever tried to cook something knows this: recipes will never tell you the whole story. Design is a complex activity: it cannot be neither fully planned/specified in advance nor comprehensively taught/described in a class, as good cooking. Design, in addition, often involves teams of people with very different expertises. They have to coordinate their activities toward a common goal. A general design framework (like figure 1) proved to be very effective in supporting the coordination of the different design activities/teams. Each group could develop a specific expertise, targeted on a given aspect of the user interface. Competence, however, is far from being enough for an effective design process. Communicative processes and social dynamics handling are those part of the recipe that are very difficult, if ever possible, to be taught in a satisfactory manner. They have to be learned by interacting.

7. References

McClelland I.L. (1993) The applied ergonomics group at Philips. INTERCHI'93 Conference Proceedings. Acm, New York. 461-462

Diaper D. (ed) (1989) Task analysis for human-computer interaction. Ellis Horwood, Chichester.

de Haan G., van der Veer G.C., and van Vliet J.C. (1991) Formal modelling techniques in human-computer interaction. Acta Psychologica 78, 27-67.

Jordan B. (1994) Ethnographic workplace studies and CSCW. Report IRL94-0026, IRL, Palo Alto.

Kirwan B., Ainsworth L.K. (1992) A Guide to Task Analysis. Taylor & Francis, Londen.

Lim K.Y., Long J. (1994) The Muse Method for Usability Engineering. Cambridge University Press, Cambridge.

Martin J. (1991) Rapid application development. Macmillan. Moran T.P. (1981) The command language grammar: a representation for the user interface of interactive computer systems. IJMMS 15, 3-50.

Moran T.P., Carroll J.M. (1996) Design Rationale. Lawrence Erlbaum Ass., New Jersey.

Newman W.M., Lamming M.G. (1995) Interactive Sytem Design. Addison-Wesley, Cambridge.

Jakob Nielsen (1993) Usability Engineering. AP Professional, Londen.

Perlman G. (1997) http://www.hcibib.org/

Shackel B., Richardson S. (1991) Human Factors for Informatics Usability. Cambridge University Press, Cambridge. Tauber M.J. (1988) On mental models and the user interface. In: G.C. van der Veer, T.R.G. Green, J.-M. Hoc, and D. Murray (eds), Working with computers: theory versus outcome. Academic Press, London. Tauber M.J. (1990) ETAG: Extended Task Action Grammar - a language for the description of the user's task language. In: D. Diaper, D. Gilmore, G. Cockton, and B. Shackel (eds), Proceedings INTERACT "90. Elsevier, Amsterdam.

van der Veer G.C., Tauber M.J., Waern Y., and van Muylwijk B. (1985) On the interaction between system and user characteristics. Behaviour and Information Technology 4, 284-308.

van der Veer G.C. (1994) Groupware Task Analysis: Modeling Complex Reality. In: R. Oppermann, S. Bagnara, D. Benyon (eds) Human-computer interaction: from individuals to groups in work, leisure, and everyday life. GMD-Studien Nr. 233, GMD, Bonn.

van der Veer G.C., van Vliet J.C., Lenting B.F. (1995) Desigining complex systems - a structured activity. Proceedings DIS '95. ACM, New York. 207-217.

van der Veer G.C., Lenting B.F., Bergevoet B.A.J. (1996) GTA: Groupware task analysis - modeling complexity. Acta Psychologica 91, 297-322.

Edited: 29 April 1997 - Gorny