Comparing Interaction Design Techniques : A Method For Objective Comparison To Find The Conceptual Basis For Interaction Design

Mark van Setten
Dept. of Ergonomics
University of Twente
PO Box 217
NL-7500 AE Enschede
+31 53 4893325
m.vansetten@pi.net

Gerrit C. van der Veer
Dept. of Ergonomics
University of Twente
PO Box 217
NL-7500 AE Enschede
+31 53 4893326
gerrit@cs.vu.nl

Sjaak Brinkkemper
Baan Company R&D
PO Box 250
NL-6710 BG Ede
+31 318 696666
sbrinkkemper@baan.nl

ABSTRACT
Part of designing the User Virtual Machine is designing the interaction between the user(s) and the system. There already exist several techniques for designing the interaction, but, once applied in practical situations, all have problems. The use of a formal comparison method combined with experience in interaction design shows that there exists a conceptual basis for interaction design. The method to find this basis is a structured approach which describes each technique objectively, compares the concepts, relations, purposes, and places in the design method. Based on this comparison the conceptual basis for interaction design can be created, which is adaptable to the design situation at hand.

Keywords
Interaction design techniques, comparison of techniques, method engineering, situational methods.

INTRODUCTION
Part of the design of user interfaces is designing the interaction between the user(s) and the system(s). To design and describe this interaction there exists a variety of interaction design techniques, e.g. Command Language Grammar (CLG) [7], External-Internal Task Mapping (ETIT) [8], Goals, Operators, Methods and Selection rules (GOMS) [4], Task Action Grammar (TAG) [10], [11], and Extended-Task Action Grammar (ETAG) [13]. Most of these techniques have problems when used in actual design situations. Some of these problems are mentioned in the first section of this paper.

On the other hand, the currently existing techniques appear to be useful in some actual design situations [1], [16]. A solution to the problems is to develop a "new" interaction design technique, based on the crucial concepts and relations between those concepts of the existing techniques. To this end the existing techniques need to be compared objectively. The second section of this paper describes a method to objectively compare design techniques. The result of such a comparison is described in the third section of this paper.

The new technique should be able to describe that part of the interaction process, which is important to the user and designer. To this end, a theoretical basis that describes this interaction process is necessary. The search for a theoretical basis, derived from experience in actual design, and the integration in the design method is briefly mentioned in the third section of this paper.

First the place of interaction design and the comparison of the interaction design techniques in an overall user interface design method are described.

PLACE OF RESEARCH
In this section we will analyze the problems. First of all we will briefly show our method of designing complex systems. Secondly we will focus on the specification of technology (designing the UVM). Finally we will present some of the major problems we met when analyzing techniques for the dialogue part of the UVM (interaction design).

Designing Complex Systems
Designing complex systems is a complex activity. Current human-machine systems will mostly not be a one-person-one-artifact system. Humans work in organizations and complex social structures. Modern artifacts like information technology are increasingly part of networks, whether physical or via transmission through floppy disks or incidental modem connections. It does not make sense

to design for the single user, without considering the organization of which he is part and the business process in which his tasks and role(s) are incorporated. Likewise, the user interface should not be considered as only the front end of the single PC. Keyboard, mouse, screen, microphone and camera are all part of the gateway to a complex workspace that is inhabited by various colleagues and stake holders to the individual's work, and that is spread out in time and space, with many discontinuities of connecteness and access.

Designing such a complex system means designing the user's virtual machine (UVM), i.e., specifying what the user needs to know about the organization, the work structure, and the shared workspace that is related to the user's work station. Consequently, design turns out to be a very complex structure of activities. Van der Veer et al. [14] provide an analysis of different design activities and methods and propose a design method that can be used in all phases of design.

The method starts with analyzing the current state of a work situation (task model 1, including (a) people, organization and roles; (b) objects, locations and history of the situation; and (c) work processes, tasks, and actions). Subsequently, a new work situation is specified (Task model 2), based on requirements of the client, on technological possibilities and constraints, and on problems and needs derived from the current situation. The next step in detailing the design is the specification of the UVM, where different aspects need to be considered:

1. the functionality of the technology to be developed;
2. the dialogue (interaction language) for the user to communicate with the technology; and
3. the representations of relevant system information to the user.

In the total method there is considerable evaluation and usability testing going on in parallel with the above indicated steps, for which various techniques have to be applied. Moreover, the whole method is highly iterative, since each following step may result in a need to reconsider any of the previous ones, and the ongoing evaluation activities will provide feedback from which design decisions have to be reconsidered.

The final result of the UVM design phase is a consistent set of specifications on functionality, dialogue, and presentation, that has passed the evaluation tests, and is specified to such an extend that there is no ambiguity in relevant design decisions (as far as the user and client are concerned) during implementation.

Designing the UVM

Designing the UVM is an integrated set of activities, though each of the three above mentioned aspects needs its own techniques and its own viewpoints. Designing the functionality requires the point of view of the future work situation (Task model 2), as well as tools that are based on a conceptual framework that provides the relevant base for specifying the semantics of the technology for the human user. Specifying the representation needs an "artistic" point of view, in that the representation should be developed for the purpose of inducing in the user's mind the intended understanding of information transferred from the machine: feedback and results of user actions, system state and system behavior, relevant aspects of past interactions. Artistic craft and insight in human communication phenomena (e.g. semiotics) have to be combined in order to develop optimal representations for well defined groups of human users.

Specifying the dialogue means the development of a "language", consisting of a syntax and of user actions and observable system events that allow the communication between the human and the machine, or even between various human partners via the machine. Techniques in this scope have to be build on insight in human capabilities of communication (psycho-linguistic insights). The dialogue needs to allow communication regarding the task delegation to the system and regarding information to be transferred between user and system. Also, the dialogue needs to be as transparent as possible, in the sense that the interaction language as such should not contribute too much to the user's task load or require extra attention or time. In addition, the learnability of the dialogue language has to be related to the intended frequency and duration of use of the system.

Main Problems of Interaction Design Techniques

Designing the dialogue aspect of the UVM (from now on referred as 'interaction design') has been a topic of interest for developers of systematic design methods like ETIT, GOMS, TAG, CLG and ETAG.

As part of our teaching interactive systems design to both university students and to experienced practitioners of software engineering [14], we observed the application of formal design methods in actual design for industry and public administration systems. Among our field experiences are case studies of systematic dialog design of a library system, various processes for the Amsterdam social security system, a photo copy shop, a communication system for a taxi cab company, and a security key system for a complex building.

In these examples we observed the attempts to apply TAG, ETAG, CLG, and various hybrids of the obvious formal specification languages.

In [5], four requirements for interaction design techniques are mentioned:

1. it should be based on both the point of view of the user, and provide a complete and accurate representation of the design (completeness);
2. it should have a wide applicability, which means the technique should be applicable in different situations (applicability);
3. analysis and predictions based on the models made by the interaction design technique should be valid (validity);
4. an interaction design technique itself should fulfill the requirements of being functional, easy to use, and easy to learn and remember (usability).

Completeness
Most of the currently available interaction design techniques are not able to completely describe the whole interaction between the user and the system. Most of the time only one aspect of the whole interaction process is described, like the mapping from user tasks to system tasks or from system tasks to the physical actions performed by the user.

Applicability
Due to the complexity of some interaction techniques, they appear to be only applicable in a few limited situations. During several years of experience in actual design of interactive systems, both in industry and public administration, techniques like CLG and ETAG showed to be very hard to use, due to its complexity. These techniques use complex formalisms, without providing a clear heuristic for structuring design decisions. They also provide too many specialized concepts to make them applicable in different situations.

Validity
De Haan also mentions that most of the validation studies for the existing interaction techniques used very simple user interfaces and where almost always performed under the supervision of the developer of the technique and not taken outside the research laboratory. This means that it is not certain that techniques are also valid in other situations than the situations used during the tests.

Usability
Currently, there is no overall design method, which includes task analysis, functionality design, presentation design, evaluation, prototyping and implementation, which integrates any of the interaction design techniques. De Haan indicates that "Formal modeling techniques are usable in design to the extend that they can be integrated with other techniques used by designers." This means that an interaction design technique, to be useable, should be integrated or should be able to be integrated in an overall design method. Another problem concerning usability is that some techniques require a substantial time to learn and use. De Haan suggests to create a designer's workbench built around a particular formal modeling technique. This can only solve the usability problem if the underlying interaction design technique is well defined and not too complex.

To solve the main problems mentioned a new interaction design technique is necessary. As the currently existing techniques appear to work in some situations, the new technique should not be developed out of the blue, but should be based on both actual experiences in design situations of considerable complexity, and on the relevant concepts and relations within the existing techniques. To this end these techniques need to be compared objectively, which will be described in the next section.

COMPARISON METHOD

This section describes a way to compare the existing interaction design techniques objectively. A method to describe a technique in an objective way is explained first. Then the way the comparison takes place is described. The techniques that have been compared are ETIT, GOMS, TAG, CLG, and ETAG. This comparison method can also be used to compare other design techniques, like data-modeling techniques, object-oriented techniques, etc. [6], [12].

ETIT	GOMS	TAG	CLG	ETAG
External Term	Goal	Simple Task	Entity[1]	Basic Task
External Task	Operator	Action[1]	Task[2]	Object
Internal Term	Method	Feature	Procedure[1]	Event
Internal Task	Selection Rule	Value	Method[1]	Place
Mapping Rule		Rule[2]	Operation[2]	State
		Rule Schema	Command	Production Rule
			Primitive Action	Keystroke

Table 1. Main concepts of the compared interaction design techniques.

[1] This concept consists on different levels of a CLG description. Each lower level is a further specification of a higher level. Example: In CLG, Task Entities (on the task level) are mapped onto System Entities (on the semantic level).

[2] The specialization of this concept has not been included in the comparison, because it is only a special case of the main concept.

Meta Modeling

The research method applied in this study is based on meta-modeling, a technique often applied in the area of Method Engineering [2]. Method Engineering is defined as the engineering discipline to design, construct, and adapt methods, techniques and tools for the development of information systems. Typical research studies include categories as:

- design of Computer Aided Software Engineering (CASE) tools;
- comparison of methods and tools;
- situational Methods, i.e. methods configured to the system at hand;
- expressiveness of specification formalism.

Models of techniques or tools, so-called meta-models, depict the concepts plus the interrelationships of the underlying methodological semantics. Possible meta-models are meta-data-models, which describe the concepts and their interrelationships of a technique, and meta-process-models, which describe the processes to apply a technique. In this study meta-data-models of the interaction design techniques were created to aid the formalization and to fill the comparison tables. Figure 1 shows the meta-data-model of GOMS.

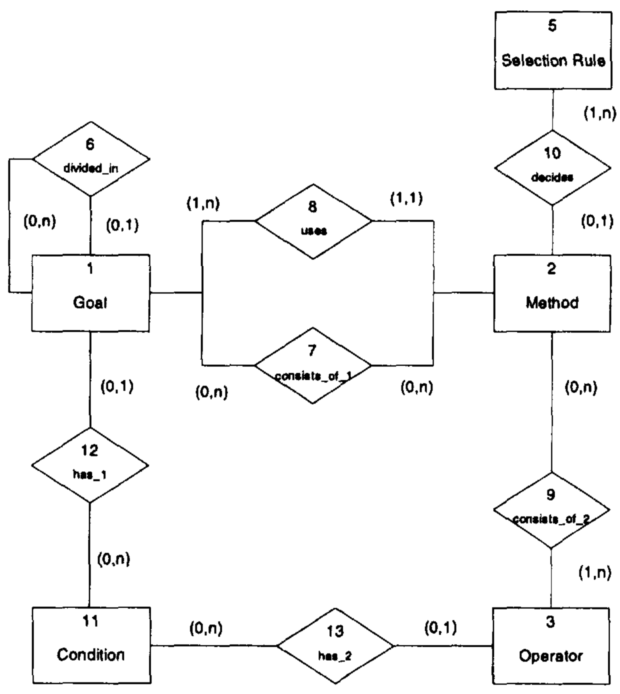

Figure 1. Meta-data-model of GOMS.

Based on the results of a comparison, the conceptual basis for a "super technique" can be created. A conceptual basis is a description (consisting of a meta-model and a formalization) of the concepts and relations between those concepts of a technique. A super technique is a technique created from the concepts and relations of other techniques. In this case, a super technique is created from the concepts and their interrelationships of the five interaction design techniques.

Comparing the Concepts

The first step in comparing the concepts is to list the main concepts of each technique. A main concept is a concept that cannot be removed from a technique without making the technique useless. Table 1 lists the main concepts of the compared techniques.

The second step is to find the main concepts for the super technique. This is done by grouping all the similar concepts of each technique into one group. Similar concepts can be found by examining the definitions of the concepts for similarities.

Example

ETIT's concept of External Task, GOMS's concept of Goal (on a certain level of description) and CLG's concept of Task are very similar. They all talk about tasks a user wishes to perform, which exist in the world of the user. So these concepts are grouped together.

Important is to notice that comparing the concepts means comparing the meaning of the concepts and not the implementation. This is because the implementation of some concepts may not always be compatible, while the concepts are used in the same way or for the same reason, which is expressed in their meaning. Finding the meaning of a concept depends mainly on experience in applying the technique in actual design situations.

Example

The concept of Rule from TAG and the concept of Procedure from CLG are implemented in two different ways. The Rules from TAG are used declarative whereas the Procedures from CLG are used as procedural knowledge. But both concepts are used for describing a mapping from the concept Task to another concept.

Next the concepts that are not used in any group are examined and the relevancy of each concept is determined. Only those concepts which appear to have some logical relation (based on knowledge of interaction) with one of the just formed groups are to be used in the super technique and each of these concepts will form a group of its own.

Example

The Method concept from CLG is defined as "the way in which tasks are associated with procedures" [7]. This concept seems to be logically related to the groups which describe external and internal tasks and the group which include mapping rules (ETIT), hierarchy and methods (GOMS), procedures (CLG), rules (TAG) and production rules (ETAG). There is a logical relation between those two groups and the method concept, as most systems offer a user different ways of performing the same task (for example: copying a file can be done by using the clipboard or by drag-and-drop). The method concept describes these choices by relating the user task to different procedures. This means that the Method concept will form a group of its own in the super technique.

Next, an appropriate name and definition for each group must be determined. These names will also be the names for the concepts in the super technique. Table 2 shows the result of the comparison of the five interaction design techniques.

Comparing the Relations

The purpose of comparing the relations is to find those associations and axioms that are to be used in the super technique. For the comparison of the relations between the concepts of each interaction design technique, the formalization of these relations are used. Formalization means describing the relations between the concepts and describing the constraints on these relations in a formal way, in this case using predicate logic's. Before an actual comparison of the relations can be made, each formal rule has to be translated to use the concepts of the super technique. This means substituting the techniques specific concept with it super concept in the concept list, associations and axioms. During this translation process, only those concepts, associations and axioms are translated which describe relations between two or more super concepts.

Example

This example is taken from the translation of GOMS. The translation of the concept list is based on Table 2. The translation of this list is:

Goal (G) = External Unit Task (EUT)
Goal (G) = Internal Basic Task (IBT)
Method (M) = Task Procedure (TP)
Method (M) = Action Procedure (AP)
Operator (O) = Action (A)

The GOMS association between a method and operator (consists_of_2) will be translated, including an axiom belonging to this association. The formal association is defined as:

predicate consists_of_2 **over** M x O
consists_of_2 (x, y) means that Method x consists of Operator y

The super concept for method (M) can be Task Procedure (TP) and Action Procedure (AP) and the super concept for Operator (O) is Action (A). But as only an Action Procedure can consist of Actions, the translation of this association will be:

predicate consists_of_2 **over** AP x A
consists_of_2 (x, y) means that Action Procedure x consists of Action y

The translation of one of the axioms belonging to this association is straightforward:

A Method consists of zero or more Operators:

$$\forall_{x \in M} [\, \exists_{y \in O} [\text{ consists\_of\_2 } (x, y)]\,] \oplus [\, \neg\exists_{y \in O} [\text{ consists\_of\_2 } (x, y)]\,]$$

which will translate to:

An Action Procedure consists of zero or more Actions:

$$\forall_{x \in AP} [\, \exists_{y \in A} [\text{ consists\_of\_2 } (x, y)]\,] \oplus [\, \neg\exists_{y \in A} [\text{ consists\_of\_2 } (x, y)]\,]$$

The substitution used in the translation is not always an one on one substitution, as one super concept can consist of more than one concept in an existing technique and vice versa. In these cases a careful examination of those concepts is necessary to make a correct translation. In the case that a concept of an existing technique translates to more concepts in the super technique, any associations using this concept may sometimes be split into more associations and axioms.

SUPER	ETIT	GOMS	TAG	CLG	ETAG
External Unit Task	External Task	Goal	-	Task	-
Internal Basic Task	Internal Task	Goal	Simple Task	System Operation	Basic Task
External Object	-	-	-	Task Entity	User Object
Internal Object	-	-	-	System Entity	System Object
Action	-	Operator	Action	Primitive Action	Keystroke
Task Procedure	Mapping Rule	Method	-	Procedure	-
Action Procedure	-	Method	Rule	Procedure	Set of Production Rules
Task Method	-	-	-	Method	-
Action Method	-	-	-	Method	-

Table 2. Grouping of the main concepts and the concepts to be used in the super technique.

Example
In the association *uses* between the Goal and a Method (a Goal uses a Method), the Goal concept translates to both the super concepts External Unit Task and Internal Basic Task. The Method concept translates to both Task Procedures and Action Procedures. There are only two logical combinations, namely External Unit Tasks using Task Procedures and Internal Basic Tasks using Action Procedures. The other combinations are not possible, as an External Unit Task cannot directly use an Action Procedure as the Internal Basic Task should always be in between and Internal Basic Tasks can not use Task Procedures, as an Internal Basic Task is already the smallest task a system offers and thereby cannot contain other Internal Basic Tasks. As a result, the association *uses* has to be split into two associations, one stating that External Unit Tasks use Task Procedures and one stating that Internal Basic Tasks use Action Procedures. Also the axioms belonging to this association have to be split.

There are two possible approaches to actual comparing the translated associations, an inductive approach and a deductive approach.

The inductive approach uses the translated associations and axioms of the existing techniques to induce the general associations and axioms of the super technique. This is done by grouping the translated associations and axioms together. This grouping is based on the concepts from the super technique used. All associations and axioms that describe relations between the same concepts are grouped together. Some associations and axioms can be used in more than one group. After this grouping, each group is examined separately. The associations and axioms of the super technique are then derived from those in the group.

In the deductive approach the associations and axioms of the super technique are created using knowledge of interaction. The translated associations and axioms of the existing techniques are used to check whether the associations and axioms of the super technique are correct or not. If there is a discrepancy a decision has to be made, either the associations and axioms of the super technique have to be changed according to those of the existing techniques, or the associations and axioms of the existing techniques are too specialized, in which case the associations and axioms of the super technique are not changed.

In both approaches it is highly likely that some associations and axioms of the existing techniques will contradict each other, or that one axiom is stricter than another. In these cases a decision must be made about which associations and which axioms are to be used. This decision is based on which axioms and associations describe best the interaction between the user and the system.

From experience in comparing the interaction techniques, the deductive approach appears to be better applicable. The main reason for this is that, although most of the existing interaction techniques do describe the same concepts, they differ greatly in the relations between the concepts. This means that deriving the associations and axioms for the super technique from those of the existing techniques will be difficult due to many contradicting associations and axioms.

After the relations have been determined the conceptual basis of the super technique can be created.

Comparing the Purposes
Every technique has been created with a purpose in mind. To compare the purposes of each technique, they first have to be translated to use the concepts from the super technique.

Example
The purpose of ETIT is *to describe the mapping between external tasks in the world of the user and internal tasks of the system*. The super concept for an external task is the External Unit Task, the super concept for the internal task is the Internal Basic Task and the super concept for mapping is called a Task Procedure. So the purpose of ETIT in super concepts is *to describe the task procedure between the external unit tasks and the internal basic tasks.*

After this translation the purposes of the techniques can be compared more objectively. Comparing the purposes of the five techniques shows that all five techniques describe the interaction between a user and a system, by describing the procedures from external user tasks to internal basic tasks and/or from internal basic tasks to actions. Some techniques also describe the mapping from external objects to internal objects.

Although all five techniques have the same general purpose, some techniques also have more specialized purposes. TAG for example also has the purpose of determining the complexity of a design by applying simple metrics. As the purpose of this study is to find the conceptual *basis* for interaction design, these specialized purposes are less important. For finding the conceptual basis, the general purpose of each technique is important.

Comparing the Places in the Design Method
The comparison of the place of the various interaction design techniques in our design method consists of two

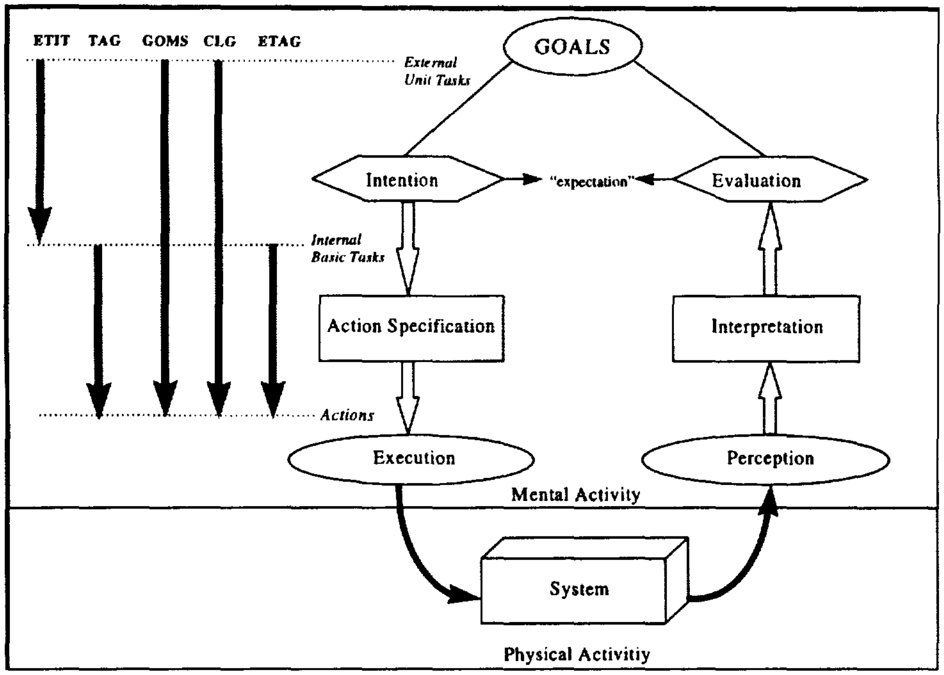

Figure 2. Comparing the places in Norman's Interaction Process Model.

parts. The first part is to look at the phase in which the technique can be used in the design method (as described in the first section). The comparison of the five techniques shows that most of them are used in the interaction design phase. A technique like GOMS can also be used in the task analysis phase, where it can describe the task structure.

The second part looks at the place and reach of each technique within a general interaction model. This comparison is necessary to show which part of the whole interaction process is covered by each technique. The general interaction model used is from Norman [9]. Figure 2 shows the general interaction model and the place and reach of each of the existing techniques in it.

This comparison shows that all the existing interaction design techniques, model only (parts of) the task delegation of the interaction process (from goals to execution). None of the five techniques take the feedback from the system to the user into account. It also shows that a technique like ETIT only describes the mapping from goals to intention, where techniques like TAG and ETAG only describe the mapping from intention to action. A technique like CLG describes the whole mapping from goals to actions.

The super technique should be able to model the whole interaction process, from goals to execution as well as the feedback from the system to the user, taking into account the perception, interpretation and evaluation from the user.

RESULT

Conceptual Basis for Super Technique

The comparison of the five interaction design techniques results in a conceptual basis for a super technique and shows which parts of the interaction process are covered by this conceptual basis. This conceptual basis can be described in the same objective way as the five existing interaction design techniques have been described. Figure 3 shows the conceptual basis of the super technique in a meta-data-model. This conceptual basis is the smallest set of concepts and relations that has to be modeled during the interaction design phase. As the comparison is also based on experience in actual design situations, this smallest set of concepts and relations is the most relevant set for every design situation.

The conceptual basis has to be evaluated for use in actual design situations. For this reason several design teams from industry and university students are using this conceptual basis in user interface design courses. The results from this evaluation have to be incorporated in the conceptual basis.

The conceptual basis and the description of the other five techniques are considered useful for situational method engineering. According to Brinkkemper [2] a situational method is "an information systems development method tuned to the situation of the project at hand." This means in this case that an interaction design technique can be changed to include other concepts and relations depending on the actual design situation. These other concepts and

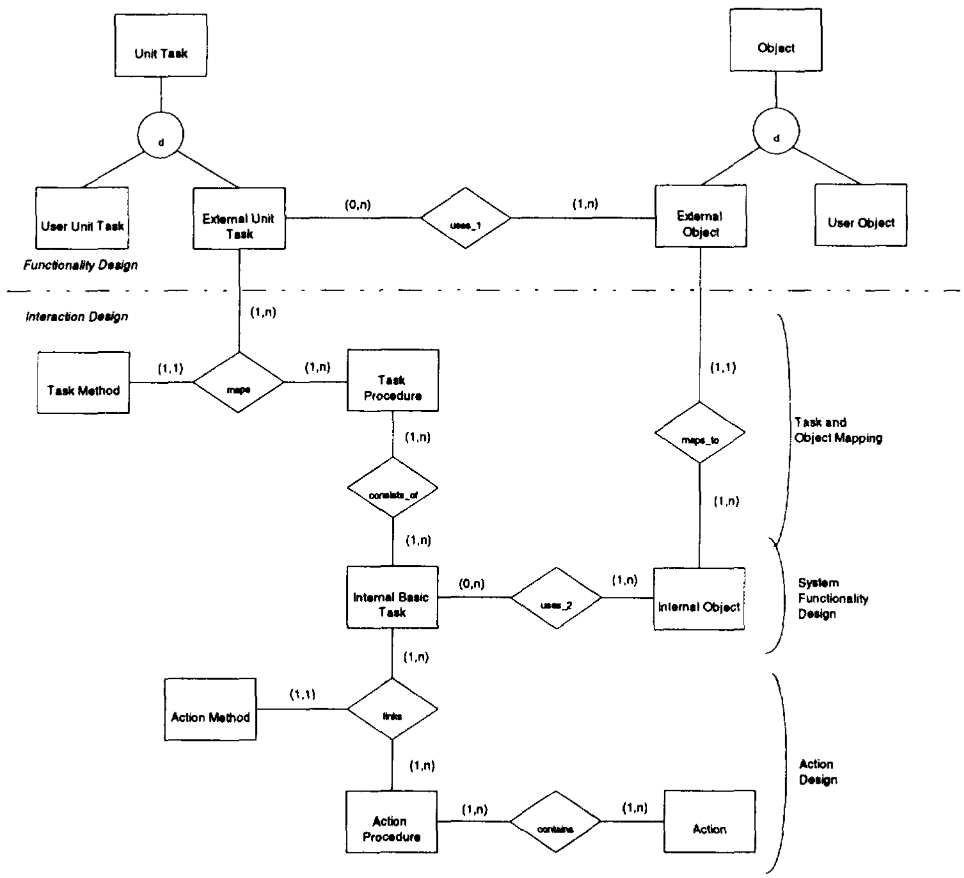

Figure 3. Meta-data-model of the Super Technique.

relations can be taken from one (or more) of the existing techniques, and from other system development methods. Because a formal description of these techniques already exists, creating a new conceptual basis for the *situated* technique is easy. This would mean following the steps as described in the previous section to compare and integrate the new concepts and relations with the existing conceptual basis of the super technique.

Example

In an actual design situation where a command language interface has to be developed it can be necessary to describe the command language itself in more detail. In this case the concepts command and argument from CLG can be used and integrated in the design technique by applying the steps described in the previous section. This new, situated, technique is now tuned to the actual design situation and therefor more complete, applicable, valid and useable.

Further Research

The conceptual basis of the super technique is not enough for a complete description of the interaction between the user and the system. As already mentioned, also the feedback from the system to the user is important. Also the design process is a major topic for further research. In this research, method engineering can also be applied.

Another notion is that the five existing interaction design techniques are all used for describing the interaction between one user and a system, whereas present day interactive systems cover several different users, with different roles, in an organization which uses complex information technology in a structure of functionality that is intended to support organizational work structures.

So far, there are no formal links defined between the interaction part of the UVM and the presentation interface (in which also the main part of the feedback lies), which is another main part of the UVM.

All these notions have to be addressed, in order to create a complete UVM design technique, which can be integrated in the overall design method. To this end, some thesis- and Ph.D.-students at the University of Twente and the Free University of Amsterdam are addressing these aspects. As the topics for further research have aspects from several disciplines, these students come from several disciplines, like technical engineering, software engineering, psychology and communication science.

CONCLUSIONS

This paper shows that there is a conceptual basis for interaction design, which can be found as the result of both experiences in actual application of interaction techniques in real life design and objectively comparing the existing interaction design techniques.

By using this conceptual basis and the result of further, multi-disciplinary, research, a complete interaction design technique can be created, which can also be integrated in an overall interaction design method. This method will be applicable in many situations, as it allows for situational method engineering. This also makes the method more valid in different situations, which will be tested in several design situations. Because the complete interaction design technique is described in an objective and formal way, support tools can be build, which will make the technique more useable in actual design.

REFERENCES

1. Attema, J., van der Veer, G.C. Design of the currency exchange interface: Task action grammar used to check consistency. Zeitschrift für Psychologie, 200 (1992), 121-134

2. Brinkkemper, S. Method Engineering: engineering of information systems development methods and tools. Information and Software Technology, 38 (1996), 275-280

3. Brinkkemper, S. Methodology of Information System Design (Lecture notes). Department of Computer Science, University of Twente, The Netherlands, 1995

4. Card, S.K., Moran, T.P., and Newell, A. The psychology of Human-Computer interaction. Lawrence Erlbaum Associates, London, 1983

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

6. Hong, S., Brinkkemper, S. Object-Oriented Method Components for Situation-Specific IS Development. In: Ram, S. and Jarke, M. (eds). Proceedings of the Firth Annual Workshop on Information Technologies and Systems (WITS'95). Aacherner Informatik Berichte, 95-15 (December 1995), 164-173

7. Moran, T.P. The Command Language Grammar: a representation for the user interface of interactive computer systems. International Journal of Man-Machine Studies, 15 (1981), 3-50

8. Moran, T.P. Getting into the system: External-Internal task mapping analysis. In: Janda, A. (ed). Proceedings CHI'83. ACM Press, New York, 45-49

9. Norman, D.A. Cognitive Engineering. In: Hutchins, E.L., Hollan, J.D., and Norman, D.A. Direct Manipulation Interfaces. Human Computer Interaction, 1 (1985), 311-338

10. Payne, S.J. Task-Action Grammars. In: Shackel, B. (ed). Human-Computer Interaction - INTERACT '84. Elsevier, Amsterdam, 527-532

11. Payne, S.J., and Green, T.R.G. Task-Action Grammar: A model of the mental representation of task languages. Human-Computer Interaction, 2 (1986), 93-133

12. Spit, M., Brinkkemper, S. and Lieberher, K. Integrating Adaptive Programming into Existing Object-Oriented Analysis & Design Methods: Do It Yourself Adaptiveness. In: Patel, D., and Sun, Y. (eds). Proceedings of the 3rd International Conference on Object Oriented Information Systems (OOIS'96). Springer Verlag, London, December (1996), 57-75

13. Tauber, M.J. ETAG: Extended Task Action Grammar - a language for the description of the user's task language. In: Diaper, D., Gilmore, D., Cockton, G., and Shackel, B. (eds). Proceedings INTERACT '90. Elsevier, Amsterdam, 163-168

14. van der Veer, G.C., van Vliet, J.C., and Lenting, B.F. Designing Complex Systems - a Structured Activity. In: Olson, G.M., and Schuon, S. (eds) Proceedings of DIS '95. ACM Press, New York, 207-217

15. van der Veer, G.C., Lenting, B.F., and Bergevoet B.A.J. Groupware Task Analysis: Modeling Complexity. Acta Psychologica, 91 (1996), 297-322

16. de Vries, G., Johnson, G.I. Het gebruik van GOMS en de 'limited information task' bij de evaluatie van een autoradio. Tijdschrift voor Ergonomie, 6 (1992), 2-9