BEHAVIOUR & INFORMATION TECHNOLOGY, 1998, VOL. 17, NO. 6, 338 – 360

Representation still matters: cognitive engineering and user interface design

CHRIS STARY† and MARK F. PESCHL‡

†University of Linz, Department of Business Information Systems, Communications Engineering, Freistädterstraße 315, A-4040 Linz, Austria; e-mail: stary@ce.uni-linz.ac.at

‡University of Vienna, Department for Philosophy of Science, Sensengasse 8/10, A-1090 Vienna, Austria; e-mail: Franz-Markus.Peschl@univie.ac.at

Abstract. With the increased utilization of cognitive models for designing user interfaces several disciplines started to contribute to acquiring and representing knowledge about users, artifacts, and tasks. Although a wealth of studies already exists on modeling mental processes, and although the goals of cognitive engineering have become quite clear over the last decade, essential epistemological and methodological issues in the context of developing user interfaces have remained untouched. However, recent challenging tasks, namely designing information spaces for distributed user communities, have led to a revival of well known problems concerning the representation of knowledge and related issues, such as abstraction, navigation through information spaces, and visualization of abstract knowledge. All of these issues are associated with mental processes and thus, might become part of cognitive models.

In this paper we reveal epistemological and methodological assumptions in the field of cognitive modeling as well as their implications for user interface design. It turns out that in order to achieve the goal of developing human-oriented (in contrast to technology-driven) human-computer interfaces developers have to develop knowledge of the structure and the representational dynamics of the cognitive systems which are interacting with the computer. We show that in a first step it is necessary to study and investigate the different levels and forms of representation that are involved in the interaction processes between computers and human cognitive systems. We propose a hybrid user modeling approach as part of the task-based development procedure in TADEUS (Task Analysis/Design/End User Systems). The hybrid approach does not only enable the representation of functional roles end users have to perform, but also *how* end users perform these roles, i.e. the representation and reflection, if not prediction of their behavior. This way, holistic system development that equally takes into account the organizational requirements *and* the end user reality at work places is facilitated.

1. Introduction

Starting with Rasmussen (1986) cognitive engineering has been established as an interdisciplinary field with knowledge representation as one of its core issues, in particular in the context of human-computer interaction, e.g. Woods *et al.* (1988, p. 3). At that time a strong need for human-oriented design had been expressed, since in the field of operator workplace design novel design solutions were required. It has been recognized that the introduction of an interactive computer system often changes the work environment and the cognitive demands placed on employees. Several findings help to detail the changes caused by the use of computer systems:

- Although the physical work load can be decreased at most of the workplaces through the utilization of computer systems, in some cases the mental workload might increase for particular users, due to inherent problems in information systems, such as according to Oliver (1995):
  - disorientation
  - navigation inefficiency, and
  - cognitive overload.
- Although computers enhance the potential in assistance of work, that extension might require revisiting the allocation of functions to humans and machines for human – computer interaction (e.g. Kantowitz *et al.* 1987).

As a consequence, the potentials of computer systems have to be tuned to the context of their utilization (Hollnagl *et al.* 1983). In order to achieve the context-sensitive utilization of technological artifacts designers need mechanisms to represent these contexts, i.e. on one hand, an adequate cognitive language of description (Rasmussen 1986), and on the other hand, a representation of the end user tasks and their organization (Johnson, 1992). This task should be accomplished through **cognitive engineering**. It has been established as a traditional applied cognitive science. The requirement for an analysis and specification of semantic and pragmatic aspects has been stressed from that time on up to recently, e.g., Roth *et al.* (1987).

In the following we will term the specification of the semantic and pragmatic aspects of a domain **representation**. It might describe mental processes as well as the results of those processes (see also Figure 1, where two different externalized models of neural representations are given). We term representations of mental processes as well as their results **cognitive models**. Such models help individuals to understand 'what knowledge of the world is needed and how this knowledge can be used to achieve effective performance' (Woods *et al*. 1988, p.34). In order to achieve effective performance, a translation of work tasks into the functions of the computer system has to be performed by the individuals interacting with the computer system, termed end users (Moran 1984). The activity of mapping the mental models to representations will be termed **cognitive modeling**. Overall benefits of cognitive modeling are expected in terms of

- predictability of human behavior in the course of human-computer interaction,
- avoidance of errors in the course of task accomplishment, and
- improved usability of interfaces, based on the represented knowledge about mental processes.

The construction of cognitive models requires to take into account several activities and steps, since human perception and learning are assumed to occur along a certain path of actions, involving several key elements of human cognition. Traditional cognitivists, such as Olson *et al*. (1990), assume that the non-observable processes of cognition are based on the interpretation of perceived physical activities, and on intentional specification of actions that are actually executed. Goals and expectations form a background against which the interpretation and specification of actions occur. Unfortunately, only few components and transitions, such as the memory performance checking intentions, have been investigated empirically (Norman 1986).

According to Norman (1983), setting up a model of cognition involves several perspectives, namely the perspective of the developer and the ones of the end users that are involved in the development. As a consequence, several types of cognitive models of a target system can be distinguished:

(i) the *mental* models of the end users of the target system: these representations are those internal models that can be specified explicitly through one or more conceptual models by means of knowledge elicitation methods;
(ii) the *conceptual* model of the target system: this is the external model that has to be developed by the designer by using elicited knowledge from several end users;
(iii) the *developer's conceptual* model of the end *user's mental* model of the target system: this is the model of the developer that he/she has in mind in the course of knowledge elicitation and the specification of the conceptual model of the target system.

In the traditional understanding of cognitive modeling all of the listed models are involved. The process for the construction of knowledge is driven by the demand to improve the understanding of individuals with respect to features of interactive systems. As Figure 1 shows it is assumed that developing this process is a two-step procedure. The goals for knowledge construction are given through the computer system itself, since it reflects the outcome of the design and implementation process. The first model reflects the perspective of the developer: The Conceptual (Representational) Model of the (target) Computer System (listed under (ii)) represents what the software developers are expecting or have expected as outcome of their work. It contains the results of the transformation of end-user and task requirements to technical functionality. This model is traditionally described through some kind of notation or language, such as semantic networks or constructs in first order logic, respectively. It contains a more or less analog representation of the goals implemented in the computer system at hand. It is also that model users apply in order to interpret the technical functionality of an artifact. This model comprises more or less knowledge about how to use system functions, rather than conveying the rationale behind, namely:

- why they have been developed this way,
- why they have to be used the way they have to be used, and
- when to use them.

In order to succeed in interactive task accomplishment users have to connect their mental models to the cognitive model that explicitly states the goals of developers that have been implemented (user perspective). Since the users' goals might not correspond to the ones from the designer's cognitive model, in traditional

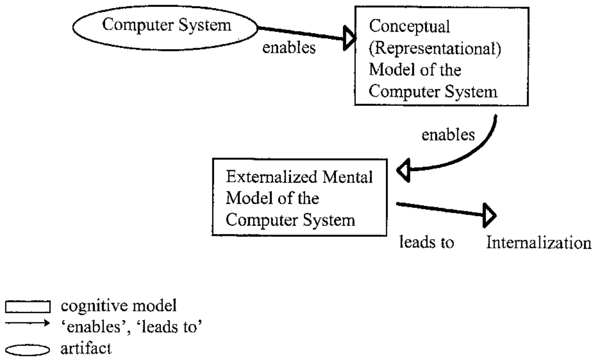

Figure 1. Externalized cognitive models with respect to interactive computer systems.

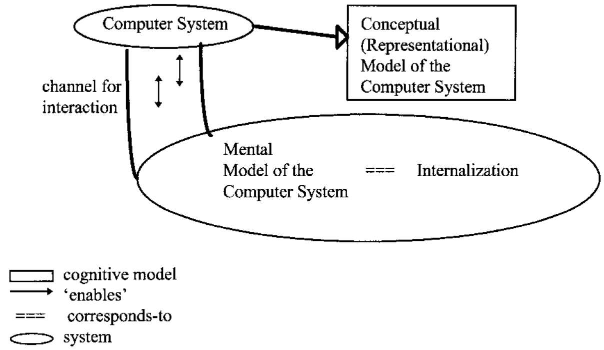

Figure 2. Results from the constructivist shift in cognitive modeling with respect to HCI.

cognitive modeling user goals have to be related to the designer's goals explicitly before they can be internalized by each of the end users. The second cognitive model (Externalized Mental Model of Computer System in Figure 1) listed under (iii), is created. Finally, internal mental activities are assumed to occur, based on the contents of the second cognitive model, leading to the internal model as listed under (i).

Mental abilities are considered to be part of a 'human information processor' and its decomposition of specific tasks in terms of goals and other components. For instance, in GOMS (Card *et al.* 1983, Olsen *et al.* 1990) goals, operators, methods and selection rules are distinguished to model human – computer interaction. Hence, the semantic and pragmatic aspects of the relationship of humans to their environment are specified through goals of agents and the constraints in the environment of the agent. Human – computer interaction is then understood to be driven by the execution of actions to achieve goals.

In questioning the understanding of human – computer interaction as an exchange of data or information based on the goal-driven execution of actions we will apply a systemic perspective of cognitive engineering, namely rethinking human – computer interaction as a mutually modulation of systems. Interaction is considered to be a mutual modulation influencing the respective (knowledge/representation) system so that a certain task can be achieved. Intentions become behaviors and activities on the human side, computation results are prepared as output on the machine side. What is referred to as human – computer interaction, is a multi-level, multi-channel, and multi-modal interaction process between two representation systems. Figure 2 illustrates the result from that shift, namely from explicitly goal-driven to representational systems matching behavior parameters along task accomplishment instead of matching developer and user goals explicitly. In applying this perspective human – computer interaction becomes the mutual adaptation of two systems involved: the computer system and the cognitive system of the user. In modeling the user this way, cognitive modeling will not have to address an externalized and internal mental model of end users any more. The transition between explicit and tacit mental representations will be resolved by unifying both perspectives through sub-symbolic representations and behavior patterns. Cognitive modeling partially becomes behavioristic—this paradigm has been termed neo-behavioristic (Stary *et al.* 1995).

However, the utilized representation scheme provides an interface to the first cognitive model identified in Figure 1, namely the conceptual model of the computer system developed by the designer. Since the representation of the conceptual model will be based on the end user tasks, it has to be considered as a representation of commonly agreed objectives for interactive task accomplishment. However, each end user is modeled through a single cognitive system with behavior based on direct, steady and adaptive modulation. This way, not only tasks might be perceived individually, but also the organization of data (required for task accomplishment), and the navigation through features for interaction at the user interface.

As a consequence, developing a human – computer interface always requires the finding and construction of a representational and computational structure that meets the constraints given by the cognitive dynamics of the potential users. Capturing the cognitive dynamics is in particular relevant due to the

- *incompleteness* and
- *instability*, due to the lack of accuracy, of mental models

with respect to the behavior of computer systems. As Norman (1983) has investigated, 'most people's understanding of the devices they interact with is surprisingly meager, imprecisely specified, and full of inconsistencies, gaps and idosyncratic quirks' (p. 8). Since constructivist modulation allows modeling steady changes from the individual's perspective, both the incompleteness and the instability of the externalized and the internal mental model can be captured accordingly. As computer systems/programs as well as input/output devices can be developed—within certain technical and economic limits—more or less arbitrarily, the 'cognitive constraints' have to be considered one of the starting points for the development of interactive systems.

Although cognitive systems are able to adapt to new situations over time, such as standard features at the user interface, the goal of human – computer interface development is to provide human-oriented access features to the structures being processed by the computer program, e.g. Johnson (1992). Human-oriented user interface development requires at least a rough understanding of the cognitive processes involved in these interactions (Stacy 1995). In order to enable an epistemologically and empirically sound integration of cognitive modeling based on constructivist understanding of human – computer interaction into techniques and processes for user interface development, in (Peschl *et al.* 1998) we suggest revealing empirical knowledge about cognitive systems, their structure, and their dynamics. Based on these findings theories can be constructed that describe and explain the observed (cognitive) phenomena when accomplishing tasks. These steps should form the basis for the development of (computational) cognitive models for particular users, as traditionally developed in (cognitive) psychology and cognitive sciences (Andersen 1990, Eckardt 1993, Johnson-Laird 1993, Oshersen *et al.* 1990, Posner 1989 etc.). Then, based on these models, human – computer interfaces might be constructed or adapted accordingly. This kind of interface should fit the cognitive dynamics of humans:

> *Human – computer interfaces should stimulate, modulate, and trigger the cognitive dynamics of the user(s) in such a way that the intended task(s) can be accomplished with minimal (cognitive) effort.*

In order to implement this overall goal several principles for designing and evaluating user interfaces have been identified, such as the suitability for tasks in cognitive ergonomics, e.g., Ravden *et al.* (1989) or task appropriateness in software ergonomics, e.g., Stary (1996b). Although these principles have been explained and based on more or less empirical findings, most of them still lack operational definitions, see e.g., for adaptation (Stary *et al.* 1997b). This deficiency can be explained through the lack of epistemologically sound development of concepts in the field of human – computer interaction, starting with empirical results and subsequent theory finding for modeling human individuals.

The problem of knowledge representation plays a crucial role in this context, since mental models require some mechanism of representation to be utilized for user interface development. Consequently, we are going to examine the different levels and forms of knowledge (representation) involved in the interaction between humans and computers. It will become apparent in the sections to come that developers can not only gain clarity and learn a lot from these considerations, but they are also enabled to explain shortcomings and failures of solutions as soon as cognitive modeling becomes part of the user-interface development process.

In section 2 conceptual modeling from the cognitive engineering perspective is contrasted with the constructivist understanding of human – computer interaction. The introduced framework is used in section 3 to model human- and task-centered interaction. First, shortcomings of symbolic representation systems, in particular for modeling cognitive processes from the neuroscientific perspective, are identified. Then, a proposal to overcome some of the existing limitations is introduced. Section 4 concludes the paper through summarizing the achieved results and sketching further research activities.

2. Traditional cognitive engineering versus constructivist understanding of human – computer interaction

First, related work will be discussed (i) with respect to the objectives of this study and the derived requirements for human-oriented modeling in section 2.1, (ii) with respect to existing analyses and reflections of cognitive models as fundamental elements of cognitive engineering activities (section 2.2). In section 2.3, the constructivist framework is introduced and contrasted to the traditional understanding of cognitive modeling. In section 2.4 we will wrap up the pitfalls in knowledge acquisition and representation that are highly related to the traditional understanding of cognitive engineering, but can be removed through the constructivist understanding of human – computer interaction.

2.1. *Human-oriented cognitive modeling*

In several application domains, such as manufacturing, the request for human-oriented cognitive modeling have become more evident: According to Böhle *et al.*

(1988) 'human – computer interaction requires more than Yes/No-judgments of user inputs [referring to the explicit representation of goals], namely the consideration of the empathy and subjective involvement' (p. 239) of workers in the course of their task accomplishment:

- Human knowledge and activities are not guided exclusively through rational behavior. However, conventional knowledge acquisition and representation techniques follow the principle of rationality, assuming some type of ideal user accomplishing tasks in an optimized way. Defining an optimal functional role behavior (without taking into account the social and perceived reality of the individuals) implements a rationalistic tradition of thought rather than a human-oriented modeling approach (Winograd *et al*. 1986).
- Tacit knowledge (Polanyi 1996) and tacit skills (Wood 1986) are essential for successful task accomplishment. In case these skills are not taken into account, humans tend to adapt to procedures given by machines and these skills are eliminated from the work practice. The consequence is a decrease in quality of task accomplishment. However, in particular for complex processes, such as in manufacturing, tacit skills are of decisive importance for the quality of work results and worker satisfaction.

From communication by means of language we know that successful communication requires mutual adjustment of the utterances of the speaker to the listener's state of knowledge. If this adjustment fails misunderstandings and misinterpretations of utterances are likely to occur. In this case, the knowledge of the listener does not help him/her to find out the intended semantics of utterances. In addition, the speaker may even want to take into account what the listener assumes the speaker knows, and so on. This infinite mutual knowledge problem, e.g., Perner *et al*. (1988), might influence communication, if there is no agreement on the common sense understanding at some point. Winograd *et al*. (1986) have termed this common sense understanding 'tradition' partners should share in communication or interaction. It is not clear for humans, how much of this common understanding of the mind actually is innate and how much communicative competence has to be acquired actively, e.g., Fodor (1988). However, cognitive models more or less represent the common sense understanding of a problem or an artifact. The less of the shared tradition they represent, the more likely are misunderstandings between end users and designers.

Another domain of interest with demand for human-oriented cognitive engineering has been defined through the introduction of global information spaces, such as the Internet, for a variety of end users. For instance, Brandt (1997) suggests to introduce constructivist concepts when users should become acquainted with ill-structured information in networked systems. He argues that constructivism shows a way to migrate already existing knowledge at the side of the end users, with novel concepts and inputs more effectively. Although Brandt recognizes the benefits of constructivist cognitive modeling, he does not conclude that techniques for representing knowledge have to take into account individual differences, e.g., Thorndyke *et al*. (1980), in order to achieve the intended strengthening of mental models through constructivist learning.

2.2. *Traditional cognitive engineering*

In the field of human – computer interaction cognitive modeling has rarely been analyzed systematically. In the following the most influential conceptual analyses for our study will be discussed. The selection of the previous analyses has been based on the observation that for the description of the contents of cognitive models, both the externalized mental one and the conceptual one, techniques from knowledge representation or theoretical computer science, such as first order logic or graph theory, have been applied.

Caroll *et al*. (1988) consider mental models to comprise knowledge about how computer systems work. They have also identified several perspectives for cognitive modeling:

- the *task* perspective in terms of goals and sub-goals users may achieve utilizing system features. This perspective is the one that is assumed to be directly supported through conventional techniques from artificial intelligence for acquisition and representation, such as interviewing key users and specifying through first order logic.
- The *interface* perspective comprising the knowledge users need to accomplish a task utilizing the data and interaction styles of a system in a certain sequence. In order to represent interface knowledge from different perspectives, traditional representation schemes of computer science have been extended, e.g. grammars (Johnson 1992).
- the *architecture* perspective comprising knowledge about the storage of data, access functions and internal processes (flow of data and control) of computer systems. Traditionally, techniques from software engineering or database management, such as object hierarchies and entity-relationship diagrams are utilized for representation, respectively.

In order for a mental representation to become an externalized (cognitive) model Caroll *et al*. (1988) have identified the following elements:

(i) the set of possible inputs; and
(ii) the set of possible outputs; as well as
(iii) a highly complex, non-linear function that relates the inputs to outputs (i.e. the representational structure).

Such representations then could be used during learning, problem solving and rationalization. This adopted perspective from natural science is exactly the one addressed by Winograd *et al*. (1986) when they refer to the rationalistic tradition of thought.

Another attempt of analytical reflection on the meaning of mental representations has been performed by Moray (1993). In interpreting mental models as mappings from external systems to human cognition he has identified the following categories of mental models:

- Mental models as imperfect *copies* of external systems. This type of model is created when humans are not capable of grasping the full complexity of external systems, according to Bainbridge (1991).
- Mental models as *logical relationships* among abstract entities and operations. This type of model is utilized to represent the way in which humans reason about syllogistic problems, according to Johnson-Laird (1983).
- Mental models as a *description of reasoning*. This approach is used to specify problem solving of scientific, causal or several logical problems, according to Andersen (1983), Gentner *et al*. (1983), and Newell (1989).

Similar to Caroll *et al*., Moray intended to clarify the possible meanings of cognitive models. He considered it not to be sufficient to identify cognitive models as what users know exactly about software. His contribution has been aimed at considering cognitive models as representations of the knowledge users have about certain aspects about computer systems. In order to bridge the gap between the various perspectives listed above, in contrast to Caroll *et al*. (1988), he proposed the use of a unifying notation, namely the one known from lattice theory. Adopting this notation hierarchy as well as causal effects of knowledge, entries (nodes) in the lattice can be represented. Caroll *et al*. (1988) have proposed some methodological guidelines in the sense of the rationalistic tradition, but have not addressed the representation of knowledge in the course of knowledge acquisition and processing.

Having reviewed related work to our study we are able to conclude: With respect to user interface design representation and acquisition the analysis performed by Caroll *et al*. (1988) and Moray (1993) show that, besides some notational proposals there is still neither a common terminological nor semantic ground where new methodologies based on empirical evidence could arise. As a consequence, in the next section we will study the systems involved and lay ground for a framework that allows a non-rationalistic but human-centered development of cognitive models.

2.3. *The interactive system revisited*

In the following we will detail the constructivist perspective on human – computer interaction with respect to the overall concept of interaction, the involved components and systems: the user, the environment, the computer system. The components and systems are then further discussed in the context of representation in section 3.

The situation designers or analysts face when developing a user-oriented human – computer interface and/or a model of cognition involves several parties and components that can be considered as particular systems:

- the *user* can be characterized as a cognitive system that tries to solve some problem or to accomplish a task more efficiently by making use of a computer system;
- the *computer* can be characterized as a system transforming inputs into outputs in a non-linear manner;
- the *devices and modalities used for interaction* (in/output) provided by the computer system enable the interaction between the cognitive system ‘user’ and the computer system;
- the *user’s sensory and motor systems*: the user’s particular interaction devices allow that external stimuli (such as pixels on a screen) may enter the (neural) representation system (via the visual system, tactile receptors, acoustic system etc.) and that internal representations can be externalized via behavioral actions. These behavioral actions (might) change the environmental structure, e.g., by moving the mouse, hitting a key of the keyboard etc.;
- the *observer*: in order to describe or predict behavior there has to be an observer. The observer has access to both the internal structures of the computer system and the behavioral structures of the user. In most cases this observer is also the

designer of the human – computer interface and/or of the cognitive model.

However, the access to the user's internal representational structures is very limited, as the user is able to externalize a small fraction of knowledge via behavioral actions, such as language. It is the task of the designer to develop an adequate model of the cognitive processes and of the potential user's internal representations (and their dynamics). The constructivist approach facilitates this task by making use of neuroscientific theories and findings from cognitive science.

Investigating the process of human – computer interaction, designers have to be aware that they are not dealing with a one-way interaction, but with systems that try to mutually influence and trigger each other in a more or less beneficial way. As it is the case in almost any interaction between a cognitive system and its environment or other cognitive systems, designers have to deal with a feedback relationship. The ultimate goal of this relationship is to establish more or less stable feedback loop based on 'smooth' interactions and on effectively triggering the respective representation/processing system.

In this type of interaction the user triggers the execution of a certain part of the computer components (program, devices) leading to a certain action within the computer system. In most of the cases the result of this action is somehow externalized and made accessible to the user, e.g. by displaying it on a screen or by acoustic output. In any case, the computer system's output perturbs the user's representational system, which in turn causes some responses by the user. These responses are externalized via the user's motor systems leading to a perturbation of the computer system, and so forth.

The addressed interactions between the user and the computer system do not only involve mechanical processes, but, more importantly, the transfer of knowledge/representations. Consequently, there have to be devices that act as interfaces between these two systems, as their internal representational structures are not necessarily compatible. At a first glance the computer system and the cognitive system do not seem to be compatible. In order to design effective interaction between the user's and the computer system's representational structures, both systems have to be considered to be responsible for creating some kind of compatibility between the internal representations of the participating systems in interaction. Their task is to transform the internal representations into structural changes of the environment, such as activating a muscle that moves a mouse or activating a pixel on a screen, and vice versa. The human and the computer are able to interact with each other via mutually changing the environmental structure/dynamics (key strokes, pointing with the mouse, patterns of pixels on the screen etc.). Similar to communication (natural, spoken, or written language) interaction only becomes possible through the use of the environment as a carrier for mutual stimulation.

2.3.1. *The User as a Cognitive System.* The key player for cognitive modeling and human – computer interaction is the human cognitive system. It is not only interacting with the computer system, but also with the rest of its environment, including other cognitive systems. From observing a cognitive system which is acting (successfully) in its environment, we can conclude that this system has to possess some kind of knowledge about its environment. Otherwise, it would not be possible to behave adequately in the environment. In case of behaving randomly the survival of the system is not very likely.

Cognitive scientists postulate the existence of a representation system that holds some kind of structural information about the environment and on how to survive within the constraints of a given internal and external environment. Furthermore, cognitivists assume that the cognitive system makes use of its representation system, in order to generate adequate behavior for survival and/or successful interaction with the environment, e.g., Andersen (1990), Boden (1990), Posner (1989), Osherson *et al*. (1990).

To achieve adequate behavior and survival a cognitive system attempts to establish a stable (feedback) relationship both inside the system and with the environment— compare with the concept of homeostasis, e.g., Maturana *et al*. (1980), Varela *et al*. (1991). In humans and most other natural cognitive systems, the nervous system is assumed to be the substratum for the representation system. The neural architecture besides the body structure (Peschl 1994a, b) holds/embodies all of the particular cognitive system's knowledge. Thus, it is responsible for its behavioral dynamics.

2.3.2. *The Environment.* Any cognitive system is embedded into its environment. The environment can be characterized as a flow of energy and information consisting of *per se* meaningless patterns and regularities. From the constructivist perspective the term 'environment' refers in principle to Kant's concept of the 'thing-in-itself'. It cannot be accessed directly. In spite of all efforts of science to find out more about the 'true' or 'objective' nature/structure of the environment, cognitive systems are only able to perceive representations/constructs of the environment, namely, those representational constructs that are generated by the nervous system in the course of interacting with the environment as well as with its own neural states.

It is only in the process of interaction with a cognitive system that environmental states/patterns receive individual meaning. According to Roth (1991, 1994) meaning or *semantics is the specific influence or the effect that environmental states/dynamics have on a specific cognitive/representational system*, i.e. on the structure and the current state of a particular cognitive system. The representational structure/state itself is the result of all phylo- and onto-genetic developments of the particular cognitive system (= the total of the system's 'experiences', i.e. the interpretations the cognitive system has performed).

In human – computer interaction we are dealing with two coupled dynamic systems, since it is an interaction between a cognitive system and its environment. Both systems

- can be characterized by having their own subsystems for interaction,
- are following their own dynamics and try to influence and modulate each other.

Although cognitive systems are also part of the environment, in particular, the human cognitive system tries to achieve a state of homeostasis (= the criterion for life and/or survival) by externalizing certain behaviors that modulate the internal and external environment in a beneficial way. In any case, a complex pattern of interactions and different levels of knowledge are involved in this interaction (see Peschl 1994b for further details). The basic assumption is that a cognitive system has to hold some kind of information or knowledge about its environment, and thus, representation of knowledge, in order to survive in this environment. Representation still matters.

2.3.3. *The Computer.* The idea of computer-supported task accomplishment is to transfer parts of representational structures (= knowledge to solve a certain problem or task) from users into a computer program that—by making use of these knowledge structures—is then capable of mimicking certain cognitive capacities at some level of abstraction. The computer system runs these programs automatically by exclusively transforming and manipulating strings of bits according to certain rules which are determined by algorithms or heuristics. As it is the case with any other artifacts, it is only the act of interpretation by a human that brings meaning into these meaningless patterns, as they are perceived as meaningful symbols as soon as displayed on the screen, or as soon as they are perceived as sound-waves and interpreted as spoken language. The computer system's output triggers and modulates the cognitive/representational dynamics of the human user who is interacting with the computer system (and vice versa).

2.4. *Pitfalls in knowledge acquisition and representation*

The epistemologically relevant part of human-computer interaction, where the latter is considered as an interaction between two dynamically coupled systems, is the process of transferring knowledge from the human cognitive system to the representational structure of the computer system (data structures, algorithms, rule systems, semantic networks, neural networks etc.). How does the computer system and its representational structure obtain the 'knowledge' that allows them to solve a problem or to achieve a certain task? There are at least two answers to this question which do not necessarily mutually exclude each other (and we will actually integrate the resulting options for representation and acquisition in section 3):

(i) The knowledge is transferred from the human (expert) to the computer system. Some kind of mapping between the human's representation system and the computer system's representation mechanisms (e.g., data structures, programs) is defined. Most of the current knowledge engineering techniques rely on the following procedure: When humans make experiences in the real world (= environment), they construct knowledge and theories about the environment. This knowledge can be externalized by using some kind of language. These linguistic expressions are then formalized by a knowledge engineer or a programmer, and transformed into algorithms, rules, and/or data structures. Hence, the computer system makes use of already predefined or 'pre-represented' representations.

The outstanding feature of computer systems is that they are able to handle syntactically huge amounts of data which normally cannot be managed by a human at the same level of abstraction. They allow to manipulate data with extremely high speed and accuracy, and thereby to make structures explicitly available, such as the rules they follow when processing inferences.

(ii) The computer system itself learns from its 'experiences' with the environment in a trial-and-error process. This strategy is the one that most of the approaches in the domain of artificial neural networks, computational neuroscience, e.g., Rumelhart *et al.* (1986), McClelland *et al.* (1986), Hertz *et al.* (1991), Churchland *et al.* (1992), and of genetic algorithms, e.g., Holland (1975), Goldberg (1989), Mitchell *et al*. (1994), follow. The basic idea can be summarized as follows: In the beginning of the learning procedure the computer system has (almost) no useful knowledge (to fulfill the desired task), i.e. its behavior follows random patterns. Learning algorithms or genetic operators adapt their representational structure (i.e. synaptic weights, genetic code etc.) in a trial-and-error process until some useful or desired behavior (judged by humans) is achieved through the representational structure.

This strategy is similar to the processes that occur whenever a human or any natural system has to learn something. He/she/it adapts to certain environmental regularities that are useful for the system's survival in order to make use of them in a beneficial way. In both cases the result is a representational structure (in the brain or in the computer system) that is said to be capable of dealing successfully with certain aspects of the environmental dynamics in the context of accomplishing a certain task, such as solving a problem. The difference between this and issue (i) is that no prefabricated chunks of knowledge are mapped/transferred to the representation system—the cognitive/computer system rather has to figure out a way to solve a certain problem by adapting its knowledge structures in a trial-and-error process.

In both cases the implicit assumption about representation is that the resulting knowledge structure in the computer system has some kind of similarity or even iso/homomorphic relationships to the environmental structure, as for instance in case (i). Looking more closely at this assumption, it implies some kind of homomorphic relationship between the structure of the environment, of the representation in the (human) cognitive system, as well as of the representation in the computer system. As we will see, cognitivists argue that due to this relationship of (structural) similarity it is possible to enable humans to solve the problem of survival in their environment. Moreover, humans are able to solve problems with this kind of 'structure preserving' representation, computer systems can do similar things by applying the same representational mechanism.

Most approaches to cognitive modeling in human–computer interaction follow the traditional understanding of cognitive engineering. Models following this tradition are based on symbol manipulation or propositions, and have not been as successful as originally hoped. Their success has been limited to rather specialized domains that can easily be described by formal specification techniques, such as first order logic. In the following, the reasons why traditional cognitive models, and thus, traditional representation techniques cannot be utilized for semantic and pragmatic representation in a straightforward way, will be described.

2.4.1. *Shifting Semantics.* It has become evident in the field of knowledge engineering, e.g., van de Riet (1987) (in particular with logic-based representations—see Peschl 1994), and software engineering, e.g., Downs (1987), that in the process of extracting knowledge from an expert or a user, and transferring it into a computer system a lot of information gets lost for various reasons (certain parts of the knowledge cannot be verbalized, cannot be formalized etc.). It does matter not only that information gets lost in the process, but also that semantics is changed in many cases. In fact, it seems that the so-called loss of information is only an extreme case of a change in the semantics. This observation does not only affect symbolic knowledge representation, but also pictorial representation, such as visual ambiguities. It has to be considered to be crucial for human–computer interaction, as most of these 'semantic shifts' occur at the critical step when one kind of knowledge representation is transformed into another.

Semantic shifts in the process of transferring knowledge from humans to computer systems and vice versa occur due to a variety of reasons:

(a) Natural language is the major instrument to externalize human (internal) knowledge. As has been shown by Polanyi (1966), Berry (1987) and others, any kind of language is capable of externalizing only a small fraction of the semantics that humans have in mind when they try to externalize a particular chunk of knowledge by making use of language. Hence, the 'tacit' or 'implicit' knowledge is not only lost in the moment of externalization, but also some kind of semantic distortion occurs. Due to the individual differences in onto- and phylo-genetic experiences of humans different receivers of an (externalized) language utterance will interpret these *per se* meaningless, syntactic environmental patterns differently, and in a different way from the sender of the message. Although both parties are referring to the same environmental pattern/syntactic structure, they have different meanings 'in mind' when they are using it or referring to it.

(b) Whenever someone/thing is externalizing behavior through (language) symbols, and someone/thing else tries to interpret these *per se* meaningless artifacts, the semantics for different users and/or designers and/or experts might differ considerably. Although they are confronted with the same symbol, icon, graphical representation etc., these artifacts might trigger different internal representations and semantics in the participating cognitive systems. As a consequence, the occurrence of semantic shifts cannot be avoided in principle.

(c) Despite all attempts to introduce 'semantic features' into symbol systems, natural language is deprived of its final semantic features and dimension in the process of formalization. Hence, the distortion is taken even one step further in the process of formalizing natural language into purely syntactic and formal structures. Symbolic representations (as well as pictorial representations) remain syntactic in principle. Losing the semantic dimension implies, however, more freedom in the process of interpreting these syntactic/formal structures that, in turn, may lead to unintended semantic shifts.

(d) In most of the artificial representation systems a lack of symbol grounding can be found, e.g. Schefe (1990). Semantics is assumed to be somehow externally defined or given. Furthermore, it is assumed that the semantics is more or less stable over a period of time. Epistemological considerations reveal, however, that:

(i) semantics change individually in minimal increments (according to the experiences that individuals make);

(ii) there is no construct such as 'the one given semantics'—public as well as private semantics are a steady flow. Semantics is always system-dependent and communication is based on mutually adapting the individual use of symbols—compare also the concept of a consensual domain as basis for a public semantics, e.g., Becker (1991), Glasersfeld (1983, 1984), Maturana (1978), Maturana *et al*. (1980).

Consequently, the idea of (a) a somehow externally (or 'naturally') given semantics as well as (b) of holding the semantics stable does not work—knowledge representation techniques should rather provide means to deal with the phenomenon of an 'experience-based individual and dynamic semantics'.

2.4.2. *Clarification of Mutual Understanding.* As mentioned already, a major distortion of semantics occurs in the process of transforming one form of representation into another, namely, in the process when an internal representation is externalized and received by another system and transformed into its internal representational format. This process happens in any type of human – computer interaction. The problem here is that, in contrast to human – human interaction, it is difficult for both parties to ask whether the respective system really 'understood' what the other was trying to convey. This fact is due to the implicit assumption, however, not working assumption, that language and even pictorial/iconic representations are based on a stable and somehow 'given' semantics. Hence, misunderstandings in human – computer interaction cannot be clarified the same way as in human – human interaction. Even in human – human communication a 100 per cent overlap cannot be guaranteed (cf. Glasersfeld 1983).

2.4.3. *Passive Mapping.* Both in propositional and in pictorial representations the underlying idea of representation can be characterized as follows: the environment is mapped more or less passively to the representational substratum. Although most approaches in this field distance themselves from the idea of a naive mapping (i.e. naive realism), an unambiguous stable referential/representational relationship between the structure of the environment and the structure of the representational space is postulated. A symbol or a (mental) image refers to, represents, or stands for a certain phenomonen, state, or aspect of the (internal or external) environment.

Empirical research in neuroscience gives evidence that no such stable and unambiguous referential relationship between *repraesentans* and *repraesentandum* can be found (Kandel *et al*. 1991, Churchland *et al*. 1992, Shepherd 1990). A referential representational relationship can be found only in peripheral parts of the nervous system. But even in these areas there is no evidence for real stability, as the original stimulus is distorted in the process of transduction. It seems that neural systems do not follow this assumption of a referential representational relationship. As it is discussed in Peschl (1994) there are not only empirical, but also epistemological and system-theoretical reasons why the concept of referential representation does not apply to neural systems. It can be shown that in highly recurrent neural architectures (as our brain) neither patterns of activations nor synaptic/weight configurations, nor trajectories in the activation space do refer to environmental events/states in a stable (referential) manner. This fact is due to the influence of the internal state on the whole representational dynamics of (as well as on the input to) the neural system. As an implication, it becomes necessary to rethink the representational relationship between the environment and the representation system (see below). Questioning this relationship is crucial for designing human – computer interfaces, as their design is based on assumptions stemming from a referential understanding of representations, such as icons, symbols, images of desktops etc.

2.4.4. *Static and Dynamic Coupling.* Two different aspects of representation are relevant to investigate the problem of coupling a cognitive system to its environment, as it occurs in the course of human-computer interaction:

(i) mapping or modeling the environment to/in the representational *structure*, i.e. the goal is an adequate and accurate model, picture, description etc. of the environment;
(ii) generating (adequate) *behavior*: an equally important task of a representation system is to enable and to generate behavior that allows the system to accomplish a certain task, such as solving a problem.

Both in the propositional and pictorial approach the aspect of mapping the environment to the representational substratum is more important than the aspect of generating behavior. The implicit assumption of these approaches is as follows. If the environment is represented/depicted as accurately as possible, then it will be extremely easy to generate behavior that adequately fits into the environment (that fulfills a desired task). As language and/or human images seem to represent our environment successfully, it follows that accurate predictions can be made by making use of these representations. Thus, the environmental dynamics can be manipulated, predicted, and/or anticipated efficiently with this kind of representation. In other words, in case the criterion of accurately mapping the environment to the representational substratum can be met, designers do not have to care about the aspect of generating adequate behavior.

2.4.5. *Instability of Referential Relationship.* From the constructivist perspective, according to Glasersfeld (1984, 1995), Maturana *et al.* (1980), Varela *et al.* (1991), Roth (1994), and Watzlawick (1984), the claim for an ‘accurate mapping’ should not and cannot be met, since humans will never have direct access to the structure of the environment. Hence, it is impossible to determine how ‘accurate’, ‘true’, or ‘close’ the representation of the environment compared to the ‘real’ environment actually is. The only level of accuracy that can be determined is the difference between (cognitive) representations of the environment and the (computational) representations that have been constructed by engineers. In many cases it has turned out, however, that the human representation of the environment is not the best solution to a given problem—consequently, it is questionable to elevate the human way of representing the environment above other forms of representation and to use it as a standard against which other forms of representation have to compete. It is by no means clear why the human (cognitive or even scientific) representation of the world should be more accurate or more adequate than any other form of representation.

As it has been discussed previously, there is no empirical evidence for explicit propositional or ‘picture-like’ representations in the brain. This fact also implies that neural systems do not generate adequate behavior by making use of referential representations. It rather turns out that any natural nervous system is the result of a phylo- and onto-genetic process of adaptation and development. The goal of this process is not to create an accurate model or representation of the environment, but rather to develop these physical (body and representational/neural) structures that embody a (recurrent) transformation capable of generating functionally fitting (= successful) behavior. In natural (cognitive) systems it seems that the aspect of generating behavior is more important than the aspect of developing an accurate model or internal ‘picture’ of the environment. What we can learn from these systems and their adaptational strategies is that it is not necessary for a system to possess an accurate mapping/representation of the environment in order to generate successful behavior. Since ‘accurate representation’ of the environment means ‘accurate’ with respect to other cognitive systems’ representation of the environment, it does not follow necessarily that an ‘inaccurate’ representation might not lead to more efficient behavior.

3. Constructivist cognitive engineering for task-based and user-oriented interaction

After having specified fundamental problems of knowledge representation and acquisition, we have to put these findings into the context of cognitive engineering for user interface development. When discussing user interface development based on nowadays technology and workplace design we have to be aware of the following facts:

(i) *Multiple Modalities for Interaction.* In the beginning, user interfaces have been mostly designed command-oriented because of the machine’s limitations to handle a variety of media, channels, and modalities at a time. Today, the technology support in principle the development of adaptable and flexible interaction modalities. At the same time, the goals the engineers have to achieve have become more and more complex, if not contradictory. For instance, computer systems not only have to be easy for most users to handle, but also to be capable of supporting trained users to exploit the artifact for more complex purposes.

(ii) *Complex Task Accomplishment*. Moreover, problem solving, i.e. task accomplishment, is often 'fuzzy' or 'wicked' in the sense, that it cannot be described precisely in advance. In such cases, solutions can only be found by moving from partial problem solving results via learning from experience to a more complete understanding.

Before providing a synergistic approach to cognitive engineering we have to give the implications of the findings in section 2 for the acquisition of knowledge about tasks and end users as well as representing the acquired knowledge in design models. Knowledge about tasks and end users is mutually related, since:

(i) constructivist cognitive models represent users. Their activities depend on their intentions which allows us to conclude that there is an indirect relationship between these intentions and the tasks they have to perform (Dutke 1994, p. 12).
(ii) fragmented pieces of observation when represented in cognitive models may not result in any meaningful implications when they are not examined in the context of end user task performance (Chen *et al.* 1997, p. 27).

Section 3.1 deals with epistemological consequences of the findings presented in section 2.4 for knowledge acquisition and representation. In section 3.2 a task-based framework for representation is enhanced with a sub-symbolic representation scheme for cognitive end user modeling. The approach is a conceptual and practical one to overcome the addressed deficiencies of traditional techniques for the acquisition and representation of end users, their functional roles and individual knowledge for task accomplishment.

3.1. *Epistemological consequences for user interface development*

3.1.1. *Predictable User Behavior?* Most of the current models of cognition are based on the (implicit) assumption that humans are perfect logicians when they perform their tasks. Humans are assumed to be able to 'determine' (complete a task) in a finite time span, e.g., Lenat (1988). The prerequisite to make this assumption work is to identify a proper representation of task accomplishment in first order logic. Church has proven a theorem stating that there is no algorithm within the predicate calculus that can achieve that predictability. Moreover, finite articulation and finite 'run-time behavior' does not relate to human cognition and behavioral dynamics, and consequently, to conceptualized cognitive models. The underlying neural structures of natural cognitive systems follow a radically different dynamic. Cognitive systems have to be understood as dynamic systems (e.g. Gelder *et al.* 1995, Port *et al.* 1995) rather than as logical theorem provers. (Human) logic is only a second or third order (representational) phenomenon that emerges at the level of so-called higher cognitive processes (Peschl *et al.* 1988). In any case, it is based on fundamental neural processes that do not follow the structure and dynamics of first order logic. Traditional techniques from artificial intelligence, such as symbol processing or first order logic, postulate that the emergent level of logic represents an adequate description, explanation, and model for the underlying internal neural/cognitive processes (see below). In contrast to that, empirical research in cognitive neuroscience, e.g., Kandel *et al.* (1991), gives evidence that no such logical structures can be found on any neural level.

3.1.2. *Empirical Evidence?* Although empirical/neuroscientific evidence for the propositional as well as pictorial approach for representation and processing is rather poor, there are areas in the brain that seem to be related to the processing of language, semantics, propositions, mental images etc. However, the only finding that is known from these areas is that if these brain areas are damaged in one way or the other, certain cognitive abilities are not present any more (Kandel *et al.* 1991, Churchland *et al.* 1992). Neuroscience provides almost no knowledge or theories concerning the processing mechanisms/architecture underlying these cognitive phenomena. From this poor evidence it seems to be questionable to postulate explicit representational concepts for human-oriented (isomorphic) cognitive engineering, such as the pictorial or propositional paradigm does.

3.1.3. *Semantics Matters.* Even when semantics and pragmatics become part of traditional cognitive models, such as proposed in Newell (1982, 1989) and Sticklen (1990), the resulting mechanisms for problem solving require dynamic checking against the validity of the represented knowledge. Following the tradition of other sub-disciplines in computer science and considering humans as well as machines to be information-processing systems exchanging does not facilitate this task, due to the separation of, syntactic, semantic, and pragmatic aspects ('... thus it is a hypothesis that these symbols are in fact the same symbols that we humans have and use everyday of our lives. Stated another way, the hypothesis is that humans are instances of physical symbol systems, ...' Newell (1980), p. 116). This 'Physical Symbol Systems Hypothesis' is widely used for modeling cognition, e.g., Newell *et al*. (1989), and has been proven useful for behavior-oriented models in commercial artificial-intelligence applications. However, the results can only be used in limited problem domains, e.g., Ernst *et al*. (1986). As long as the application of symbols remains restricted to the level of syntactic processes, that hypothesis may be justified. Whenever it is applied to more complex domains, such as human–computer interaction, it has to be reflected epistemologically, e.g., Peschl (1990), to provide evidence for usability at the semantic level.

3.1.4. *Structure Versus Behavior?* In both the pictorial and the propositional approaches to representation a similar concept of processing is applied. An algorithm manipulates/operates on the representational structure (i.e. on the symbols or mental images). There is a clear distinction between the processing part and the representations, on which these processes operate (i.e. processor-memory distinction). The processing part seems to be actively involved in the dynamics of the system, as it operates on the representations. The representations, on the other hand, seem to play a rather passive role for two reasons:

(a) as mentioned above, they are the result of having been projected from the human representation system to the artificial representation system (i.e. they are passive in the sense of being preprocessed and passively mapped);
(b) an algorithm executes operations over these representations which are assumed to stay in a stable relationship with the environment, i.e. they remain rather passive as they are manipulated by an algorithm similarly as humans manipulate the (passive) matter of the environment.

This concept of distinguishing between processing and memory has its roots in the structure of the Turing machine that inspired the computer metaphor for cognitive processes. In neural systems, however, no such distinction can be found. Usually, the synaptic connections/weights are considered to 'hold the knowledge' of the neural system. Patterns of neural activations are assumed to be responsible for the representation of the current representational state. It is not clear which part of the system takes over the role of the processor. Furthermore, the synaptic weights (= the neural system's 'knowledge') turn out to be not passive at all—they are responsible for controlling the flow/spreading of the patterns of activations. It can be concluded that it is the interaction between the patterns of activations and the configuration of the synaptic weights that is responsible for both the representation of the knowledge and for generating the system's behavioral dynamics.

3.1.5. *Functionalism = Neo-Behaviorism?* The lack of empirical evidence might be one of the reasons why traditional cognitive modeling restricts itself to a functionalistic account in most cases. Cognitive models describe the functional properties that can be derived from the 'behavioral surface' of the observed cognitive system. These behavioral descriptions are used as 'explanatory vehicles' for internal representational processes. A lot of speculation and commonsense concepts are involved in these explanations/theories about internal representational processes, as the actual internal/neural structures are not taken into account. However, with the advent of modern techniques, theories, and methods in empirical neuroscience, as well as of novel concepts from computational neuroscience, such as proposed by Churchland *et al*. (1990), Churchland *et al*. (1992), Anderson *et al*. (1991), Hanson *et al*. (1990), Gazzaniga (1995), Sejnowski *et al*. (1990), basic concepts have been discovered that can be applied to any level of neural processing. They address the spreading of activations, distributed processing and representation, adaptive processes, 'Hebbian' learning as the basis for any kind of learning—LTP, LTD etc., according to Hebb (1949), Brown *et al*. (1990), Nicoll *et al*. (1988) and others. These findings already suggest a completely different concept of (neural) representation mechanisms/concepts than the propositional and/or pictorial approaches postulate.

3.2. *Static and dynamic user modeling in TADEUS*

In this subsection we will revisit TADEUS, a modeling approach and an environment for task-based and user-oriented user interface prototyping and generation. Since the approach has been restricted to static modeling, the user model will be enhanced with dynamic modeling facilities, meeting the objectives for constructivist cognitive engineering.

The TADEUS (Task Analysis/Design/End User Systems) approach (Stary 1996a, Stary *et al*. 1997a) provides a framework (in the sense of a global

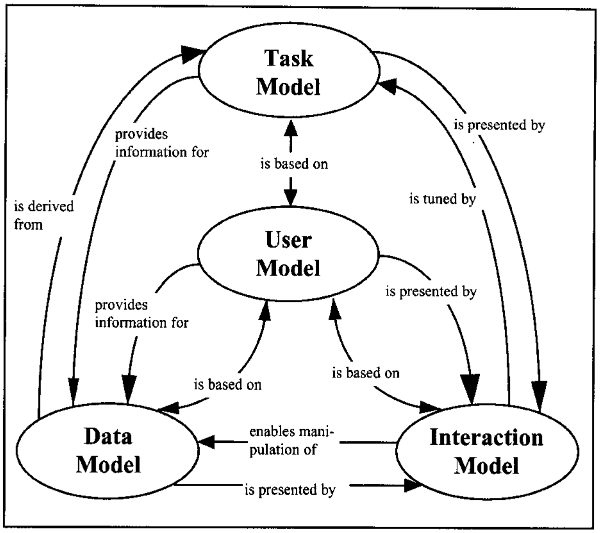

Figure 3. The TADEUS reference model (framework).

reference model, see figure 3) for representation, a methodology, and a corresponding environment for user interface development. The inputs for the development of the TADEUS framework and representation scheme have been provided by techniques from workflow modeling as well as by user interface description languages. TADEUS aims at integrating task-based and user-oriented developnent of interactive software, as these issues are mutually related (see the beginning of this section).

The understanding of end users and their organization of work requires a conceptual framework of context-sensitive components for interactive systems: task model (including a model of the organization and its workflow), problem domain (data) model, user model and interaction model. All of these have to be related to each other statically and dynamically.

The task model comprises the splitting up of end user tasks according to the economic and social organization of work. The user model details the individual perception of tasks, data structures and interaction devices, such as personal experience and preferences, access modalities to tasks and data, individual task organization, and social conventions at the work place. The data model provides the static and dynamic information about the functionality of the system and has to be derived from the task model. The interaction model captures all devices and styles that are required by the users during interaction. In relation to the task and the data model, all presentation issues concerning tasks, functions, and data structures have to be specified in the interaction model.

In order to enable workflow-oriented prototyping TADEUS has been based on a single-object oriented notation, namely the one proposed for Object-Oriented Systems Analysis OSA (Embley *et al.* 1992). Hence, the knowledge about the functionality of an interface, as well as about the behavior (how tasks are accomplished) can be kept encapsulated, transparent and consistent throughout analysis, design and implementation. Moreover, since the design is specified in an object-oriented notation, code can automatically be generated from the representations in the TADEUS environment.

The basic design activities using TADEUS for user interface development are as follows:

- specify a *task* and *user (role) model* according to the organization of work;
- derive a *data model* from the tasks and user organization (as specified in the task model) in the problem domain;
- establish an *interaction model* through the integration of the task model and the data model with interaction devices and styles (modalities);
- give the *functional specification* after or while *prototyping* with end users, in order to generate code.

In order to develop a user model, the designer has to find answers to the following questions:

- Who are the users of the system?
- How do they perceive and accomplish tasks?
- Which artifacts do they use to perceive and accomplish their tasks and how do they use these artifacts?
- What kind of data do they need to manipulate to achieve their tasks?
- How do they arrange these data for task accomplishment?

The designer might follow the subsequent procedure to provide the required inputs for setting up the user model:

(1) *Identify the functional roles required for task accomplishment.* Information might be found in charts describing the organization of work and the qualification profiles for the tasks. Interviews and other data acquisition methods are advised to achieve a complete description.

(2) *Compare the behavior of end users during task accomplishment to the formal descriptions acquired in step 1.* Interviewing, observing and other acquisition methods should lead to a consistent picture of the work and user context.

(3) *Identify the artifacts that the users (want to) use to accomplish the tasks*. Interaction media that are used for task accomplishments are defined.
(4) *Define the data that the users need to manipulate*. For each user or user group the data have to be acquired that have to be manipulated from the organizational as well as individual perspective.

Designers may handle user, interaction and data modeling separately as long as required, as long as they act within the context of the task model. Finally, an (object) architecture remains to be implemented as the result of several modeling activities—a strategy conventionally followed in traditional object-oriented development. However, the design process can be kept as a loose order of specification activities and mutual adaptation procedures within and between different models (see also the semantic relationships in Figure 3 for the integrated process of software, data, and dialog engineering).

Since the starting point for system design is a proper representation of the knowledge acquired through business analysis, TADEUS supports the representation of the flow of work comprising the static and dynamic organization of tasks, positions and roles of people involved in task accomplishment. The Object Relationship Diagram (ORD) in Figure 4 of an airline agency is an example for the structural representation of that type of design knowledge.

State transition diagrams (Object Behavior Diagrams OBDs) have to be specified, in order to reflect the detailed process of task accomplishment, and finally, the behavior of the application to be built. The tasks (activities) involved in the described business in Figure 4 are the handling of Flight Request and Ticket Request. The roles of people involved are Employee, Customer, Manager, and Agent. The temporal relationship used in the ORD for indicating the flow of work is 'before'. Data flow specifications are specified through 'utters' to provide the required input for task accomplishment in the ORD. The passing of control information is also captured in the ORDs through 'informs' between Manager and Agent after updating flight data. Several assignment relationships are used for further task description and the TADEUS model integration:

- between tasks and roles, e.g. Agent 'handles' Flight Request;
- between tasks and subjects of work, e.g. Update Request 'concerns' Flight;
- between roles and subjects of work, e.g. Agent 'attempts to match flight request with' Flight.

3.2.1. *Static User Modeling.* In TADEUS, static user modeling involves several relationships. Each of the involves the functional role of end users:

- handles
- creates
- concerns
- informs
- controls
- is part of.

In the following these conceptual relationships are detailed in their respective context:

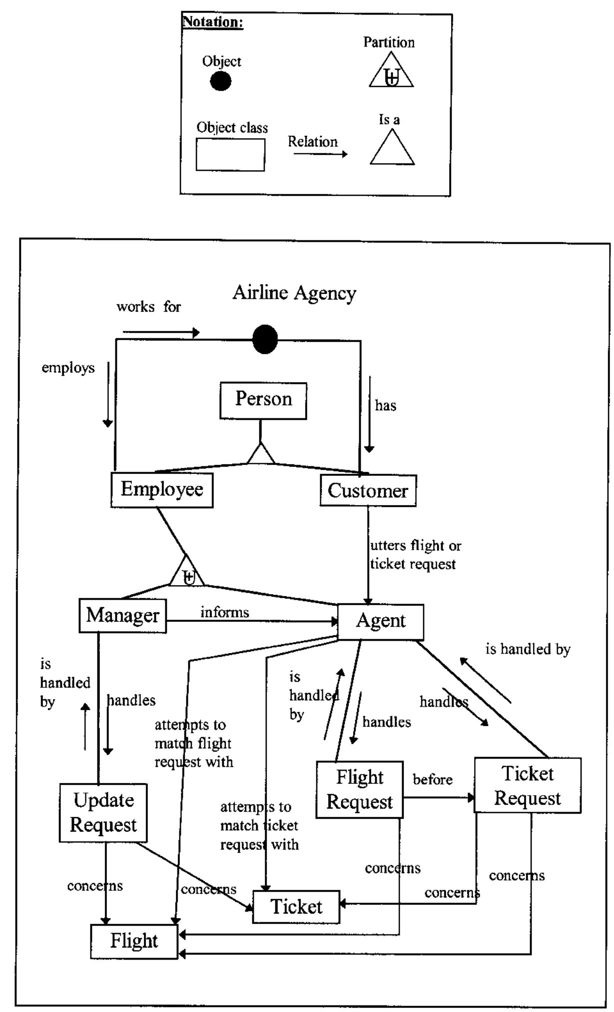

Figure 4. A Sample Object Relationship Diagram ORD of an Airline Agency, comprising user roles, tasks, and problem domain data (i.e. an integrated view of three TADEUS models).

Handles

Involves: roles, activities (tasks).

Meaning: denotes the responsibility of a role for a particular activity: The person behind the role may:

- delegate this activity to another agent and/or
- control its accomplishment, or
- perform it by him/herself.

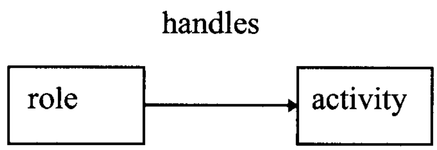

Figure 5. The semantic relationship 'handles' in TADEUS.

Creates

Involves: roles, activities (tasks, sub-tasks), problem domain data.

Meaning: denotes the creation of data through a sub-task or task.

Figure 6. The semantic relationship 'creates' in TADEUS.

Concerns

Involves:
- roles via 'handles' or 'controls',
- activities (tasks or sub-tasks),
- problem domain data,
- relevant factors for task accomplishment (e.g., quality measures).

Meaning:

- denotes un-specific,
- mediate mutual influence of data, factors or activities.

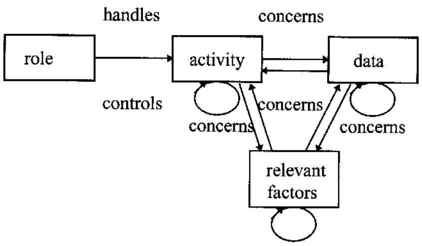

Figure 7. The semantic relationship 'concerns' in TADEUS.

The activities are involved through their assignment to roles that are considered to denote the responsible persons for that activity. The latter situation is expressed through the 'handles' relationship between role and activity.

Relevant factors capture organizational or social settings that are not further refined in the course of application development.

In case no role can be assigned to an activity, 'informs' is used to relate functional roles (see below).

Informs

Involves: roles, activities (tasks, sub-tasks), problem domain data.

Meaning: denotes the passing of information (about data or an activity) from one role to another.

Comments: it denotes the flow of communication with or without data—the option is indicated through a wildcard(*).

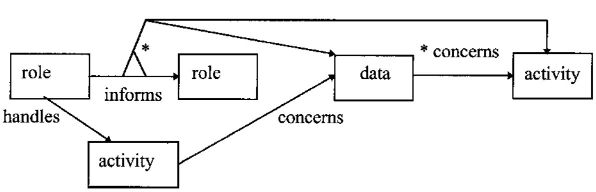

Figure 8. The semantic relationship 'informs' in TADEUS.

Controls

Involves: roles, activities (tasks, sub-tasks), problem domain data.

Meaning: denotes the control of activities or data (that might concern an activity) through a role.

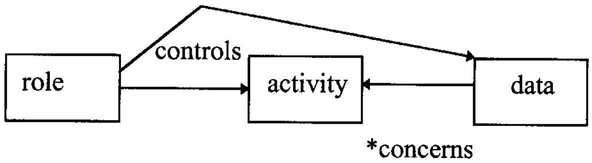

Figure 9. The semantic relationship 'controls' in TADEUS.

Is-part-of

Involves: roles, activities (tasks, sub-tasks), problem domain data.

Meaning: denotes sub-tasks of tasks or sub-tasks; it is sufficient to specify one or more sub-tasks that are related temporally to other sub-tasks.

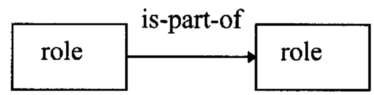

Figure 10. The semantic relationship 'is-part-of' relevant for static user modeling in TADEUS.

The relationships are kept consistent through TADEUS checks throughout the entire development. These checks are provided through particular algorithms for each of the relationships: they implement the

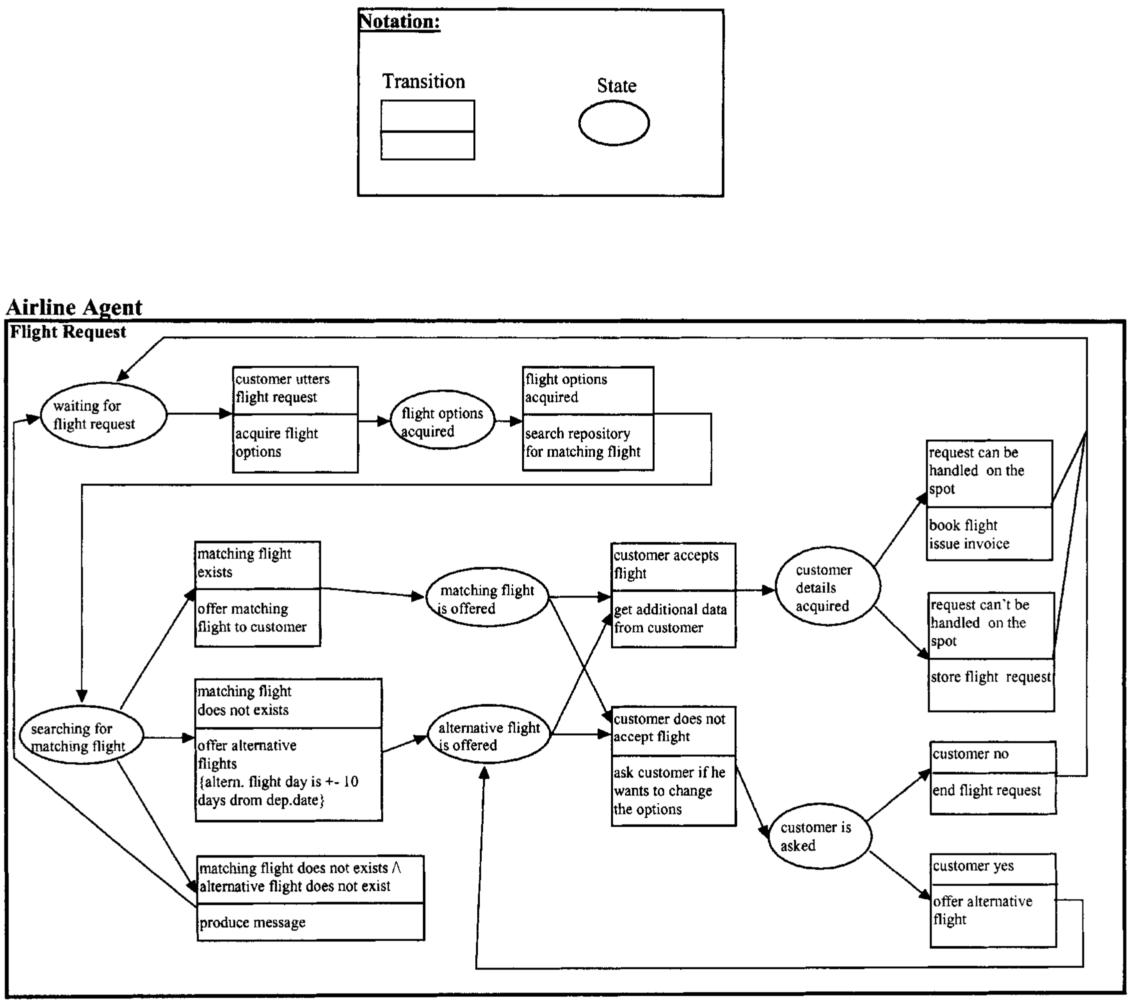

Figure 11. Sample Object Behavior Diagram (OBD) for the user role 'Agent' handling a flight request.

intended semantics at a syntactic, procedural layer. For instance, the 'before' relationship in the ORD in Figure 4 is checked in all the OBDs involving the handling of flight and ticket requests (e.g., in the OBD shown in Figure 11). Figure 11 shows a workflow for the handling of flight requests, as stereotypically performed by airline agents. This way the intended behavior of people and artifacts from the organization's perspective becomes transparent, and static user modeling is enabled from the perspective of task accomplishment.

In order to specifiy the user interface, interaction features specified in the interaction model have to be assigned to the tasks (presenting them to end users) as well as to the data that are going to be manipulated in the course of task accomplishment, for details see Stary *et al*. (1997a). In order to finalize the overall specification of an application in TADEUS, the OBDs of the involved interaction modalities have to be synchronized with task and data handling procedures that are represented in OBDs, too. This way, the life cycles of the objects involved in interaction are synchronized.

As can easily be recognized from this approach, cognitive models of users, such as the airline agent, will remain stereotypical as proposed by Rich (1979) and driven by the idea to build and maintain long term models of individual users, e.g., Finin *et al.* (1985). Stereotypical cognitive engineering and thus, static user modeling has been based on traditional symbolic representation and processing mechanism, such as, e.g. described in Newell *et al*. (1989). Although they might be based on end user tasks, static user modeling assumes an ideal user that will follow exactly predefined sequences of function calls to accomplish a task. However, there are more perspectives involved in interactive task accomplishment. As Tauber (1991) has demonstrated in ETAG, the user's perspective has to be considered and modeled explicitly. This requirement leads to the rethinking of the TADEUS user modeling approach. According to Beale *et al*. (1992) static and dynamic user modeling are complementary components of user interfaces, where the

- static user model is constructed to understand the behavior of these end users that are considered to be the most representative ones (key users), and the
- dynamic user model accounts for various individual behaviors of users within the context of task accomplishment.

3.2.2. *Dynamic User Modeling Coupled with Static Modeling.* We will use dynamic user modeling to identify user behavior patterns that do not only reflect the individual user perception of interaction features, but also the individual organization of tasks within the limits of the task domain. Hence, the OBD for task accomplishment in Figure 11 as well as the assignment of interaction modalities to tasks and in/output components for data manipulation might undergo significant changes according to individual needs and preferences.

In order to recognize, classify by tasks, and adjust user patterns in changing environments traditional representation and analysis techniques did not meet the expectations in dynamic user modeling, e.g., Ye *et al.* (1991), Beale *et al*. (1992), Sanchez *et al*. (1992). As the major reason for deficiencies, the inefficiency of handling knowledge in traditional knowledge engineering has been identified. Neural networks, e.g., Rumelhardt *et al.* (1986), provide more efficient mechanisms to handle implicit knowledge and process knowledge in parallel. In particular, neural network techniques allow for efficient examination of observed information before a system decision is made, since noise and inconsistencies might hinder completing cognitive models or user profiles (Chen *et al*. 1997).

Modeling the knowledge about end user task accomplishment dynamically facilitates system adaptation and improves the effectiveness of feedback to users meeting their abilities and cognitive style. Hence, an application should be able to infer what a user knows in a task and interaction domain. With the help of pattern association default reasoning can be simulated to create a semantically significant number of outputs according to a minimal set of inputs.

Table 1. Part of the adjacency matrix for handling a flight request through agents (see also figure 11).

	acquire flight options	search repository for matching flight	offer matching flight to customer	offer alternative flight	produce message	get additional data from customer
acquire flight options	\	1	0	0	0	0
search repository for matching flight	0	\	1	1	1	0
offer matching flight to customer	0	0	\	0	0	1
offer alternative flight	0	0	0	\	0	1
produce message	1	0	0	0	\	0
get additional data from customer	0	0	0	0	0	\

From the cognitive engineering perspective the dynamic user model in TADEUS is constructed from a predefined 'best-practice'-approach for task accomplishment (i.e. the negotiated model before implementation, as demanded by Winograd *et al*. 1986) as follows:

Step 1: Identify the Starting Point: Starting point can be any Object Behavior Diagram OBD of the TADEUS design representation. Hence, we follow the same principle as we did for the static design. The designer might select a particular perspective (and subsequently switch between several perspectives) of an application without having the risk to cause inconsistent behavior of the application, since all parts of an application's specifications are mutually linked and each of them is updated correspondingly. As a consequence, user modeling might start from any perspective the designer has developed so far in the course of refining the task model. Hence, the starting OBD might either concern the presentation and organization of

- tasks in the workspace
- data required for task accomplishment
- interaction modalities.

Although dynamic user modeling might start with any perspective, it has to be noted that all activities are performed within the context of the Object Relationship Diagram (ORD) of the task model. It captures the essential structure of tasks and task accomplishment which cannot be questioned by the users, since it is the backbone of the business at hand. Hence, the global logic of a business process cannot be manipulated at the level of dynamic (or static) user modeling. What can be adapted, are the presentation of tasks, the selection of a set of preferred ways to solve problems, and the way data for task accomplishment are arranged, i.e. the coupling of data and task management with interaction modalities.

Dynamic user modeling might lead to a change of paths (state transitions) in OBDs that constitute the life cycles of objects. Since the OBDs that serve as input to the dynamic user modeling part of TADEUS represent the ideal case or assumed best practice of task accomplishment (according to the tradition of static user modeling), they can be considered to be an ideal input for a learning mechanism such as a neural network. This situation corresponds to the case when an employee starts doing his/her job on the very first day, and is provided with a VDU-workplace description from the management. Such a description captures default assumptions about the features for interaction and the organization of tasks. This starting point is then successively changed according to the individual style of work of the employee.

In summary, for step 1 a particular OBD has to be selected, e.g., the one shown in Figure 11. It represents the default task accomplishment from the organization of tasks and business logic at hand.

Step 2: Perform Graph-Matrix Analysis: The paths of the OBD selected in step 1 are converted to sets of paired actions that become entries of a so-called adjacency matrix (McGrew 1992), see table 1: '1' in a cell indicates that there is a link between two states in the OBD, e.g., between 'acquire flight options' and 'search repository for matching flight', '0' indicates that there is no connection between the involved states.

The permutation of corresponding rows and columns enables the rearrangement of the relations represented by the entries. This way, underlying behavior patterns might become evident, in particular, since the matrix is not only used to represent the assumed ideal behavior, but also the actual individual behavior end users follow. In the latter case, the entries of the matrix contain the frequency of all the actually followed sequences of state transitions—the cells of the matrix might contain higher numbers than 1, in order to represent the frequency. This procedure might be applied for a set of users, if the designer is interested in clustering user behavior over the entire population of end users.

Step 3: Neural Networking: In this step the matrix is fed to a Kohonen network. It is a self-organizing (i.e. unsupervized) associative memory system (Kohonen 1989, Kohenen *et al*. 1992). Hence, learning occurs

- without being biased by any observer (as it has been the case for static user modeling by the analyst or designer), and
- without indicating desired behavior (output) to given inputs.

According to the individual behavior of an end user

- data are collected with respect to the selected sequence of state transitions;
- behavioral patterns can be identified through categorizing sequences of state transitions; and
- user behavior can be predicted, as soon as the user enters a particular sequence.

The procedure for these activities is the following: Each accomplishment of tasks is represented through an adjacency matrix. Feeding it to a Kohenen network leads to a vector representation of the matrix: Each row of the matrix corresponds to a vector. Each vector has as many elements as states might be involved in task accomplishment.

The Kohonen network comprises 96 nodes in a hexagonal space. Hence, the input vectors have to be mapped on this space: For each state of the OBD a node of the network is assigned. The assigned nodes represent the set of marker vectors. In the course of learning, i.e. processing input vectors that stem from any task cycle in the OBD, the states have to be recognized by the network. Those states that are recognized, are marked (labeled). Some of the states will be recognized (i.e. assigned to predefined nodes), some of them will be assigned to separate nodes of the network. Finally, each node of the network that identified a state contains a vector of the weights of the state relative to the other states. This way those states that are mutually related beyond a certain threshold, e.g., '0', can be interpreted to be strongly connected: Patterns of preferred interaction features might occur that can be the basis for further analyses.

Step 4: Extract Consequences for Further Development/ Future Application Behavior: In case the runtime of the application environment is able to react immediately the recognized behavior can be supported directly. In any case, the result of the learning process should be retransferred to an OBD that can be contrasted with the OBD that has been identified as the starting point for dynamic user modeling (step 1). As a consequence:

- regularities in user behavior for task accomplishment could be visualized;
- problem solving activities could be optimized, redefining the initially assumed 'best practice' of accomplishing tasks, e.g. through high-level activities (macros);
- multiple patterns for accessing the same information could become evident or be derived;
- user profiles could be established.

4. Conclusions

User interface development and software cannot be treated in an isolated way anymore, as traditionally assumed by developers. In the field of user modeling it has been recognized that 'system's beliefs about users should be based on the context of the user's task performance. The fragmented pieces of observations may not result in any meaningful implications if they are not examined associatively' (Chen *et al.* 1997, p. 27). Development and performance should rather follow an integrated approach, capturing the:

- software engineering perspective for structured development and functional design;
- data engineering perspective for defining a proper data model;
- business process (re-)engineering perspective, in order to capture the nature and flow of tasks and the organization they are embedded in;
- knowledge engineering perspective to enable machine intelligence through adapting the user interface and the application towards individual user needs as well as to organizational changes; and, last but not least, the
- dialog engineering perspective, in order to take into account components and styles of interaction.

In becoming aware of the demand for integration, another requirement becomes evident, namely to achieve design representations that feature an implementation-independent handling of the structure and behavior of interactive systems. Unfortunately, current trends in industrial software development, such as the provision of user interface libraries (e.g., Microsoft's MFC approach), have to be considered as a step back, since preprogrammed building blocks of software deliver a behavior that cannot be put into its context at design time, e.g., Beer *et al.* (1998). In particular, the gap between the current representational capabilities of cognitive models and the requirements for cognitive modeling in human – computer interaction, does not enable cognitive models to become part of holistic design representations.

In order to embed cognitive engineering approaches fully into interactive software development activities we had to reveal fundamentals in understanding cognitive modeling and the representation of knowledge to facilitate human – computer interaction. Hence, we first analyzed the tradition in cognitive engineering which:

- emphasizes a static representation of user and task knowledge, and
- enforces symbolic representation of knowledge in terms of goals and other explicit elements to capture the structure and dynamics of task accomplishment.

In addition, traditional cognitive engineering is oriented towards an idealized behavior of end users, assuming a set of stable parameters to describe knowledge. This way, neither the individual perception of tasks and task accomplishment, nor changes in skills, preferences, abilities, and behavior can be reflected in time and accuracy.

Based on these findings we had to develop a completely different understanding of human – computer interaction, namely a framework that captures individuality: Human – computer interfaces serve as devices and a form of representation where information is exchanged between an artificial and a natural representation system, in order to accomplish tasks cooperatively. The representations of the computer system and the cognitive system are mutually modulating and influencing each other. From that constructivist perspective, humans and the computer should over time become compatible systems that trigger processes mutually. Interaction has then to be considered as a process transforming data from one representation (system) to another one, via an environment that is neither accessible directly through the computer nor the cognitive systems of end users.

In order to provide interaction features according to cognitive constraints given by the individuals involved in interaction, we have reviewed currently available empirical knowledge about the dynamics of representations in cognitive systems. It turned out that currently applied techniques for knowledge acquisition and representation in cognitive engineering do not meet the needs that are related to constructivist modeling. In particular, there is neither a stable referential representational relationship between the environment and the systems involved in interaction (which is required for traditional cognitive modeling), nor a 'naturally given' common semantics that can be mapped onto symbols for processing. Moreover, traditional engineering relies on a passive reception of behavior data that hinder any change.

In our solution the shortcomings have been removed through enhancing TADEUS (Task Analysis/Design/End User Systems), a technique to acquire and represent design knowledge at an implementation-independent layer. TADEUS captures the context of interaction through modeling tasks and users. It integrates static and dynamic task and user modeling. Information of the static user model serves as input to a neural network that checks the accuracy and validity of the conventionally represented design knowledge. A feedback mechanism ensures the timely correction of parameters of the static representation, in case changes at the user interface have occurred.

The predictability of user behavior has still to be investigated, e.g. through coupling a back-propagation network with the proposed Kohonen layers. Further investigations and tests that should provide empirical evidence for the proposed concepts are currently performed within the TADEUS development environment. Our further research will focus on a set of case studies whether this integrated approach is manageable and actually reduces the time lag that is currently found, until the computer system is able to catch up with changes at the user interface. In case these studies will have led to improvements, the ultimate goal in human–computer interaction has been achieved, namely, that user interfaces stimulate, modulate and trigger the cognitive dynamics of their users in a way that the intended tasks can be accomplished with minimal cognitive effort.

References

ANDERSON, J. R. 1983, *The Architecture of Cognition* (Cambridge, Massachusetts: Harvard University Press).

ANDERSON, J. R. 1990, *Cognitive Psychology and its Implications* (New York: W. H. Freeman).

ANDERSON, J. A., PELLIONISZ, A. and ROSENFELD, E. (eds) 1991, *Neurocomputing 2. Directions of Research* (Cambridge, MA: MIT Press).

BAINBRIDGE, L. 1991, Mental Models in Cognitive Skills, in A. Rutherford and Y. Rogers (eds), *Models in the Mind* (New York: Academic Press).

BEALE, R. and FINLAY, J. (eds) 1992, *Neural Networks and Pattern Recognition in Human–Computer Interaction* (New York: Ellis Horwood).

BECKER, A. L. 1991, A Short Essay on Languaging, in F. Steier (ed.), *Research and Reflexivity* (London: Sage Publishers), pp. 226–234.

BEER, A., MOHACSI, S., STARY, C. 1988, Integrated Design and Automated Test Case Generation, in *Proceedings 'Quality Week 98', San Francisco*.

BERRY, D. C. 1987, The Problem of Implicit Knowledge, *Expert Systems*, **4**(3).

BODEN, M. A. (ed.) 1990, The Philosophy of Artificial Intelligence (New York: Oxford University Press).

BÖHLE, F. and MILKAU, B. 1988, Computerized Manufacturing and Empirical Knowledge, *AI & Society*, **2**(3), 235–243.

BRANDT, D. S. 1997, Constructivism: Teaching for Understanding of the Internet, *Communications of the ACM*, **40**(10), 112–117.

BROWN, T. H., GANONG, A. H., KARISS, E. W., KEENAN, C. L. and HEBBIAN, 1990, Synapses: Biophysical Mechanisms and Algorithms, *Annual Review of Neuroscience*, **13**,, 475–511.

CARD, S. K., MORAN, T. P. and NEWELL, A. 1983, *The Psychology of Human-Computer Interaction* (Hillsdale, New Jersey: Lawrence Erlbaum).

CAROLL, J. M. and OLSON, J. R. 1988, Mental Models in Human-Computer Interaction, in M. Helander (ed.), Handbook of Human-Computer Interaction, (Elsevier) pp. 45–65.

CHEN, Q. and NORCIO, A. F. 1997, Modeling a User's Domain Knowledge with Neural Networks, *Internatioanal Journal of Human–Computer Interaction*, **9**(1), 25–40.

CHURCHLAND, P. S., KOCH, C. and SEJNOWSKI, T. J. 1990, What is Computational Neuroscience? in E.L. Schwartz (ed.), *Computational Neuroscience* (Cambridge, MA: MIT Press).

CHURCHLAND, P. S. and SEJNOWSKI, T. J. 1992, *The Computational Brain* (Cambridge, MA: MIT Press).

DOWNS, T. 1987, Reliability Problems in Software Engineering—A Review, *Computer Systems, Science and Engineering*, **2**(3), 131–147.

DUTKE, St. 1994, *Mental Models: Constructs of Knowledge and Understanding* (Göttingen: Verlag für Angewandte Psychologie) [in German].

ECKHARDT, B. 1993, *What is Cognitive Science?* (Cambridge, MA: MIT Press).

EMBLEY, D. W., KURTZ, B. D. and WOODFIELD, S. N. 1992, *Object-Oriented Systems Analysis. A Model-Driven Approach* (Englewood Cliffs, New Jersey: Yourdon Press).

Ernst, M. L. and Ojha, H. 1986, Business Applications of Artificial Intelligence KBs, *Future Generations Computer Systems*, **2,** 75 – 116.

Finin, T. and Drager, D. 1985, GUMS1. A General User Modeling System, in *Proceedings 1st International Conference on Expert Database Systems*.

Fodor, J. A. 1988, *Psychosemantics: The Problem of Meaning in the Philosophy of Mind* (Cambridge, MA: MIT Press).

Gazzaniga, M. S. (ed.) 1995, *The Cognitive Neurosciences* (Cambridge, MA: MIT Press).

Gelder, T. and Port, R. 1995, It's About Time: An Overview of the Dynamical Approach to Cognition, in R. Port, T.V. Gelder (eds), *Mind as Motion* (Cambridge, MA: MIT Press).

Gentner, D. and Stevens, A. L. (eds) 1983, *Mental Models* (Hillsdale, NJ: Lawrence Erlbaum).

v. Glasersfeld, E. 1983, On the Concept of Interpretation, *Poetics*, **12,** 254 – 247.

v. Glasersfeld, E. 1984, An Introduction to Radical Constructivism, in Watzlawick P. (ed.), *The Invented Reality* (New York: Norton) pp. 17 – 40.

v. Glasersfeld, E. 1995, *Radical Constructivism: A Way of Knowing and Learning* (London: Falmer Press).

Goldberg, D. E. 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning* (Reading, MA: Addison-Wesley).

Hanson, S. J. and Olson, C. R. 1990, *Connectionist Modeling and Brain Function: The Developing Interface* (Cambridge, MA: MIT Press).

Hebb, D. O. 1949, *The Organization of Behavior. A Neuropsychological Theory* (New York: Wiley).

Hertz, J., Krogh, A. and Palmer, R. G. 1991, *Introduction to the Theory of Neural Computation, Vol. 1, Lecture Notes of Santa Fe Institute Studies in the Sciences of Complexity* (Redwood City, CA: Addison-Wesley).

Holland, J. H. 1975, *Adaptation in Natural and Artificial Systems: An Analysis with Applications to Biology, Control, and Artificial Intelligence* (Ann Arbor: University of Michigan Press).

Hollnagl, E. and Woods, D. D. 1983, Cognitive Systems Engineering: New Wine in New Bottles, *International Journal on Man – Machine Studies*, **18,** 583 – 600.

Johnson, P. 1992, *Human – Computer Interaction. Psychology, Task Analysis and Software Engineering* (London: McGraw Hill).

Johnson-Laird, P. N. 1983, *Mental Models* (Cambridge, MA: Harvard University Press).

Johnson-Laird, P. 1993, *The Computer and the Mind. An Introduction of Cognitive Science* (London: Fontana).

Kandel, E. R., Schwartz, J. H. and Jessel, T. M. (eds) 1991, *Principles of Neural Science* (New York: Elsevier) 3rd edition.

Kantowitz, B. and Sorkin, R. 1987, Allocation of Functions, in Salvendy, G. (ed), *Handbook of Human Factors*, (New York: Wiley & Sons).

Kohonen, T. 1989, *Self-Organization and Associative Memory* (New York: Springer).

Kohonen, T., Kangas, J. and Laaksonen, J. 1992, *SOM_PAK: The Self-Organizing Map Program Package, Version 1.2* (Espoo, Helsinki University of Technology Laboratory of Computer and Information Sciences).

Lenat, D. B. 1988, When Will Machines Learn? in *Proceedings 'International Conference on 5th Generation Computer Systems', ICOT*, pp. 1213 – 1245.

Maturana, H. R. 1978, Biology of the Language: The Epistemiology of Reality in H. R. Maturana (ed.), *Perception: The Organization and Embodiment of Reality*, (Braunschweig: Vieweg), pp. 236 – 271.

Maturana, H. R. and Varela, F. J. (eds) 1980, *Autopoiesis and Cognition: The Realization of the Living*, Vol. 42 of Boston Studies in the Philosophy of Science (Dordrecht, Boston: D. Reidel).

McClelland, J. L. and Rumelhart, D. E. (eds) 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Psychological and Biological Models, Vol. II* (Cambridge, MA: MIT Press).

McGrew, J. 1992, Task Analysis, Neural Nets and Very Rapid Prototyping, in B. Russell and J. Finlay (eds), *Neural Networks and Pattern Recognition in Human – Computer Interaction* (New York: Ellis Horwood), pp. 91 – 100.

Mitchell, M. and Forrest, S. 1994, Genetic Algorithms and Artificial Life, *Artificial Life*, **1**(3), 267 – 291.

Moran, T. P. 1984, Getting Into a System: External – Internal Task Mapping Analysis, in *Proceedings CHI'83*, (Amsterdam: North-Holland), pp. 45 – 49.

Moray, N. 1993, Formalisms for Cognitive Modeling, in M. J. Smith and G. Salvendy (eds), *Human – Computer Interaction. Applications and Case Studies*, (Amsterdam: Elsevier), pp. 581 – 586.

Newell, A. 1980, Physical Symbol Systems, *Cognitive Science,* **4,** 135 – 183.

Newell, A. 1982, The Knowledge Level, *Artificial Intelligence*, **18,** 87 – 127.

Newell, A., 1989, *Unified Theories of Cognition* (Harvard: University Press).

Newell, A., Rosenbloom, P. S. and Laird, J. E. 1989, Symbolic Architectures for Cognition, in M.I Posner (ed), *Foundations of Cognitive Science*, (Cambridge, MA: MIT Press), pp. 93 – 131.

Nicoll, R. A., Kauer, J. A. and Malenka, R. C. 1988, The Current Excitement in Long-term Potentiation, *Neuron*, **1**(2), 97 – 103.

Norman, D. A. 1983, Some Observations on Mental Models, in A. L. Stevens and D. Gentner (eds), (New Jersey: Lawrence Erlbaum), pp. 7 – 14.

Norman, D. A. 1986, Cognitive Engineering, in D. A. Norman and S. W. Draper (eds), *User Centered System Design: New Perspectives on Human – Computer Interaction* (Hillsdale, New Jersey: Lawrence Erlbaum), pp. 31 – 61.

Oliver, R. 1995, Interactive Information Systems: Information Access and Retrieval, *Electronic Library*, **13**(3), 187 – 193.

Olson, J. R. and Olson, G. M. 1990, The Growth of Cognitive Modeling in Human – Computer Interaction Since GOMS, *Human – Computer Interaction* **5**(213), 221 – 266.

Osherson, D. N. and Lasnik, H. (eds) 1990, *An Invitation to Cognitive Science, Vol 13,* (Cambridge, MA: MIT Press).

Perner, J. and Garnham, A. 1988, Conditions for Mutuality, *Journal of Semantics*, **6,** 369 – 385.

Peschl, M. F. 1990, *Cognitive Modeling* (Deutscher Universitätsverlag/Vieweg, Wiesbaden). [In German].

Peschl, M. F. 1994, Autonomy vs. Environmental Dependency in Neural Knowledge Representation, in R. Brooks and P. Maes (eds), *Artificial Life IV*, (Cambridge, MA: MIT Press), pp. 417 – 423.

Peschl, M. F. 1994a, Embodiment of Knowledge in the Sensory System and its Contribution to Sensormotor Integration. The Role of Sensors in Representational and Epistemological Issues, in P. Gaussier and J. D. Nicoud (eds), *From Perception to Action*, (IEEE Society Press), pp. 444 – 447.

Peschl, M. F. 1994b, *Representation and Construction. Concepts from Cognition and Neuroinformatics as Fundamentals of a Naturalized Epistemology and Philosophy of Science* (Braunschweig/Wiesbaden: Vieweg). [In German].

Peschl, M. F. and Stary, C. 1998, The Role of Cognitive Modeling for User Interface Design Representations: An Epistemological Analysis of Knowledge Engineering in the Context of Human – Computer Interaction, *Mind and Machines*.

Polanyi, M. 1966, *The Tacit Dimension* (Garden City, NY: Doubleday).

Port, R. and v. Gelder, P. (eds) 1995, *Mind as Motion: Explorations in the Dynamics of Cognition* (Cambridge, MA: MIT Press).

Posner, M. I. (ed.) 1989 *Foundations of Cognitive Science* (Cambridge, MA: MIT Press).

Rasmussen, J. 1986 *Information Processing and Human – Machine Interaction. An Approach to Cognitive Engineering* (New York: North Holland).

Ravden, S. and Johnson, G. 1989, *Evaluating Usability of Human – Computer Interaction. A Practical Method* (Chichester: Ellis Horwood).

Rich, E. 1979, User Modeling Via Stereotypes, *Cognitive Science*, **3,** 329 – 354.

Roth, G. 1991, The Constitution of Semantics in the Brain, in S. J. Schmidt (ed.), *Memory* (Frankfurt/Main: Suhrkamp), pp. 360 – 370. [In German].

Roth, G. 1994, *The Brain and its Reality. Cognitive Neurobiology and its Philosophical Consequences* (Frankfurt/Main: Suhrkamp). [In German].

Roth, E. Bennett, K. and Woods, D. D. 1987, Human Interaction with an 'Intelligent' Machine, *International Journal of Man – Machine Studies*, **27,** 00 – 00.

Rumelhart, D. E. and McClelland, J. L. (eds) 1986, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Foundations, Vol I* (Cambridge, MA: MIT Press).

Sanchez-Sinencio, E. and Lau, C. 1992, *Artificial Neural Networks: Paradigms, Applications, and Hardware Implementations* (New York: IEEE Press).

Schefe, P. 1990, Dimensions, Challenges and Limitations Reconstructing Knowledge in Languages of Artificial Intelligence, in H. Bonin (ed.), *Demystification of Expert Systems* (Heidelberg: Decker & Müller), pp. 13 – 28. [In German].

Sejnowski, T. J., Koch, C. and Churchland, P. S. 1990, Computational Neuroscience, in S. J. Hanson and C. R. Olson (eds), *Connectionist Modeling and Brain Function: The Developing Interface* (Cambridge, MA: MIT Press), pp. 5 – 53.

Shepherd, G. M. (ed.) 1990, *The Synaptic Organization of the Brain* (New York: Oxford University Press).

Stacy, W. 1995, Cognition and Software Development, *Communications of the ACM*, **38**(6) 31.

Stary, C. and Peschl, M. 1995, Towards Constructivist Unification of Machine Learning and Parallel Distributed Processing, in Ford, K., Hayes, P. (eds), Android Epistemology, (MIT Press), pp. 183 – 214.

Stary, C. 1996a, *Interactive Systems. Software Development and Software Ergonomics* (Wiesbaden: Wieweg). [In German].

Stary, C. 1996b, Integrating Workflow Representations into User Interface Design Representations, *Software—Concepts and Tools*, **17,** 173 – 187.

Stary, C., Vidakis, N., Mohacsi, S., Nagelholz, M. 1997a, Workflow-Oriented Prototyping for the Development of Interactive Software, in *Proceedings COMPSAC'97*, (Washington D.C.: IEEE), pp. 530 – 535.

Stary, C. and Totter, A. 1997b, How to Integrate Concepts for the Design and the Evaluation of Adaptable and Adaptive User Interfaces, in *Proceedings 3rd ERCIM Workshop User Interfaces for All* (INRIA), pp. 45 – 60.

Sticklen, J. 1990, Problem Solving Architectures at the Knowledge Level, *Journal of Experimental and Artificial Intelligence*, **1**(1), 1 – 52.

Tauber, M. J. 1991, ETAG—Extended Task Action Grammar, in *Proceedings INTERACT'91* (IFIP, Elsevier).

Thorndyke, P. W. and Stasz, C. 1980, Individual Differences in Procedures for Knowledge Acquisition from Maps, *Cognitive Psychology*, **12,** 137 – 175.

van de Riet, R. 1987, Problems with Expert Systems, *Future Generations Computer Systems*, **3,** 11 – 16.

Varela, F. J., Thompson, E. and Rosch, E. 1991, *The Embodied Mind: Cognitive Science and Human Experience* (Cambridge, MA: MIT Press).

Watzlawick, P. (ed.) 1984, *The Invented Reality* (New York: Norton).

Winograd, T. and Flores, F. 1986, *Understanding Computers and Cognition, A New Foundation for Design* (Norwood: Ablex).

Wood, S. 1986, New Technologies, Organization of Work, and Qualifications: The British-Labor-Process Debate, *Prokla*, **2,** (Berlin: Rotbuch). [In German].

Woods, D. D. and Roth, E. M. 1988, Cognitive Systems Engineering, in: Helander, M. (ed.), *Handbook of Human – Computer Interaction* (Amsterdam: Elsevier Science), pp. 3 – 43.

Ye, N. and Salvendy, G. 1991, Cognitive Engineering Based Knowledge Representation in Neural Networks, *Behaviour & Information Technology*, **10**(5), 403 – 418.