The influence of conceptual user models on the creation and interpretation of diagrams representing reactive systems

Hermina J.M. Tabachneck-Schijfa\textsuperscript{a}, Jan Herman Verpoortena\textsuperscript{b}, Rob van de Weg\textsuperscript{a} and Roel J. Wieringa\textsuperscript{a}\textsuperscript{b}

\textsuperscript{a} Department of Information and Computing Sciences, Universiteit Utrecht, Utrecht, THE NETHERLANDS
\textsuperscript{b} Department of Computer Science, University of Twente, Enschede, THE NETHERLANDS

In system design, many diagrams of many different types are used. Diagrams communicate design aspects between members of the development team, and between these experts and the non-expert customers and future users. Mastering the creation of diagrams is often a challenging task, judging by particular errors persistently found in diagrams created by undergraduate computer science students. We assume a possible misalignment between human perception and cognition on the one hand and the diagrams’ structure and syntax on the other. This article presents the results of an investigation of such a misalignment. We focus on the deployment of so-called ‘conceptual user models’ (mental models, created by users in their mind) at the creation of diagrams. We propose a taxonomy for mental mappings, used for categorization of representations. We describe an experiment where naive and novice subjects created one or several diagrams of a familiar task. We use our taxonomy for analysing these diagrams, both for the represented task structure and the symbols used. The results indeed show a mismatch between mental models and currently used diagram techniques.

Keywords: System design, Diagram, Visualization, Mental model, Symbol mapping.

1 INTRODUCTION

In software systems design, two types of systems can be discriminated: reactive systems and transformational systems. Reactive systems are nonterminating interactive systems that respond to stimuli in order to bring about desirable effects in their environment. Examples are information systems, workflow systems, and groupware. Transformational systems merely compute an output from an input, sometimes with some interaction, and then terminate. Examples are compilers, assemblers, and expert systems. The design of transformational systems differs from reactive systems; we will only discuss reactive system design.

Reactive system design is complex, among others because of the interaction of the system with the entities and behavior in the environment. Hence, system design methods (e.g. Yourdon, Statemate, UML) are also complex. Many diagrams of many different types need to be used, visualizing the design and particular aspects thereof, e.g., message sequences between objects in a UML ‘Sequence Diagram’, or states and their transitions in a UML ‘State Transition Diagram’. Surprisingly, computer science students find learning and creating many of these diagram types a difficult task, although the syntactic complexity of the vast majority of any one diagram type is low. Many errors and ambiguities are found in diagrams created by students. As a preparatory data analysis, during a number of years we collected and analyzed such with the aim to improve instruction. We concluded that problems can only partly be explained by the complexity of the assignments, students' insufficient knowledge of the diagram type, insufficient understanding of the system under design or the system domain. Therefore, factors outside this scope must exist.

Diagrams not only communicate design aspects between members of the development team (experienced users of diagrams), but also between such experts and the non-expert customer and future users. If novices encounter fundamental problems during creation of these diagrams, we can assume that naive and novice users will also experience problems with the interpretation of diagrams. This makes the issue far more general than just the learning difficulties of computer science students.

We assume that the problems in learning and creating at least partly stem from problems in the alignment between human perception and cognition on the one hand and the diagrams’ structure and syntax on the other. Therefore, we have chosen for a human-centered research approach. Diagrams map to particular system aspects, as do the symbols in the diagram. Do particular types of mappings block or support the elicitation of an effective conceptual user model (from now on simply called 'mental model')? We examine a hypothesis about the effect of mental models and particular mappings with an experiment, carried out with naive and novice users. Next, we will give the rationale for our choices 'mental model' and 'mapping'.

Mental Models: Mental models are an assumption about the construction of small-scale conceptual models of the perceived reality in our mind. The Scottish psychologist Kenneth Craik\textsuperscript{1} coined the term. He assumed that people need them to be able to reason, anticipate events and to support explanation. Use of the mental model construct is pervasive in the HCI field, probably caused by Donald Norman’s\textsuperscript{2} contribution. He stressed the difference between the "design model", upon which the system is designed, and the "user's
model", the basis for use of the system. For many reasons (e.g. different kinds and levels of expertise), these two models differ, often drastically. A screen interface design is an example of a system image, as is a system diagram: Both communicate the system design. If mental models don't connect to the 'language' that a particular diagram type offers, or to its underlying model, visualization (at design time) or interpretation (at consume time) problems arise. In HCI, the mental model construct supports the creation of a usable design and as such it is located at the core of the user-centered design approach. The assumption is that users use their mental model both to interpret the visible parts of a system in order to decide how to manipulate them towards a particular goal (procedural mental model, e.g. 'task-action mapping'), and make guesses as to what goes on behind the scene (structural mental model, 'surrogate model'). Experts have very flexible procedural mental models because of an accurate knowledge of the system structure and ensuing correct structural mental model. Non-expert users have effective but often incomplete procedural mental models, and always have incomplete and often incorrect structural mental models. Which doesn't matter: it only needs to help users in reasoning about the system while performing tasks. The emphasis is on the procedural knowledge.

Mental models don’t only work for the system you develop them for. Perhaps the simplest example is the tap: water comes out faster if you turn it anticlockwise. Having formed this mental model, you will automatically apply it to other tap-like objects (e.g. gas tap) and even more distant objects like nuts (open/close-loosen/tighten analogy). Analogy and similar meaning are the keywords for a proper understanding of this phenomenon. People analogize often and automatically. Without any problem we understand electrical current from an example using water flow. Although physically these do not match at all, both images share the flow analogy and do express the same meaning - it is the similarity in meaning that allows analogical mapping from one area to the other. Analogies work by mapping similar elements from one thing or situation to another (see for an in-depth discussion of how analogy works).

Mapping Definitions and Problem Exploration: From the previous section it can be understood that the most effective diagrams are the ones that enable users to easily experience an analogical mapping between the diagram and the system under design. This turns our attention to how diagrams are structured and to the shapes and meanings of the used symbols (see figure 1).

Diagramming means mapping to a process or structure. In general, both process and structure can be represented in a non-mapping way, by using images and/or language, but can also be represented by using particular mappings, i.e., analogies. Each has its own qualities. To describe mapping qualities, we assembled the several kinds of mappings found in literature into a mapping taxonomy consisting of eight mapping types, four to be used for the diagram and four for the contained items (symbols). The structure of the taxonomy is achieved by breaking down the key concept 'analogy', first into indirect and direct analogy, next for structure and item, and so on (see figure 2).

Each representation has its own qualities. Language can be used for describing a system, but, as most mental models will be based on visual information, translating to language requires a lot of cognitive effort. It is then still extremely difficult to check the consistency and completeness, both in the mind and from the text, because that requires translation back to the visual format. However, just substituting one word for one object/concept is easier. Because of our experience with language, words and their objects/concepts are so closely connected that reading a single word immediately calls up the image, and vice versa. As it is faster to write a word than to draw an image, words, in the form of labels are often preferred to images in diagrams, but still, the more words are used, the more must be mentally built up and memorized.

Concrete (non-mapping) images are the most effective seen from a cognitive standpoint: If we recognize the image, nothing needs to be remembered or to be processed, as our knowledge system is directly addressed by the visual stimuli, resulting in immediate understanding. Our mental system is very
tolerant with images: often the outline and a few details are enough to trigger recognition.

The 'direct' category of the mappings is almost as good as the non-mapping visualizations, because many relations to reality are preserved, which trigger recognition. However, direct mappings aren't always possible. Then, we have resort to indirect ones. Such images are not immediately recognized but have to be learned. This makes learnability an important issue for abstract symbols and structures (types 1, 2 and 3).

Fig. 2. Types of representations and proposal of a mapping taxonomy, each type clarified with some examples. Non-mapping representations are La=language (either a label or annotation) and Im=image (4 gradations Im/++ to Im/--; ++ means much details, -- means a label is required for understanding). Eight mapping types are proposed.

If we apply this knowledge to diagrams, we have to face some facts. First, in system development often little can be visualized with images or direct mappings, because of the kinds of aspects to be represented. For instance, no such visible thing exists for the notion 'state' in a State Transition Diagram. That is probably the reason why text labels and abstract symbols (mapping type m3; from now on, we will represent mapping types with m1..m8, and non-mappings with La/label and Im/++ to Im/--. See figure 2) are so prominently used. Though, within a diagram type, an abstract symbol, usually a simple shape, carries a single meaning expressly so labelling is not needed, the content has to be denoted with text labels. Second, it is unclear to which degree particular notions used in diagramming connect to mental models. Do our mental models contain concepts such as 'transition', 'fork', or 'specialization', and how much difference do our mental models make between 'action' or 'event'? This possible mismatch problem can be illustrated with some of the errors persistently made by undergraduate students in State Transition Diagrams (STDs): (1)
they find it difficult to decide between a representation as an attribute value or as a state; (2) different students represent different states in the same assignment; (3) many find it confusing that the concept 'state' both applies to working and waiting states and (4) many put both states and activities in the state spaces.

Section 2 contains the research questions, hypotheses, and a description of the experiment design. The outcomes of the experiment are in section 3, and our conclusions in section 4 of this paper.

2 RESEARCH QUESTIONS, HYPOTHESES AND EXPERIMENT METHOD

Research Questions: On basis of the above problem exploration, we are curious about the difference between naive and novice users. Naive users are non-technical students, not ‘tainted’ by knowledge about process diagrams, and, in general, have been much less exposed to abstract representations than novice users. Novice users are undergraduate computer science students who have learned many diagram types but are still not skilled experts at this point. We constructed the following research questions:

1. Which mapping types are preferably used for symbols selected by naive users, and how do these differ from the ones used in the diagram techniques?
2. Do naive users produce self-generated instances of an existing diagram type, and to which extent do novices reproduce such instances when they have free choice to create whatever they like?

Hypotheses: Our main hypothesis is that many diagrams in system development do not effectively connect to the default way of forming mental models. This applies both to the 'language' that a particular diagram type offers and to the overwhelming use of abstract symbols. Although we cannot validate this hypothesis for each diagram type separately, we expect to find patterns that can be generalized. The main hypothesis leads to the following specific hypotheses:

Symbols: 1. Naives follow the 'natural way of lowest resistance': They prefer non-mapped and direct mapped representations (very little cognitive effort required).
2. Novices deploy learned knowledge thus heavily use symbols from existing techniques.

Diagrams: 1. We expect naives' diagrams structure to resemble their default mental models, resulting in a mix of visualized processes and static structure ('surrogate model'), but mainly procedural oriented (“and then – and then” stories). We don't expect existing diagram types to pop up.
2. Novices deploy learned knowledge and therefore will try to express their internal representation into an existing diagram technique, but without much success. We expect problems such as an inconsistent semantics.

Experiment Design: We designed an experiment, in which naive and novice subjects create one or more diagrams from one assignment. For three reasons, we started our research with STDs. First, STDs have a long history and are still heavily used. Second, students find this diagram in particular difficult to master, as is illustrated in section 2.2. And third, the state-transition concept itself is claimed to correspond to a type of mental model7. Similar to the state machine concept ('Mealy machine') underlying STDs, the state-transition concept in mental models discerns actions leading to states. It is largely based on the script idea: A complex task is achieved in several steps, each step with a particular result, which is the next state. For instance, the Yoked State Space hypothesis states that a user performs a semantic mapping (comparison) between a goal space and a device space, in order to decide about the next step to be taken.

The task used in the experiment has a very high potential to be described using states and transitions: making a phone call. Moreover, this task is very familiar to all subjects so we can reasonably expect that all have formed a mental model of this task. First, subjects received a questionnaire to investigate their possible technical knowledge in general and their experience with phone equipment in particular. Next, we asked the subjects to draw a schematic representation of a phone call, in their own way. The exact formulation contained no bias to a particular outcome. After half an hour, the subjects were interrupted and were asked to explain everything about their representation. All diagrams produced in the experiment were analyzed by two researchers together (to achieve consensus in cases of doubt) for the deployed symbol mapping types, number and semantics of different symbols, used diagram type(s) if any, consistency, and completeness of visualization. Data was classified according to the representation and mapping types shown in figure 2.

3 RESULTS AND DISCUSSION

The experiment was carried out by 21 naive and 17 novice participants, which produced resp. 29 and 37 schemas. The analysis resulted in a great amount of data, which cannot be presented due to space
limitations. We will present the most significant results pertaining to our hypotheses in figure 3.

| Symbols: Naives follow the 'natural way of lowest resistance': They prefer non-mapped and direct mapped representations (very little cognitive effort required). | Nodes: 81% non-mapped, from those two-third La/label and one-third Im. The remaining 19% are mapped encodings, from those two-third m3 or m4 and one-third m7 or m8. Edges: 47% mainly Im/--. The other 53% are mapped encodings, from those m8=41% and m3= 59%. (7% nodes and 48% edges contain >1 codings) |
| Novices deploy learned knowledge thus heavily use symbols from existing techniques. | Nodes: 99% non-mapped, from those 94% labels and 6% images(Im/...) Edges: 10% non-mapped, all images Im/-- 90% mapped, all indirect mappings m3 (1% nodes and 3% edges contain >1 codings) |
| Diagrams: We expect naives' diagrams structure to resemble their default mental models, resulting in a mix of visualized processes and static structure but mainly procedural oriented. | m1=17%, m2=38%, m5=42%, m6=3%. So, 56% mainly process-oriented, 41% mainly static-structure-oriented. 21% of all diagrams is a mix of a process and static structure, applying to one-third of the mainly static oriented diagrams. So, 27% of all schemas are static structures only, partly (approx. one-third) cartoon-alike drawings. |
| Novices deploy learned knowledge and therefore will try to express their internal representation into an existing diagram technique, but without much success. We expect problems such as an inconsistent semantics. | m1=76%, m2=24%. In the m1 category, over half were STD-alikes. Other choices are ER, Flow chart, Sequence diagram, SSD, Activity diagram, Communication Diagram. 14% of all diagrams use a combination of techniques. 78% of all diagrams contain semantically inconsistencies. Only 2 diagrams (5%) are a correct application of a technique. |

Fig. 3. Overview of the most significant results

Naives and novices behave as expected. We are a bit surprised by the number of static structure diagrams produced by naives, although most are of a low quality (surrogate models). Qualitative analysis shows that if naives use indirect mapping, they do because of no other choice. One of the eye-catching things is the 78% inconsistency rate in diagrams from the novices (naives produce 52% inconsistent diagrams). Another surprise is that, because of the nature of the task, there is not even a single indication of an STD-like diagram among those produced by naives.

4 Conclusion

As expected, novices do attempt to express their ideas in learned diagram techniques. That is to say, their diagrams often superficially resemble such techniques. However, although many techniques are deployed, almost all diagrams show (major) deviations from the technique, and all but one subjects produced inconsistent results. Naives did not show any sign of an existing diagram technique. We therefore conclude that indeed a mismatch between mental models and many diagram techniques can be assumed.

We started the research with the knowledge that diagram techniques are difficult to master. This doesn't mean that the effort isn't worth the result. But to improve the learnability (and readability by naives), the nature of mental models must be taken into account. For instance, information that takes little effort to encode is more likely to be a part of natural mental models. This explains the naives' preferences for types Im, La/label and direct mappings. We suggest first to pay attention to the currently used symbols.

REFERENCES