Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2006 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2006

Cognitive Load Reduction Through the use of Building Blocks in The Design of Decision Support Systems

Gwendolyn Kolfschoten
Delft University of Technology

Edwin Valentin
Delft University of Technology

Jan de Vreede
Delft University of Technology

Alexander Verbraeck
Delft University of Technology

Cognitive Load Reduction Through the use of Building Blocks in The Design of Decision Support Systems

Gwendolyn Kolfschoten,
Delft University of Technology
g.l.kolfschoten@tudelft.nl

Edwin Valentin,
Delft University of Technology,
Systems Navigator
edwin.valentin@systemsnavigator.com

Gert-Jan de Vreede,
University of Nebraska at Omaha,
Delft University of Technology
gdevreede@mail.unomaha.edu

Alexander Verbraeck,
Delft University of Technology,
University of Maryland
a.verbraeck@tudelft.nl

ABSTRACT

Processes and tasks in organizations become increasingly complex and dynamic. This requires managers of expert teams to quickly gain knowledge and insight outside their prime area of expertise. In these situations analysis tools and decision support tools are required. Often, such tools are used by experts to compose models that managers can use to gain specific insight in complex tasks and decisions. An observed paradox in this process is that once the first model is made, the insight into the system reveals the "real problem" and thus several iterations of the analysis, design and modeling are required to create a model that provides the required support. A proposed solution to increase the efficiency of re-designing is the use of patterns, also named building blocks. This allows the expert to re-use components to accommodate new requirements. However, the advantage of building blocks goes beyond re-use, design efficiency and flexibility. This paper argues that in addition to the benefits described above, there is a specific added value for the use of building blocks by novices to acquire analysis, modeling and design skills. We propose that building blocks decrease the cognitive load of both the design task and the effort of acquiring these skills. We use cognitive load theory from educational psychology to theoretically underpin this proposition. Empirical evidence is presented through two exploratory experiments.

Keywords

Building blocks, Cognitive Load, Design skills, Modeling skills, Expertise reversal effect.

INTRODUCTION

Organizations face the challenge of increasingly complex processes and tasks (Huber 1984). To deal with this complexity, organizations often need the expertise of several people to solve problems, make decisions, and accomplish tasks. Mintzberg describes 2 types of professional organizations, a professional bureaucracy and an adhocracy (Mintzberg 1983). In professional organizations, decisions need to be based on knowledge and expertise from different disciplines. Using the input of others, managers try to gain enough knowledge and understanding to make decisions or perform tasks outside their prime area of expertise.

Combining the information and expertise of people is necessary since decisions are often too complex for one individual to understand all implications. To support this process, Decision Support Systems (DSS) are available to offer decision support such as for example brainstorming, simulation, information systems and business games, methods where increased insight is gained and information is retrieved that is not readily available in an explicit form. DSS are "interactive computer-based systems that help decision makers utilize data and models to solve unstructured problems (Sprague and Carlson 1982, p. 4.)." Most of these DSS require support by experts who can analyze and model the current situation and future scenario's to eventually implement these with advanced technology, to build a decision support tool. Naturally this involves a costly investment.

An additional problem with the use of these methods is that once the models are used by the expert to provide answers, the insights will reveal additional questions, for which a second model is required. For example, during the design of a highly innovative underground transportation system, each outcome of a simulation experiment generated new insights and ideas which led to new requirements to the vehicles, the control system and management layers (Heijden, Harten, Ebben, Saanen, Valentin and Verbraeck 2002). Each requirement had to be analyzed and had to be modeled again, which eventually increased the project time and delayed the decision making by the problem owners.

This effect can be expected if we consider that the reason for the use of such methods lies in the complexity of the system and the lack of insight. Once this insight is gained, new questions are revealed, and a second iteration is required to offer the information and insights required to make the decision. The problem with these situations is often that the initial model does not have the capabilities to offer the required insights, and in many situations, initial design or modeling choices make it difficult to adjust the model, and consequently these iterations are frustrating and costly.

Many solutions have been proposed to better manage and control these iterative modeling and design cycles (Alexander, Ishikawa, Silverstein, Jacobson, Fiksdahl-King and Angel 1977; Boehm 1988; Gamma, Helm, Johnson and Vlissides 1995). One approach is the use of building blocks; components that can be used to build models, making the design effort faster, the resulting system more flexible and at the same time offer a valuable library of best practices to advance the expertise and research in the field. A specific context in which such solution is valuable is the development of DSS.

This paper discusses the role of building blocks in the training of novice designers and model builders. We propose that building blocks do not only enable efficiency and flexibility of the design effort for novices, building blocks also increase their understanding of the design process and decrease the cognitive load of acquiring analysis, modeling and design skills to build DSS. We will therefore analyze the building block concept in the light of Cognitive Load Theory (CLT) (Sweller 1988). CLT explains how information can be offered to users in a way that allows them to optimally use the capacity of the human brain for learning and comprehension. The remainder of this paper will first explain the Building Block (BB) concept and CLT. Secondly, we present a position on how the use of BBs decreases cognitive load and enhances fast understanding. Next, we present the result of two exploratory experiments in which the effect of BBs on modeling skills is analyzed. We use these results to evaluate our propositions and recommend further research.

BUILDING BLOCKS FOR MODELING

Building blocks are design components that are used to easily configure a system or model within a certain domain. In this paper we refer to a variety of models such as models to analyze a current situation or a scenario, games to test scenarios, and process models of collaboration support to efficiently acquire group results. The development of some of these models is supported by available suites consisting of components to build these models. For example, a generic simulation environment offers components to build a simulation model and a Group Support System (GSS) offers tools to design a collaboration process. However, configuration of these environments and systems are complex and require expert skills.

An alternative way of using these environments and systems is to configure components for a specific purpose or apply the elements of these environments and systems according to a specific recipe. Such recipe is a design pattern. A design pattern is a reusable solution to a recurring problem in a specific domain. Patterns can be combined in a pattern language, in which the combined patterns do not only offer a library of solutions but also a vision or worldview of the domain (Alexander et al. 1977). In modeling, we use the term Building Block for a specific type of patterns. Verbraeck et al (Verbraeck, Saanen, Stojanovic, Shishkov, Meijer, Valentin and van der Meer 2002) defined *a building block as a self-contained, interoperable, reusable and replaceable unit, encapsulating its internal structure and providing useful services or functionality to its environment through precisely defined interfaces.* Example use of building blocks for modeling are the process model of a brainstorm session, a simulation model or a multi-actor game, we will present two types of building blocks to illustrate our propositions; thinkLets (building blocks for collaboration processes) and Simulation Building Blocks.

ThinkLets are building blocks for the facilitation of collaboration processes. Facilitating collaborative work processes and operating Groups Support Systems is a professional skill that requires training and practice. Especially the design of effective and efficient collaboration processes is a challenging task (Clawson and Bostrom 1995). A thinkLet is the smallest unit of intellectual capital to create a pattern of collaboration (Vreede and Briggs 2005). Examples of thinkLets are various brainstorming techniques, group voting methods, consensus building techniques etc. Each thinkLet creates a unique pattern of collaboration; that is a unique interaction among group members to achieve their goal. A thinkLet provides a reusable, transferable, predictable, and documented facilitation technique (Vreede and Briggs 2005). Currently, expert facilitators have documented over 50 thinkLets to constitute a building block library for the design of collaboration processes. Novice facilitators can use this library to design and facilitate collaboration processes.

Simulation building blocks (Verbraeck et al. 2002) are building blocks to enhance the development of simulation models in a certain domain. A simulation building block is a composition of simulation logic that provides a configurable domain specific representation. An example of a simulation building block for the simulation of a transportation system is a configurable "vehicle" or a configurable "station", with which different traffic delays and capacity problems can be simulated. Simulation experts develop sets of simulation building blocks for a specific domain and practitioners in these domains can compose simulation models of their system without the need to become an expert in the underlying simulation environment or statistical analysis.

COGNITIVE LOAD THEORY

Cognitive load can be defined as *the cognitive effort made by a person to understand and perform his task* It has both a task-based dimension (mental load) and a person-based dimension (mental effort) (Sweller, Merrienboer and Paas 1998). Cognitive load theory (CLT) is based on the assumption that our short-term or working memory is limited to seven plus or minus two information elements (Miller 1956). This is the information that we can process at a certain moment.

Besides working memory the model assumes that we have a long-term memory in which information is stored, in so called schemas[1] (Sweller 1988). To learn we need to consciously combine individual elements of information to build schemas. Schemas can be handled by our working memory as an individual component. The schema is not just a storage frame; information in the schema can be retrieved unconsciously. An example of this is reading. Experienced readers for instance do not process every character they read anymore; they recognize entire words, or even parts of sentences (Sweller et al. 1998). Therefore, the larger the schema, the more information we can process in the same time in our working memory, the faster we can gain new understanding and combine schemas to find solutions or answers to problems.

The availability of schemas determines the difference between experts and novices in several ways (Sweller 1988): An expert, compared to a novice does not have more schemas, but larger schemas. A second difference is that an expert recognizes patterns of problems from previous experience, and combined these in his schema with solution-directions, while novices do not possess such schema and thus have to solve the problem from scratch. This lack of sophisticated schemas causes another difference between novices and experts. Experts categorize their knowledge based on different solution modes, while novices do not yet see the direct relation between problems and solutions, they can only structure their schemas based on surface structures such as shared objects.

The cognitive load theory explains how we use our cognitive capacity to construct schemas. There are 3 types of cognitive load (Sweller 1988):

- Intrinsic cognitive load, is the cognitive load that is inherent to the task, and that is defined by the intrinsic task complexity.
- Extraneous cognitive load, is the cognitive load caused by the presentation and transition method of the information.
- Germane cognitive load, is the cognitive load when the working memory is used to build the schemas and store them in the long term memory.

Reduction of the intrinsic or extraneous cognitive load makes capacity available within the working memory to increase the germane cognitive load and thus make it possible to build schemas to store the information in the long term memory; which constitutes learning and understanding. CLT offers different methods to reduce extraneous cognitive load such as offering parsimonious information elements and avoiding split attention that is caused by disintegrated information such as a picture with a separate description (Sweller et al. 1998). In addition Pollock et al (2002) suggest that the intrinsic cognitive load can be reduced for complex systems that are difficult to understand even with very low extraneous cognitive load. They suggest offering a basic framework that can be schematized and in which interaction between information elements is removed. These schemas can than be used as a basis to learn the other material by offering the complete information with the interaction as a second step in the learning process.

Cognitive load can thus be reduced, leaving more memory space for germane cognitive load; the building of schemas. However, when these schemas are already available and automated, as is the case in the mind of an expert, the methods tend to have a reverse effect (Kalyuga, Ayres, Chandler and Sweller 2003). For instance, offering explanation to the meaning of different shapes in a model will help a novice in modeling, but for an expert who already understands the meaning of the shapes, this is redundant information, and trying to ignore it or accidentally reading it will be redundant and thus increase

[1] The plural of schema is schemata, but we will stick to the language used by Sweller et al.

cognitive load. This is called the expertise reversal effect; methods to reduce cognitive load for novices can increase cognitive load for experts and thus design support and modeling support should be different for experts and novices.

Developing models to support decision making is a complex task. It requires the understanding of many elements and relations (intrinsic cognitive load), designs are often represented in "coding" such as a modeling language (extraneous cognitive load), and besides building initial schema of these concepts (germane cognitive load), it also requires creativity of the designer (additional germane cognitive load). Offering a supportive method for this design effort can reduce cognitive load of the design task, but should be aimed consciously on either novices or experts to avoid a counter effect as described by Kalyuga et al (Kalyuga et al. 2003). In the next section we will discuss how the use of building blocks affects the cognitive load of the design and modeling effort.

THE EFFECT OF BUILDING BLOCKS ON COGNITIVE LOAD OF EXPERT AND NOVICE DESIGNERS

Building blocks offer ready made solutions for frequently recurring problems. A process often used for design or problem solving exists generally of several steps that include identification of the issue, analysis, finding (and evaluating) alternatives, choice and implementation (Ackoff 1978; Checkland 1981; Couger 1995; Drucker 1967). The use of building blocks changes this approach. Where a general design approach requires finding and evaluating alternative solutions, a design process with building blocks requires only the choice or reconfiguration of building blocks, which can be easily combined and customized for implementation. This eliminates a complex and challenging step from the design process and it makes the design more flexible, as building blocks can be replaced or re-configured to adapt the design to new requirements. It is therefore expected, that building blocks quicken the design process, and especially quicken the effort of altering a model or design.

Besides this efficiency effect, building blocks are expected to have an effect on cognitive load. The effect of building blocks on extraneous cognitive load is not determined by the building block concept or the approach. Building blocks can be represented as (software) components or descriptions (recipes), and in some domains they might even be offered as physical components. However, building blocks are coherent components, and thus offer a good basis to integrate documentation and offer a parsimonious component to avoid the split attention effect (Sweller et al. 1998).

For intrinsic cognitive load we see a large parallel with the effect described and tested by Pollock et al. (Pollock et al. 2002). Pollock describes an approach to teach students information that is too complex to understand at once. Such information has such a high element interactivity (related information elements that cause complexity and are thus more difficult to learn and understand) that too many concepts should be held in the working memory at once to understand the concept. With no prior schema of the information, trying to understand it is very difficult. The student will have to build schemas without having an overview of the concept he is to learn. The approach prescribes to offer the information in smaller steps. For instance, the student first learns how to perform a certain task, without learning the explanation of why it is performed that way. Once this is understood and captured in a schema, he can step by step learn to understand the logic and reasoning behind the approach.

The building block approach does essentially the same. It divides a complex system in recognizable components, and first explains the designer how he can combine and use these components to represent the system. After this is understood, the designer can (if needed) further learn to understand why they work and what the logic behind these building blocks is.

Pollock found convincing evidence that this approach, named isolated-interacting elements approach had a significant effect on novice students, while it did not have an effect on more experienced students (note, no negative effect was found for these students) who, most likely already possess some schemas of the information, and therefore do not need to study the separate components. We therefore offer the following propositions:

P (1a): Novices will faster gain understanding in modeling and design skills, when they learn to design with the building block approach first, before they learn to understand entire systems or models.

P (1b): Experienced (domain experts or more experienced designers) will not gain faster understanding of modeling and design skills with the use of building blocks.

The last type of cognitive load, germane cognitive load is to be stimulated. More germane cognitive load is better, as it constitutes learning. However, Sweller, (Sweller et al. 1998) in his explanation about schema construction, raises an interesting conclusion. Schemas have two purposes: the storage and organization of information, and the reduction of working memory load. Through automation, larger schemas can be used in the working memory as one component. Experts in complex tasks such as chess and physics do not have better problem solving skills, rather they have a large set of larger, more complex schemas in their long term memory, that represent patterns of problems and related solutions. With these patterns available they can recognize the problem and pick the right solution (Sweller 1988; Sweller et al. 1998). Chi, Glaser and Rees (Chi, Glaser and Rees 1982) discovered that experts categorize problems based on the solution type, while novices categorize problems based on surface structures such as shared objects (all problems related to airplanes). This could indicate that the schema that novices build initially when they learn "problem solving from scratch", are not efficient. The use of building blocks, which are in fact large solution based schemas, might thus offer the novices a framework to build better schemas from the beginning of their learning process. This leads us to the following propositions:

P (2a) Training novices with the use of building blocks will increase the quality of the schemas they build to represent a system.

P (2b) Training novices without the use of building blocks will cause them to build (in a similar time frame) lower quality schemas to represent a system.

In the following section we will describe two experiments in which novices (with different experience levels) and experts design decision support with and without the use of building blocks.

EXPERIMENTS

ThinkLet use in facilitation training

One of the main tasks of a facilitator is to design the collaboration process. In this experiment we asked novices with different amounts of training and experience to design collaboration processes using a thinkLet library. In this experiment[2] two groups of undergraduate students from Delft University of Technology, (n8 and n5) 12 novice designers in total, (one student participated in both groups) were divided in 3 experience categories based on the number of training hours they received and their experience in facilitation and GSS use. Where the novices only got an introduction about GSS and facilitation, the experienced students had actually experienced what it is to facilitate a collaboration process.

- Level 1: 2-4 hours, no facilitation experience, some with minor technical facilitation experience.
- Level 2: 8-10 hours, some GSS experience, but no facilitation experience.
- Level 3: approximately 20 hours, limited facilitation experience.

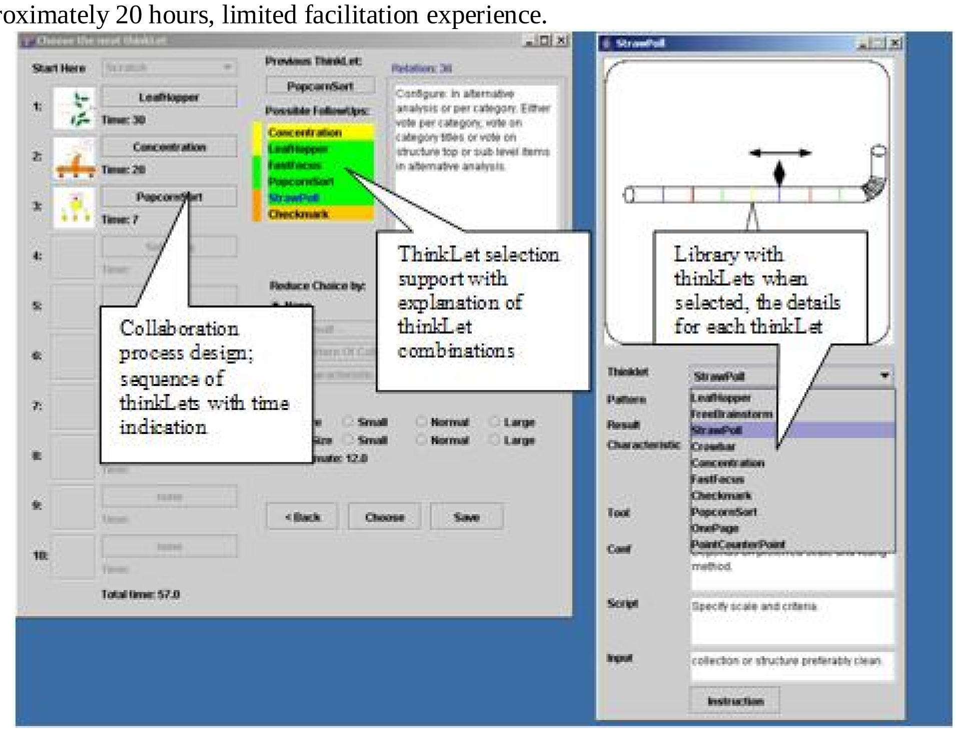

Figure 1. ThinkLet building block library with selection support.

[2] Published also in Kolfschoten, G.L., and Veen, W. (2005) Tool Support for GSS Session Design, *Proceedings of the Hawaii International Conference on System Sciences*, Los Alamitos, IEEE Computer Society Press.

The students participated in a full day workshop in which they had to design 3 collaboration processes based on a case description. The case descriptions, based on real sessions, contained a goal statement, a group description, and the contours of the task. The students were supported in the design effort with a tool that offered them support in the choice among ten thinkLets (collaboration process building blocks). The resulting process model had to specify all information required to facilitate the collaboration process. Despite the small number of students we made some interesting observations.

We measured the quality of design on a 1 to 10 scale. Two teachers of a facilitation class, both experienced facilitators, assessed the quality of the designs, and conflicts in assessment were resolved. Additionally the time spent on each design was measured. Table 1 indicates the quality of the first, second and sometimes third process model (not all students made all 3 designs due to lack of time) made by novices with different training history. Interesting is that the novices get up to speed and outperform more experienced students. The latter already adapted their own design approach, and found it hard to use the new approach. The efficiency is indicated by the time spent for a design. The time spent on design does decrease over time only for more experienced novices. The novices with low training time did not get faster in their design. A general decrease in design time could have been caused by the fact that motivation and energy were lower at the end of the day.

	level 3		*level 2*		*level 1*	
Design	Quality	n	Quality	n	Quality	n
1	6.5	6	6.3	2	5.6	3
2	5.9	6	6.7	2	5.7	5
3	6.2	4	6.9	1	7.3	2

	level 3		*level 2*		*level 1*	
Design	Time	n	Time	n	Time	n
1	79	6	94	2	108	5
2	83	6	86	2	70	5
3	38	4	60	1	105	2

Table 1 design quality (1-10 scale) and design time (in minutes) per experience level for design with building blocks.

Building block experiment

In 2004 a range of laboratory experiments was performed (Valentin, Verbraeck and Sol 2003) by novices in simulation (undergraduate students from the Delft University of Technology) and simulation experts (simulation consultants of the company Rockwell Software, daily working with the simulation environment Arena).

The participants in the laboratory experiment received an individual assignment to develop a simulation model for a public transport system within 8 hours. Half of the participants received a set of simulation building blocks that were developed for modeling of public transportation systems and the other half had to use the basic modeling constructs of the Arena simulation environment. Fig. 2 shows the building block environment used. Afterwards the result of the novices and the experts was judged by professional simulation experts that have been involved in simulation studies in the public transportation sector for years as well as people who have acted as problem owners in these systems.

The novices in simulation that worked with the generic simulation environment worked hard, but were miles away from a complete solution. Most of these novices used a structured way of working and first focused on the movement of trains, before introducing passengers that use the trains. However, these novices spent a lot of time searching through the help files and evaluating different alternatives for modeling the system. Therefore, the novices spent most of their time in learning how to use the simulation environment, instead of developing a simulation models. The experts that used the generic simulation environment succeeded in developing models, but they got lost in details. The part of the simulation model that they had working was fine, but their model was not yet fully working and they would need a couple of hours more to finish the model.

The developers (both experts and novices) that worked with the simulation building blocks did not have any difficulty in developing a simulation model. Within only two hours they had simulation models that seemed to represent the system, but the models did not work as intended. The novices expected that they made mistakes. They started to make changes to the configuration of the building blocks and searched in the documentation of the building blocks to solve the problems and in the end, provided a model that offered sufficient insight for decision making. The experts worked in a different way. When they noticed problems, they expected that the building blocks were incorrect. They knew the simulation environment, and the things they noticed in the simulation building blocks collided with their existing schemas of simulation models and their working. The experts tried to understand the technology behind the simulation building blocks and once they noticed the structure and logic underneath, they started to trust and understand the simulation building blocks and noticed that the mistake was due to their parameterization of the simulation building blocks and not due to faulty building blocks. The experts succeeded in correcting their simulation model and performed a couple of simulation experiments.

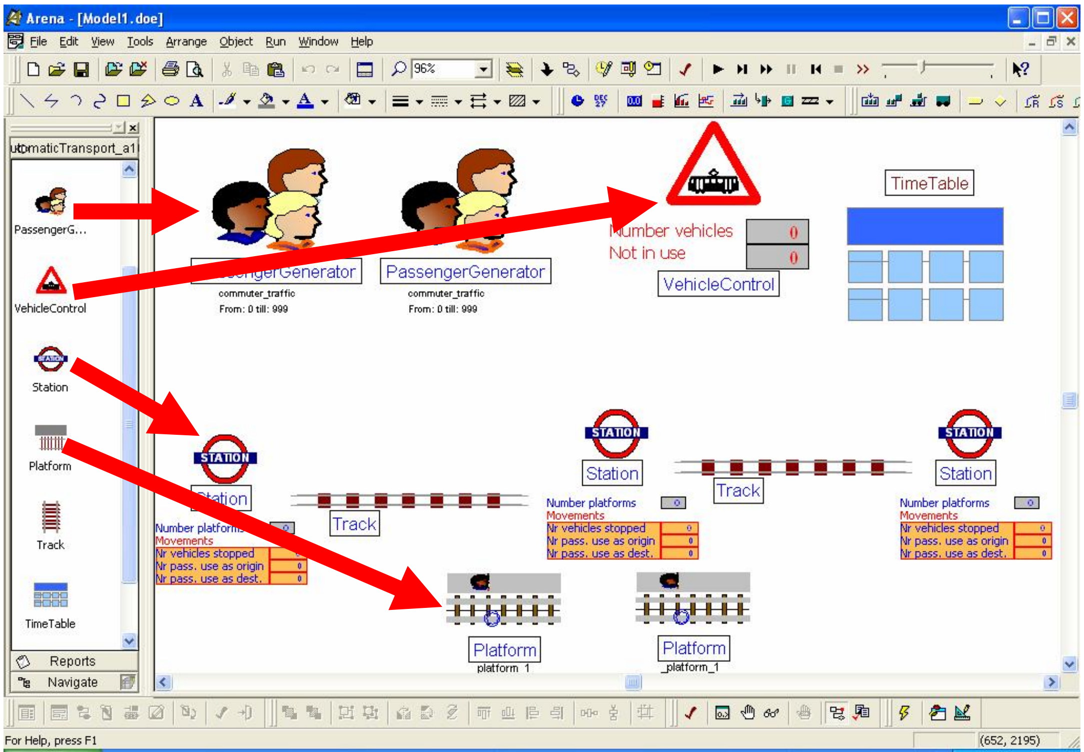

Fig. 2: Simulation building blocks

CONCLUSIONS AND RECOMMENDATIONS

In both of these illustrative experiments we saw the effect of building blocks on the cognitive load of the design and modeling effort, and of acquiring these skills. We saw a clear difference in the effect of the building blocks on experts and on novices. In this section we will further elaborate on this difference.

Experts and novices make simulation models in a different way. Experts recognize a system and are capable of linking the knowledge they have from the system to the modules available in a simulation environment. Novices are less aware of the capabilities of the modules of a simulation environment and thus have more difficulty to create a simulation model. We noticed the same effect with the thinkLets. Experienced students complained that they had to find the building block that offered them the tools and methods they were used to apply, while novices in the end provided better models with the use of the building blocks than the experienced users. This seems to corroborate with our first proposition; building blocks help novices to faster gain understanding in modeling and design skills, while experts felt disturbed and disrupted in their effort by the building blocks.

Our second proposition stated that novices would develop higher quality schemas. Or first experiments suggest that the models made by novices were of higher quality than those of the experts, while the time they spend on working with and learning about the modeling approach and the decision support systems was significantly shorter. This leads us to the tentative conclusion that the use of building blocks does not only affect the efficiency of the design and modeling effort, it also constitutes ***learning efficiency of novices*** to designing and modeling skills and it enhances the quality of their design.

Future research should further analyze the added value of the use of building blocks in the transition of complex skills, such as the development of DSS, but also in other complex design disciplines. Increased understanding and improvement of the learning efficiency effect of building blocks could allow for the building of design and modeling support suites that allow novices to these disciplines to model and design their own decision support systems and this would make customized decision support available for a larger audience of managers and professionals in organizations without the need for large investments in training and the acquisition of external expertise. Additionally it might be interesting to look into the transition of such design approach to experts, which would require a technique to replace or alter an existing schemas construction to build one that is more efficient.

REFERENCES

1. Ackoff, R.L. (1978) The Art of Problem Solving, John Wiley & Sons.
2. Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-King, I, and Angel, S. (1977) A Pattern Language, Towns, Buildings, Construction, Oxford University Press, New York.
3. Boehm, B. (1988) A Spiral Model of Software Development and Enhancement, *IEEE Computer*, 21, 5, 61-72.
4. Checkland, P.B. (1981) Systems Thinking, Systems Practice, John Wiley & Sons, Chichester.
5. Chi, M., Glaser, R., and Rees, E. (1982) Expertise in Problem Solving, in Sternberg (Ed.) Advances in the Psychology of Human Intelligence, Erlbaum, New Jersy.
6. Clawson, V.K., and Bostrom, R.P. (1995) The Importance of Facilitator Role Behaviors in Different Face to Face Group Support Systems Environments, *Proceedings of the Hawaii International Conference on System Sciences*, Los Alamitos, IEEE Computer Society Press, 181-190.
7. Couger, J. D. (1995) Creative Problem Solving and Opportunity Finding, Danvers, Mass: Boyd And Fraser.
8. Drucker, P.F. (1967) The Effective Executive, William Heinemann Ltd, London.
9. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995) Elements of Reusable Object-Oriented Software, Addison-Wesley Publishing Company.
10. Heijden, M.C. van der, Harten, A. van, Ebben, M.J.R., Saanen, Y.A., Valentin, E.C., and Verbraeck, A. (2002) Using Simulation to Design an Automated Underground System for Transporting Freight Around Schiphol Airport, *Interfaces*, 32, 4, 1-19.
11. Huber, G.P. (1984) The Nature and Design of Post-Industrial Organizations, *Management Science*, 30, 8, 928-951.
12. Kalyuga, S., Ayres, P., Chandler, P., and Sweller, J. (2003) The Expertise Reversal Effect, *Educational Psychologist*, 38, 1, 23-31.
13. Kolfschoten, G.L., and Veen, W. (2005) Tool Support for GSS Session Design, *Proceedings of the Hawaii International Conference on System Sciences*, Los Alamitos, IEEE Computer Society Press.
14. Miller, G.A. (1956) The magical number seven plus or minus two: some limits on our capacity for processing information, *Psychological Review*, 63, 81-97.
15. Mintzberg, H. (1983) Structure in Fives: Designing Effective Organizations, Prentice-Hall Inc., Englewood Cliffs.
16. Pollock, E., Chandler, P., and Sweller, J. (2002) Assimilating Complex Information, *Learning and Instruction*, 12, 61-86.
17. Sprague, R.H., and Carlson, E. (1982) Building Effective Decision Support Systems, Prentice-Hall, Englewood Cliffs.
18. Sweller, J. (1988) Cognitive load during problem solving: Effects on learning, *Cognitive Science*, 12, 257-285.
19. Sweller, J., Merrienboer, J.G. van, and Paas, F.G.W.C. (1998) Cognitive Architecture and Instructional Design, *Educational Psychology Review*, 10, 3, 251-296.
20. Valentin, E.C., Verbraeck, A., and Sol, H.G. (2003) Effect of Simulation Building Blocks on Simulation Model Development, *Proceedings of the International Conference of Technology, Policy and Innovation*, 54-61.
21. Verbraeck, A., Saanen, Y., Stojanovic, Z., Shishkov, B., Meijer, A., Valentin, E., and van der Meer, K. (2002) What are building blocks?, in A. Verbraeck and A.N.W. Dahanayake (Eds.) Building blocks for Effective Telematics Application Development and Evaluation, Delft University of Technology, Delft, 8-21.
22. Vreede, G.J. de, and Briggs, R.O. (2005) Collaboration Engineering: Designing Repeatable Processes for High-Value Collaborative Tasks, *Proceedings of the Hawaii International Conference on System Science*, Los Alamitos, IEEE Computer Society Press.