HUMAN TECHNOLOGY

An Interdisciplinary Journal on Humans in ICT Environments ISSN: 1795-6889

www.humantechnology.jyu.fi Volume 2 (2), October 2006, 202–224

CREATING A FRAMEWORK FOR IMPROVING THE LEARNABILITY OF A COMPLEX SYSTEM

Minttu Linja-aho
Tekla Corporation
Finland

Abstract: *When designing complex systems, it is crucial but challenging to make them easy to learn. In this paper, a framework for improving the learnability of a complex system is presented. A classification of factors affecting the learnability of a building modeling system as well as guidelines that refine the factors into practical ways of action are introduced. The factors and guidelines include issues related to the user interface, conformity of the system to user's expectations, and training. The classification is based on empirical research during which learnability was assessed with several methods. The methodology and the classification of learnability factors can be used as references when analyzing and improving the learnability of other systems. System developers and training providers can utilize these guidelines when striving to make systems easier to learn.*

Keywords: *learnability, ease-of-learning, complex systems, grounded theory, guidelines.*

INTRODUCTION

As complex systems get more and more common in various problem domains, it becomes necessary to make them easily learnable. Good learnability will lead to acceptable learning times, sufficient productivity during the learning phase, and greater satisfaction in new users. However, designing complex systems that are easy to learn is challenging. Complex systems need to provide a wide variety of functionality and to support complex task flows and object structures. There is a danger of complexity leading to long and unproductive learning times.

Another challenge with improving the learnability of complex systems is that the changes made in the system must not decrease the efficiency of use (Santos & Badre, 1995). It has been discussed whether learnability and efficiency actually support each other or rather, in fact, contradict. Several studies have indicated that learnability and efficiency are congruent. Whiteside, Jones, Levy, and Wixon (1985), for example, noticed in their study concerning several command, menu, and iconic interfaces that the best system for novice users was also the best for expert users, and the worst system for novices was the worst for experts.

However, some researchers (e.g., Goodwin, 1987) have pointed out that experts and novices may have different requirements for a system: Abbreviations and shortcuts, for example,

URN:NBN:fi:jyu-2006519

will improve the performance of experts but may slow down the learning of novices. Thus, balancing learnability and efficiency requires careful consideration.

In any case, novices are an important user group and therefore the learning dimension should be taken into account when designing a system. Compacting the learning process and reducing the length of training needed and the number of problems that new users face will save costs for the organization that has taken the system into use and, in many cases, the system provider as well. If users consider the system easy to learn, they are more likely to pass through the learning stage and continue using the system regularly. Satisfied learners may also tell other prospective users about an easily learned system and thus perform efficient peer-to-peer marketing.

To improve the learnability of a system, a general understanding of the factors affecting learnability is needed. In this paper, a classification of learnability factors related to a building modeling system is introduced. Practical guidelines that can be used by product developers who design new systems or redesign existing ones are presented as well. I believe that the classification of factors and the guidelines are useful for developing complex systems that are easy to learn.

LEARNABILITY

In this article, the word *learnability* signifies how quickly and comfortably a new user can begin efficient and error-free interaction with the system, particularly when he or she is starting to use the system. It can be seen from this definition that both objective and subjective facets of learnability are considered: the speed of learning (quickly) and the subjective satisfaction of the learner (comfortably). The goal of the learning process is efficient and error-free interaction. In the literature, the terms *ease-of-learning* and learnability often have been used interchangeably.

Multiple other definitions for learnability exist in the literature, and they differ from each other slightly. For example, Bevan and Macleod’s (1994) definition of learnability comprises the usability attributes of satisfaction, effectiveness, and efficiency that are evaluated within a certain context, namely the context of a new user. In the ISO 9241 standard (International Organization for Standardization [ISO], 1998a and 1998b), learnability is also defined through the three attributes of efficiency, effectiveness, and satisfaction. Dix, Finlay, Abowd, and Beale (1998) define learnability as the ease with which new users can begin effective interaction and achieve maximal performance. In summary, what most of the definitions have in common is that they address the initial usage experience and include a criterion such as effectiveness or efficiency that can be used to measure the learning results. In addition, some researchers have emphasized that the term *learnability* should also cover expert users’ ability to learn functions that are new to them (Sinkkonen, 2000). While this perspective is important, I considered it feasible to concentrate on one group of users, namely new users, in this research.

The importance of learnability in determining system acceptability has been noticed early (e.g., Butler, 1985). Lin, Choong, and Salvendy (1997) found that learnability is correlated with user satisfaction. The learnability of complex systems is especially critical, as the complexity tends to make the unproductive learning period longer than what is desired by the user and the managers in the organization.

The Relationship of Learnability and Usability

There are contradicting views of how learnability relates to usability. Some researchers consider learnability to be a subconcept of usability (e.g., Elliott, Jones, & Barker, 2002). Nielsen (1993) presents five subattributes of usability: learnability, efficiency, memorability, errors, and satisfaction. In the same book, Nielsen presents 10 usability heuristics that should be considered when designing user interfaces. Dix et al. (1998) in turn divide usability into the three attributes of learnability, flexibility, and robustness. Lin et al. (1997) list eight attributes: compatibility, consistency, flexibility, learnability, minimal action, minimal memory load, perceptual limitation, and user guidance.

Elliott et al. (2002) have discussed the relationship of learnability and usability in their publication. They refer to several studies indicating that the concepts of learnability and usability are strongly related and even congruent. Roberts & Moran (1983), for example, found that procedural complexity underlies both the performance of experts and the learning of novices. Whiteside et al. (1985) have also stated that the concepts of usability and learnability are congruent. Based on these studies, Elliott et al. (2002) made the conclusion that elements from models for usability can be incorporated to models of learnability as well. However, other researchers (e.g., Paymans, Lindenberg, & Neerincx, 2004) have noted that sometimes learnability and usability may be contradictory: that issues that improve learnability actually reduce usability. This is related to the question of how learnability and efficiency relate to each other, which I discussed earlier in this article.

Based on the literature review and my experiences, I expected learnability and usability to have several issues in common. However, I expected during the study that I would also notice issues that affect learnability but are not included in common models of usability. I will discuss the relationship of learnability and usability later in this article after presenting the empirical results.

Aspects of Learnability

Learnability studies have often concentrated on the effect of the user interface design on learnability (see Elliott et al., 2002; Lin et al., 1997). Naturally, the user interface is crucial for learnability, as it essentially forms the link between the user and the system. Different researchers stress various issues as determinants of user interface learnability. Rieman, Lewis, Young, and Polson (1994) emphasize the effect of consistency. Green and Eklundh (2003) in turn emphasize the naturalness of interaction. Dix et al. (1998) have presented five principles that support user interface learnability: predictability, synthesizability, familiarity, generalizability, and consistency. Elliott et al. (2002) found four factors that determine the learnability of a system: transparency of operation, transparency of purpose, accommodation of the user, and the sense of accomplishment. The two first elements are determined by the user interface design and, according to Elliott et al., (2002), the accommodation of the user and the sense of accomplishment follow them causally.

Applying these principles to user interface design helps in designing systems that are easy to learn. However, to improve learnability, the correspondence between the system and users' expectations must be analyzed too, as expectations have a remarkable effect on learning. Users' expectations may cover the scope, underlying concepts, and basic functionality of the system.

Kellogg and Breen (1987) among others, have stated that differences between the users' expectations and the actual system can cause learning difficulties. I decided to use the theory of mental models as a basis for analyzing these differences.

Mental models are internal representations of entities with which we interact. According to Fein, Olson, and Olson (1993), a mental model of a computerized system may contain information on the system functionality, components of the system, related processes, and their interrelations. Fein et al. (1993) write that learning can be viewed as a process in which the user processes information and thereby his or her mental model is changed. According to Shayo and Olfman (1998), a user's mental model helps him or her to plan how to interact with the system, interpret the behavior of the system, and perform correctly when problems occur.

As the goal of this study was to provide tools to make the learning process faster, I needed to analyze the entire learning process, from the first interaction with the system, through the training process, and into the post-training phase. In this study, I paid special attention to the training arrangements, as changes in training are a rather quick and easy way to improve learnability. To analyze the effect of training, it is useful to know something about the human learning process and different learning theories.

Multiple theories of learning exist, developed by different schools of scientists. The current HCI (human-computer interaction) research has tended to adopt a cognitive perspective on learning (Elliott et al., 2002). Cognitive theorists stress the importance of internal thought processes and mental structures, as opposed to behavioral scientists' emphasis on behavioral patterns, reinforcement, and conditioning. In this study, I adopted a cognitive perspective to learning and adjusted it with the ideas presented by constructivists. Constructivism is based on cognitive science, and cognitive scientists and constructivists see learning rather similarly. According to constructivists, learning can be defined as a process of building and reorganizing mental structures. Constructivism also states that knowledge is never independent of the learner and the learning context. The learner combines new information with his or her existing knowledge to form a more accurate model of the subject (Marton & Booth, 1997). This view of learning is closely related to the theory of mental models, as both stress the importance of changes in a human's internal knowledge structures. I saw the constructivist learning theory combined with the concept of mental models as a good basis for analyzing the learning process and the effect of training.

In this study, I concentrated on analyzing the three aspects of learnability that were mentioned above: user interface design; differences between users' expectations and the system, which can be analyzed through the theory of mental models; and the effect of training on the learning process. These aspects are later referred to as user interface, conformity to user's expectations, and training. Figure 1 illustrates this approach to learnability.

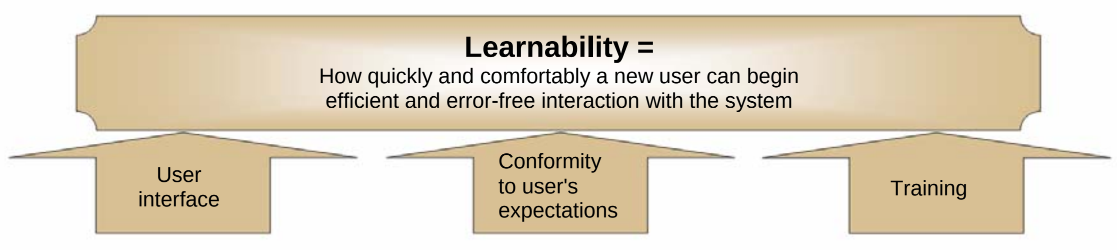

Figure 1. A definition of learnability and the aspects addressed in this study.

THE BUILDING MODELING SYSTEM

In this study, I analyzed the learnability of the Tekla Structures program, a building modeling system that has been developed by the Tekla Corporation. The primary users of the Tekla Structures system are structural engineers. With Tekla Structures, structural engineers can create a three-dimensional model of steel and concrete parts, connections, and other details of a building. Structural analysis can be done using the information contained in the model. The system is very complex in that it provides a wide selection of functionality and supports complex task flows and object structures. My expectation in undertaking a case study on the Tekla Structures program is that it would provide information that could be used to improve the learnability of this particular system as well as be used as a reference when improving the learnability of other complex systems. A typical user interface state of the Tekla Structures system is shown in Figure 2.

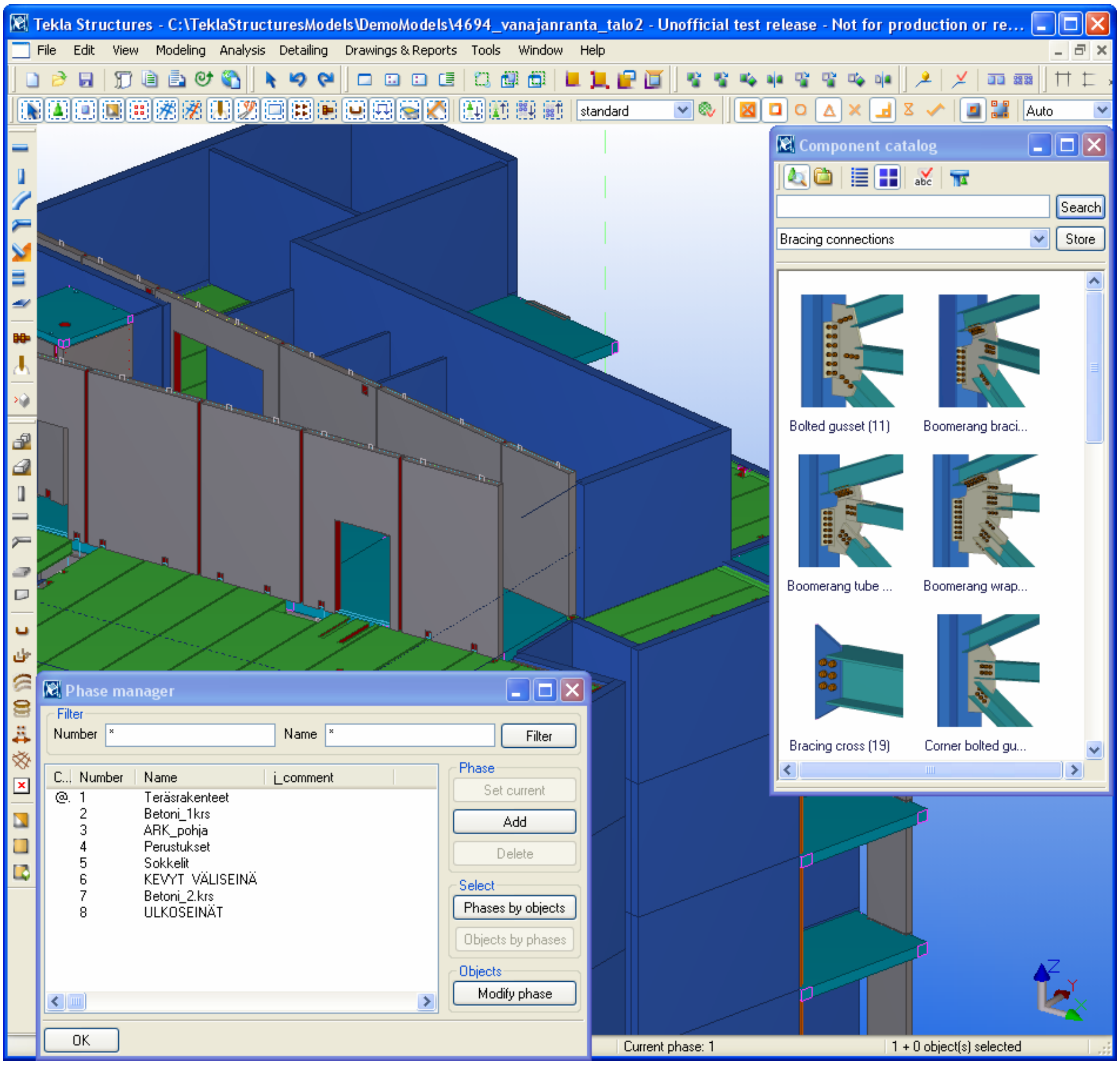

Figure 2. User interface of the Tekla Structures system. (Model by Antti Pekkala, A-Insinöörit, 2003)

To support learning, the Tekla Corporation organizes a three-day training course. However, because of the complexity of the system, only a small subset of its features can be addressed in the training and the learning period continues after the formal training. Improving learnability would result in a desired reduction in the learning time.

The training course was a good opportunity to observe the beginning of the learning process. I also observed and interviewed users before and after the training. I describe these research activities in the following section.

RESEARCH METHODS

The purpose of the empirical learnability research was to identify the factors that affect the learnability of the Tekla Structures system and to develop ways to improve learnability. This research was spread over a 3-month period in order to obtain information on different phases of the learning process.

Six novice users who had an engineering or technical drawing background were chosen as subjects. Two of them had worked in the building-modeling domain for only a few months, two of them for about 2 years, and two of them for more than 20 years. All of them had some experience with CAD (computer-aided design) systems but five of them had no experience with Tekla Structures and one of them had tried the system for only a day.

Four research methods were used at different phases of this study in order to collect versatile information and to capture as many different issues affecting learnability as possible. The four research methods are presented in the following sections. The choice of the research methods was highly dependent on the definition of learnability presented in the beginning of this article. I wanted to address both the objective and subjective facets of learnability and to observe how efficient and error-free the users could be in performing tasks with the system in each learning phase.

Pre-Training Interviews

The purpose of this research method was to acquire information on the mental models that users had before interacting with the system. This information is useful because differences between users' mental models and the system may explain learning difficulties.

An interview method similar to the one employed in this study was used by Dykstra-Erickson and Curbow (1997). They studied the learnability of a document management platform called OpenDoc. In the interviews that they conducted, they asked users to comment on user interface prototypes. Their goal was to address users' expectations on how to use certain system features.

In this study, the six subjects were interviewed individually and in-person for about 45 minutes. Interviews were conducted during a two-week period before the training. During the interviews, the user interface of the Tekla Structures system was shown to the users and questions were asked about the user interface elements. Subjects were also asked how they expected certain basic modeling tasks to be performed. They were allowed to test some procedures briefly with the system and comment on them. Interview questions included, as a sample, the following:

- Which icons do you find familiar? What do you think the others represent?

- Which do you expect to be the biggest differences between this system and the software you used before?
- How would you start creating columns and beams?
- How do you think you can copy and mirror elements?
- How do you expect changes in the model to affect drawings?

The interviews were audio recorded. The comments were transcribed to a written form after the interview. The interview language was Finnish and I translated users' comments into English for this article.

Training Observation

A basic training course organized for new users was observed to acquire information on the beginning of the learning process. The purpose was to see which functions were difficult to learn, what kind of problems users faced when learning to use the system, what training methods were used, and how training affected the learning results.

Training observation as a method for studying learnability has not been widely discussed in literature. However, it has been mentioned by Karn, Perry, and Krolczyk (1997) as one method for collecting learnability data. Because training sessions are organized regularly for new Tekla Structures users, training observation was an easily arranged and efficient method for evaluating learnability.

The training course that I observed lasted 3 days. All six users who attended the training course had been interviewed prior to the course. The training course consisted of demonstrations given by the instructor and exercises that the subjects performed according to the instructions in the training material. The training material was available in both printed and electronic form. The instructor helped the subjects with the problems they faced while doing the exercises.

I observed the six subjects while they performed the exercises and took notes on an observation template, which was a table with the following columns:

- main topics covered in the training (which corresponded to chapters in the training material)
- time that was spent with each main topic
- subtopics covered in the training (corresponded to subsections in the training material)
- teaching methods
- concepts that were explained
- concepts that were not explained
- references in the training material to additional learning resources (the references were available as links in the electronic version of the training material)
- questions that the subjects asked
- behaviors of the subjects.

Usability Tests

The purpose of the scenario-based usability tests was to assess the outcome of the training and the self-learning phase that followed. The tests were expected to reveal issues that are problematic for new users.

Elliott et al. (2002) and Roberts and Moran (1983), for example, have evaluated learnability with scenario-based tests in which users were observed while completing test tasks. Corresponding methods have been used by numerous other researchers for evaluating usability.

In this study, the usability test consisted of 19 test tasks. The tasks contained the most essential phases of a real modeling project, but on a smaller scale. The subjects received the same initial information as in a normal project, and a task scenario was presented to indicate the goal of the modeling task.

Each subject was observed individually while completing the tasks with the system. The usability test lasted about 1 hour and was organized at each subject's office. The subject was asked to think aloud while completing the tasks (see Salter, 1988, for the think-aloud method). The researcher observed the behavior of the subject and took notes on the performed steps, errors, and subject's comments. The test sessions were also audio recorded.

The test was repeated twice for each subject, immediately after the training session and 2 months later. The tasks in these two usability tests contained the same essential phases but the details of the tasks differed slightly.

Subjective Satisfaction Questionnaire

After the usability test sessions, the subjects were asked to fill in a two-page questionnaire. The purpose was to address subjective opinions on issues that affect learnability. The need for assessing subjective satisfaction can be inferred from the definition of learnability that contains the word *comfortably*. The use of a questionnaire for measuring the subjective satisfaction dimension of learnability has been suggested in the ISO 9241 standard (ISO, 1998b).

The questionnaire was divided into four sections: general questions, learnability of the user interface, materials and training, and function-specific questions. The questionnaire contained 30 questions in total. A 5-point Likert scale (see Lewis, 1995), with a pair of polar adjectives as anchors, was used. The questionnaire answers were scored and average grades were calculated for each question.

RESULTS

Pre-Training Interviews

The pre-training interviews provided evidence that the users had rather detailed assumptions about the Tekla Structures system but their assumptions were often partly incorrect. I noticed that users based their expectations mainly on the software they had used earlier. For example, when users were asked about mirroring objects or modifying part marks, they explained how the operation was performed with the software they were familiar with and that they expected Tekla Structures to work similarly. This is in line with the theory of mental models. The users' mental models were based on familiar software programs; their mental model would change, then, as they learn more about the new system.

It was also observed that users guessed the functionality of and could use the simplest features of the system surprisingly well without any training. For example, users were able to create a model with some columns, beams, and connections. In these cases, the system seemed to direct the user to perform the right sequence of actions. On the other hand, users could not perform more complex tasks without instructions, such as controlling the connection parameters or changing the drawing layout.

A sample of issues that were surfaced in the pre-training interviews as well as the number of users with whom each issue was noticed are listed in Table 1.

Table 1. A Sample of Issues Noticed in the Pre-training Interviews.

Issues noticed	# of subjects
The meaning of the basic command buttons used in the system--OK, Apply, and Modify--were not intuitive to the subjects.	4
The subjects expected that objects would be mirrored similarly in a 3D environment as in a 2D environment, which is not true.	2
The subjects were not familiar with the concept of numbering.	4
The subjects could easily place building elements in the model without any training.	5

I formulated the issues that were surfaced in the pre-training interviews as “observations on learnability.” In total, 41 observations concerning learnability were collected. In some cases, several related issues were combined into one observation sample. Presenting related issues as one chunk made it easier to analyze the data after completing all the research activities.

Training Observation

In similar fashion to the pre-training interviews, it was observed that the subjects could create basic objects rather easily during the training sessions. They had difficulties with advanced modification tasks and tasks with several phases. When doing those tasks, they often needed help from the instructor, even if very detailed instructions for performing the tasks were given in the training materials.

The observed group was very active in the training. They asked questions about user interface elements, the meaning of concepts, task sequences, and problems they faced when doing exercises. They commented about things that they considered difficult. During the training, 289 questions and comments about difficult things were raised by the subjects. The questions and comments proved to be especially useful in analyzing learnability. A sample of the subjects’ questions and comments are listed in Table 2. The six subjects are marked with the letters A to F.

Table 2. A Sample of the Questions and Comments Recorded During the Training Observation.

Questions posed
(D) “What is the difference between ‘From plane’ and ‘On plane’?”
(E) “What did it do? ‘Pick object’?”
(B) “Why can’t I see the hollow core slabs?”
(F) “Why aren’t all the connections created?”
(A) “Does it accept both capital and small letters? In what form should the profile be given?”
(D) “What did ‘n’ in the drawing list mean?”
(C) “What is the difference between the ‘Save and freeze’ and the ‘Save’ command?”
(C) “And all these windows… It depends on so many things.”

By placing related issues together, a total, 118 distinct observations concerning learnability were gathered. Many of these were based on the users' questions and comments. Some observations were based on the analysis of user behavior, task sequences, and related problems.

Usability Tests

All subjects had some difficulties when working on the test tasks, even if they had completed similar tasks in the training. There were certain problems that confronted a remarkable number of subjects, sometimes even many times during one usability test. Some of the problems that were surfaced in the usability tests had also been identified during the training observation, but usability tests also revealed some new problems. The problems that the users faced allowed me to identify several issues that degrade learnability. However, during the tests I also recorded several things that support learnability, as well as design suggestions for the user interface and training proposed by the subjects. A sample of the problems that the subjects faced is listed in Table 3.

Table 3. A Sample of the Problems Subjects Faced During the Usability Tests.

Problems observed or raised by subjects	# of subjects
The subjects did not remember that object combinations could be stored as a custom component.	5
The subjects had difficulties with entering points for a custom component.	4
The subjects often forgot to run numbering before creating drawings.	5
The subjects were confused about Tools and Setup menus, since both have a numbering item.	3
The subjects wondered if they had succeeded in creating drawings as they saw nothing happening on the screen.	4
The subjects did not understand the difference between the drawing mode and the modeling mode.	4

In total, 137 observations on learnability were made during the usability tests. Of this total, 60 observations were unique to the test right after the training and 36 were unique to the test two months later; 41 of the observations were present in both tests.

Subjective Satisfaction Questionnaire

The questionnaire results revealed how the subjects saw the learnability of the system, the quality of the support materials, and which operations they considered difficult. The questionnaire answers were scored from 1 to 5. The average scores ranged from 1.0 to 4.7, which means that there were clear distinctions between different items.

The scores that the subjects gave for different system functions were in line with the observations made during the training and usability tests. The subjects gave low scores for the functions that had been observed to be problematic and high scores for the ones with fewer problems. A sample of the questionnaire results are shown in Table 4.

In total, 18 observations concerning learnability were extracted from the subjective satisfaction questionnaire results. The questionnaire did not produce quite as much information for the analysis of learnability factors as the other research methods presented above, but this method was important for validating whether the observations and the subjective opinions were in line.

Table 4. A Sample of the Questionnaire Results.

Questionnaire results
Questions related to the user interface: The subjects considered "Remembering names and use of commands" to be the most difficult.
Questions related to training material: The subjects considered the instructions on the computer screen and the actual training sessions the most useful. They considered the training material CD and context-sensitive help the least useful. Help pages and training sessions got a medium score for usefulness.
Function-specific questions: The subjects indicated that creating grids and concrete or steel parts were easy. They indicated that updating and modifying drawings, exporting and importing data, specifying model properties, and modifying material catalogs were difficult. The other eight phases of the modeling process got a medium score for usefulness.

DISCUSSION

The empirical research activities produced a good quantity of data and allowed me to understand the issues affecting learnability rather well. My goal was to process the data further and group the observations into a set of factors affecting learnability. I also wanted to turn the observations into guidelines that could be followed to produce systems that are easy to learn. Next, I describe how I used the grounded theory method to further process the data, and introduce the learnability factors and guidelines that I extracted from the data.

Use of the Grounded Theory Method

After completing the empirical research, a large body of data was available. To compare the data found with different research methods, all the observations were collected into a large table and related observations were combined into one. As a sample, if a subject had stated in the pre-training interview that a certain user interface element was misleading, and in the usability tests, the same element was observed to cause problems, these two items were presented as one observation. After combining the items, there were 237 observations altogether in the table. The table also contained references regarding the research methods used and the subjects from whom each observation was made.

Next, the observations were classified in order to find a set of learnability factors that would cover all the observations. The grounded theory method was used for this. The analysis using the grounded theory method starts with an unorganized set of data and with no predefined theoretical framework. Themes and patterns are identified from the data. As the analysis proceeds, more evidential data for the themes and patterns is sought. In this case, I identified several classes that the observations fell into and tried to identify learnability factors that covered all the observations within one class.

Elliott et al. (2002) used the grounded theory method for creating a model of learnability for hypermedia authoring tools. According to them, the grounded theory method is useful in that it can produce a theory that fits the available set of data, which was the goal in this research as well.

As an outcome of the classification process, 18 factors affecting learnability were identified. The factors were divided into three main categories, namely the user interface, conformity to user's expectations, and training.

The classification of factors affecting learnability and the list of observations were then used to create guidelines for improving learnability. When creating the guidelines, the observations related to each learnability factor were treated one at a time. The observations were arranged into subgroups such that one guideline could cover all of the related observations. I made sure that the guidelines were not overlapping but covered all the observations. In total, 64 guidelines were created to encompass the 237 observations.

As a sample, the observations that led to the identification of the factor Continuity of Task Sequences as well as the related guidelines are presented in Tables 5 to 7. Each table contains the observations related to a specific guideline. The research methods and the number of subjects that each observation was based on are also noted. The supporting data for the learnability factor Continuity of Task Sequences originates mostly from the training observation and usability tests because many problems related to the continuity of task sequences only arise when users perform their tasks with the system.

Table 5. Observations Supporting the Continuity of Task Sequences and Guideline 3.1.

Guideline	Observation	Subjects' comments	Empirical research method	# of subjects
3.1 Provide links between the different steps of a task.	Values on several dialog boxes affect numbering, which increases users' memory load. Subjects were not able to define the numbering settings.	(D) "I don't know at all where I can modify anything. If I change it here, does it do everything again?" (E) "How can I see how the parts have been numbered? What are the numbering settings?"	Training observation First usability test	3 4
	The task sequence for filtering objects out of view requires too much memorization. Subjects had difficulties with remembering the sequence.	(D) "What was the name of the command? I had to filter something. That was rather difficult."	Training observation Second usability test	3 1
	The task sequence for defining drawing classifier settings requires too much memorization. Subjects needed help, as they could not memorize all the steps.	(C) "And all these windows… It depends on so many things."	Training observation Second usability test	5 1
	The task sequence for creating fittings requires too much memorization. Subjects needed help, as they could not memorize all the steps.	(E) "Why doesn't it let me fit this?"	Training observation	3
	The task sequence for creating cuts requires too much memorization. Subjects needed help, as they could not memorize all the steps.	(D) "How do I create cuts?"	Training observation	2
	The task sequence for cutting parts requires too much memorization. Subjects needed help as they could not memorize all the steps.	(C) "What should I do? I chose first properties and then scissors. Is it now ok? Have the properties (of the silo) changed?"	Training observation	5
	The task sequence for importing components requires too much memorization. Subjects needed help, as they could not memorize all the steps.	(F) "How do I know where the component will be placed?"	Training observation	2
	The task sequence for creating AutoConnections requires too much memorization. Subjects needed help, as they could not memorize all the steps.	(D) "What does the whole thing do?"	Training observation	4
	The task sequence for binding components to planes requires too much memorization. Subjects needed help, as they could not memorize all the steps.	(F) "How can I bind it?"	Training observation	3

Table 6. Observations Supporting the Continuity of Task Sequences and Guideline 3.2.

<table>
<tr><th>Guideline</th><th>Observation</th><th>Subjects' comments</th><th>Empirical research method</th><th># of subjects</th></tr>
<tr><td rowspan="6">3.2 Integrate the tasks if they need to be completed sequentially.</td><td rowspan="2">Tools and Setup menus both have an item with the same name (Numbering). This confused the subjects.</td><td>(B) “There is numbering in two places? This is confusing.”</td><td>First usability test</td><td>2</td></tr>
<tr><td>(C) “Ok. It was wrong numbering.”</td><td>Second usability test</td><td>2</td></tr>
<tr><td rowspan="3">There is no link from the Custom component creation tool to the catalog of components. Subjects were not able to find the Custom component that they had just created.</td><td rowspan="3">(E) “I don’t remember where I can find it now.”</td><td>Training observation</td><td>1</td></tr>
<tr><td>First usability test</td><td>3</td></tr>
<tr><td>Second usability test</td><td>1</td></tr>
<tr><td>There is no link from the Export file dialog to the directory where the file is exported. Subjects were not able to check the exported file.</td><td>(E) “Where was it then… ... I cannot find it anywhere.”</td><td>First usability test</td><td>2</td></tr>
</table>

Table 7. Observations Supporting the Continuity of Task Sequences and Guideline 3.3.

<table>
<tr><th>Guideline</th><th>Observation</th><th>Subjects' comments</th><th>Empirical research method</th><th># of subjects</th></tr>
<tr><td rowspan="7">3.3 Make the basic steps of a task easily visible and do not complicate them with advanced options.</td><td rowspan="2">There are too many controls on the Create report dialog. Subjects were not able to identify the correct button and they had difficulty creating reports.</td><td rowspan="2">(D) “So what are we looking for now? Material… ... Is it this one (Create from selected)? ... No, do I need to select them?”</td><td>First usability test</td><td>2</td></tr>
<tr><td>Second usability test</td><td>1</td></tr>
<tr><td rowspan="3">There are too many settings on the Create basic view dialog. Subjects had difficulties with creating a basic view.</td><td rowspan="3">(C) “What does it need now? ... If I press this [OK], does it create it or do I need to save first? ... And I cannot make the view that I want in any way? ... If I take this away, or do I need to take it away?”</td><td>Pre-training interview</td><td>1</td></tr>
<tr><td>Training observation</td><td>2</td></tr>
<tr><td>Second usability test</td><td>2</td></tr>
<tr><td rowspan="2">Too many complex settings are shown for drawings. Some of the subjects could not create a basic drawing without instructions in the usability tests, but they only remembered that a lot of complex settings need to be defined.</td><td rowspan="2">(A) “I will not succeed in it. I think that was a very complex thing to create drawings. …The most difficult thing was creating drawings. I had a lot of difficulties with it.”</td><td>First usability test</td><td>2</td></tr>
<tr><td>Second usability test</td><td>2</td></tr>
</table>

Learnability Factors

Applying the grounded theory method led to the identification of 18 learnability factors. Seven of them are related to the user interface, four to conformity to user’s expectations, and seven to training. An overview of the factors is presented in Figure 3.

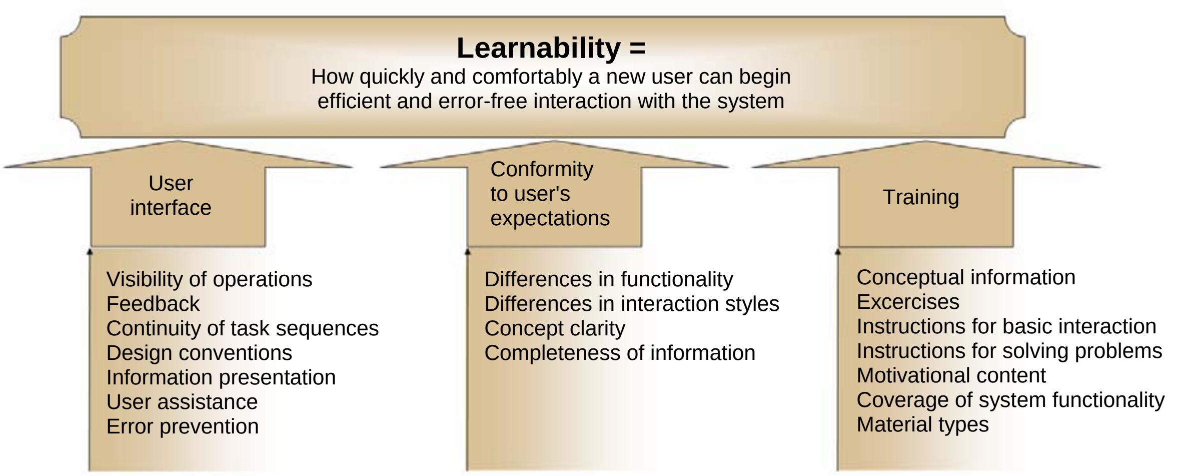

Figure 3. Overview of learnability factors.

Factors Related to the User Interface

The supporting data for the factors related to the user interface arose from situations in which the user interface was misleading or not understandable. Some of these factors are familiar from usability checklists (e.g., Nielsen, 1993). However, the factors in my classification concentrate specifically on the issues that are important for a novice user.

It is indicated in Table 8 how many of the 237 observations support each of the factors. A summary of the supporting observations is presented as well. Each factor is presented in more detail below.

Table 8. Learnability Factors and Observations Related to the User Interface.

Learnability factor	# of observations	Summary of observations
Visibility of operations	68	Subjects had problems finding commands that were not clearly visible near the object with which they were interacting. In addition, the subjects did not necessarily remember command names or locations of commands in the user interface.
Feedback	23	Subjects were often unsure about whether they succeeded with a certain operation in the absence of a confirmative feedback message. They would also have needed feedback about the current system state.
Continuity of task sequences	16	Discontinuities in task sequences were problematic for the subjects. They often did not recognize the way to proceed and, as a result, failed to complete the task.
Design conventions	14	User interface elements that were designed conventionally were easy to understand but unconventional ones caused problems.
Information presentation	45	Graphical presentations or fields without explanations caused problems for the subjects.
User assistance	10	In many problematic situations that were observed in the training and learnability tests, users sought properly designed user assistance to help them overcome the problem.
Error prevention	6	A large portion of the observed errors were made by many, and in some cases all, of the six subjects.

- *Visibility of operations.* An essential requirement for a learnable user interface is the visibility of possible operations. Whereas expert users can rely on experience, novice users must deduce possible operations and inputs from the hints given by the interface.
- *Feedback.* Feedback is useful for experienced users but especially important for novices. They need feedback on the results of operations and the system's state.
- *Continuity of task sequences.* A desirable situation is that when users start a command from a menu or by clicking on an icon they are directed by the system until the desired end result is reached. Users should not be required to jump from one menu item or dialog box to another while performing a single task.
- *Design conventions.* If design conventions are followed, users can easily grasp the meaning and usage of one program's elements from those they have seen in other applications. Design conventions arise from user interface standards and the most common office, Web, or domain-specific software.
- *Information presentation.* Novice users need more detailed descriptions for commands, input fields, and image details than experts do. Special attention should be given to the amount and clarity of information as well.
- *User assistance.* The system should instruct the user and provide additional information on the user interface elements and the related tasks. Current technologies allow user assistance to exist as part of the user interface rather than as a separate help system.
- *Error prevention.* A large portion of the most common errors could be prevented by making small changes in the user interface. In general, the most common causes of errors can be identified by observing new users interacting with the system.

Factors Related to Conformity to User's expectations

The learnability factors in this group reflect the effect of differences between the users' existing mental models and the actual system. These differences may cause learning difficulties. The evidential data for these learnability factors arose from situations in which the subjects expected the system to function differently than it did, and therefore faced problems. In those situations, the subjects had formed their mental model mainly on the basis of a system they had used earlier.

The numbers of observations supporting each learnability factor as well as summaries of the observations are presented in Table 9. Next, each factor is then described in more detail.

Table 9. Learnability Factors and Observations Related to the System Structure.

Learnability factor	# of observations	Summary of observations
Differences in functionality	9	When the subjects described their expectations for new software, it turned out that they based their expectations on their experiences with software they are familiar with. Differences in the functionality between the old and new software caused problems for the subjects.
Differences in interaction styles	16	It could be deduced from the subjects' comments that mental models concerning interaction styles were based on the subjects' experiences with other applications, most commonly office software or operating systems. Subjects expected interaction styles to be domain-independent.
Concept clarity	30	Concepts that had not been used elsewhere caused problems unless they were very self-explanatory, communicated clearly in the user interface, and contained familiar terminology.
Completeness of information	60	Lack of information about the user interface elements, system concepts, and causes and effects of operations caused difficulties with using the system.

- *Differences in functionality.* The functionality of different software applications naturally varies. Usually, it is not desirable to avoid those differences; instead, users should be supported in learning the new functionality.
- *Differences in interaction styles.* Interaction styles of various software applications also vary. Some of this variation may be necessary because of the different nature of the applications; however, some of it is avoidable. Designing the software so that it supports common interaction styles makes the software easier to learn.
- *Concept clarity.* When starting to use a new software application, the user usually needs to learn new concepts. To support learning, new concepts should be communicated clearly with familiar and understandable terminology.
- *Completeness of information.* The change in user's mental model can be facilitated by providing enough information about user interface elements, concepts that are present in the system, and causes and effects of operations.

Factors Related to Training

In this section, training factors that were noticed to affect learnability are presented. The information was extracted from the training observation and comments that the subjects made in the usability tests after the training. When designing training courses to support learning as best as possible, these issues should be considered.

The number of observations supporting each of the factors and a summary of the observations are presented in Table 10. Each factor is then described in more detail next.

Table 10. Learnability Factors and Observations Related to Training.

Learnability factor	# of observations	Summary of observations
Conceptual information	45	Missing conceptual information made the subjects face problems when completing tasks.
Exercises	44	Several subjects commented that they learn best by completing exercises. However, completing a task according to step-by-step instructions provided did not always lead to a persistent learning result.
Instructions for basic interaction	14	Subjects were not familiar with all the basic interaction strategies even after the training, which caused problems.
Instructions for solving problems	16	Subjects were not very well prepared for solving problems themselves but asked for external help when facing problems.
Motivational content	3	It could be deduced from the subjects' comments that they were weighing the advantages of learning the software against the effort spent using it.
Coverage of functionality	9	Some tasks that are central to users' work had received only a little attention in the training and thus the subjects had problems with performing them in usability tests.
Material types	13	Several observations concerning the appropriateness of different material types were made. Users' opinions on the usefulness of different material types varied in different phases of the learning process.

- *Conceptual information.* Conceptual information helps the user to build a revised mental model of the system. For skill learning, mere memorization of procedures is not enough; it is desirable that one truly understands the procedure on a conceptual level as well. Therefore, conceptual information should be included in the training process.
- *Exercises.* For skill learning, it is necessary to practice operations by completing exercises. However, the nature of the exercises also matters. Training should contain exercises that encourage users to process new information and to apply it to new situations.
- *Instructions for basic interaction.* Teaching basic interaction strategies thoroughly in the training will raise productivity during the post-training learning period. This is because users will not need to spend time with simple interaction problems.
- *Instructions for solving problems.* Users will usually face problems when starting with a new software application. To moderate this, users should be equipped with problem solving skills during training. This would help them to use the application competently and independently when no instructor is available to help.
- *Motivational content.* Motivational content is important because it affects the learning behavior of users both during and after the training. Motivational content encourages the users to devote effort to learning more persistently.
- *Coverage of functionality.* Training should concentrate on the system functions that are essential for the users. This can be done only after carefully analyzing user needs.
- *Material types.* The type of the material that is used in training and provided for additional support should be carefully considered. The quality of the material also naturally affects users' perception of its appropriateness.

Learnability Guidelines

Based on the observations and the learnability factors, 64 guidelines for improving learnability were created. They cover issues related to the user interface, conformity to user's expectations, and training. The guidelines are presented next.

Guidelines Related to the User Interface

Altogether, 28 guidelines were formulated for improving the learnability of the user interface. The guidelines are presented in Table 11, and can be used as a checklist when designing new user interface elements. Existing parts of the user interface can also be compared against the

Table 11. Guidelines Related to the User Interface.

Factor	Learnability guidelines	
Visibility of operations	1.1	Place related operations within the same location.
	1.2	Make all controls visible.
	1.3	Distinguish visually the items that cannot be used in a certain situation.
	1.4	Support direct manipulation.
	1.5	Direct the user to give the right input.
	1.6	Avoid modes, or if that is not possible, then indicate the mode clearly.
Feedback	2.1	Provide a system response when the user performs an action.
	2.2	Provide a directive system feedback if the user tries to perform an operation that is not possible in a certain situation.
	2.3	Indicate the existence of hidden information.
Continuity of task sequences	3.1	Provide links between the different steps of a task.
	3.2	Integrate the tasks if they need to be completed sequentially.
	3.3	Make the basic steps of a task easily visible and do not complicate them with advanced options.
Design conventions	4.1	Use controls that are familiar from other applications.
	4.2	Use familiar task sequences for operations that are not domain specific.
	4.3	Provide templates to direct the user to the desired design style.
Information presentation	5.1	Organize menus so that they support user tasks.
	5.2	Design descriptive labels.
	5.3	Avoid system-oriented symbols or abbreviations.
	5.4	Avoid any unnecessary information.
User assistance	6.1	Provide information on existing objects.
	6.2	Inform the user about errors.
	6.3	Give instructions for solving a problem.
	6.4	Design clear instructional texts.
	6.5	Provide advanced and beginner modes.
	6.6	Provide several forms of user assistance.
	6.7	Integrate user assistance into the system interface.
Error prevention	7.1	Automate operations that do not require user action.
	7.2	Change errors to alternative paths of operation.

guidelines and necessary adjustments can be made. Naturally, applying the guidelines requires careful consideration of the user interface elements in question and possibly some expertise in human-computer interaction.

Guidelines Related to Conformity to User's Expectations

Ten guidelines concerning conformity to user's expectations were formulated and they are summarized in Table 12. The guidelines can be referred to when designing new features or introducing new concepts to the system. The guidelines address the issues that may affect the adaptation of users' mental models. As these guidelines are related to the system's structure, underlying concepts, and basic functionality, they must be taken into account early in the system development process.

The problem with creating guidelines for the learnability factor Differences in Functionality was that those differences can seldom be avoided. The very reason to have a new software application is that it meets distinct needs not met by other software applications. Therefore, it is desirable to make the new software application different from others. Clarity in instruction can help bridge the differences between the former mental model and the new mental model.

Table 12. Guidelines Related to Conformity to User's Expectations.

Factor	Learnability guidelines	
Differences in functionality	1	Do not avoid introducing new kinds of functionality but assist the user in learning them.
Differences in interaction styles	2.1	Follow design conventions for controls and task sequences.
	2.2	Allow the user to interact with objects as in other similar software applications.
	2.3	Use menu titles that are familiar from other software applications.
Concept clarity	3.1	Use terminology that is familiar from the real world or other software applications.
	3.2	Avoid terminology that may be cause incorrect associations.
	3.3	Avoid system-oriented terminology.
	3.4	Clarify concepts with symbols and images.
Completeness of information	4.1	Provide explanations for new concepts in the interface.
	4.2	Help the user to perform actions.

Guidelines Related to Training

Table 13 summarizes the 26 learnability guidelines related to training that were formulated on the basis of the observations. They are expected to cover the training issues that have the most significant effect on learning results. The contents and organization of existing training setups can be compared against the guidelines to find the necessary adjustments.

Training sessions differ from each other in terms of the type and number of participants, the duration of the training, the complexity of the subject, practical and physical arrangements, as well as many other dimensions. Therefore, some of the guidelines presented here are intentionally left on a rather abstract level. They present issues that should be checked to assure effective training but the training organizer must also adapt them, as needed, to find the best solution for each training context.

Table 13. Guidelines Related to Training.

Factor		Learnability guidelines
Conceptual information	1.1	Clarify the meaning of unfamiliar terms.
	1.2	Explain the relationship between concepts.
	1.3	Clarify the underlying principles that determine how the system is used.
Exercises	2.1	Introduce the basic form of an operation and require the learner to apply it to new situations.
	2.2	Encourage the learner to actively process the information.
	2.3	State the goal of each exercise clearly.
	2.4	State the conditions in which the operation can be performed.
Instructions for basic interaction	3.1	Demonstrate how to interact with objects.
	3.2	Demonstrate how to adjust the basic settings.
	3.3	Demonstrate how to use the basic controls.
Instructions for solving problems	4.1	Instruct about the available documentation.
	4.2	Demonstrate how to use the documentation.
	4.3	Instruct how to contact support personnel.
	4.4	Address the most common causes of error.
Motivational content	5.1	Summarize the contents of the training at the beginning of the session.
	5.2	Concentrate on practical issues that each learner will need in his/her work.
	5.3	Follow up with learners, if possible.
Coverage of functionality	6.1	Get to know the learners and their needs.
	6.2	Adjust the material to cover all the core tasks.
	6.3	Adjust the time that is spent on each core task according to the difficulty and importance of the task.
Material types	7.1	Provide help that is integrated into the user interface and can be easily accessed from within the system.
	7.2	Provide printed material or dual monitors in training.
	7.3	Provide a limited amount of material to be covered in detail, and supplemental material to be referred to later.
	7.4	Design a clear layout for material.
	7.5	Provide material in the native language of the learner, if possible.
	7.6	Provide search possibilities for digital material.

Comparing the Learnability Factors and Guidelines to Previous Research

Several classifications exist on the factors that affect the usability of a system. In many of those studies, learnability is seen as a subfactor of usability. However, the classifications of factors affecting learnability are less common.

My learnability guidelines and the usability guidelines that have been presented in the literature have some issues in common. For example, I have Error Prevention in the list of user interface related learnability factors, and Nielsen (1993) includes it in his list of usability heuristics. One of my user interface-related learnability factors is Visibility of Operations, whereas Nielsen stresses the visibility of system status in his heuristics.

However, the classifications of usability attributes seldom address the issues that I have in the categories of Conformity to User's Expectations and Training. In the beginning of this article, I discussed how usability has been divided into subattributes by Nielsen (1993), Dix et al. (1998), and Lin et al. (1997). All of these researchers concentrate on attributes of the user interface and not on user's expectations or training. There may be situations in which training is not available and it is not possible to change the underlying system concepts to correspond to user's expectations. Then, it may be sufficient to evaluate only the effect of user interface on learning. However, in most cases, it is beneficial to take a multifaceted view of the learning process and address also user's expectations and training, as has been done in this study.

Nevertheless, the classifications of usability attributes presented in the literature and my classification of factors affecting learnability do not contradict each other, but rather, in fact, are complementary. My detailed classification can be used to analyze the learnability of complex systems corresponding to the building modeling system, and to identify ways to improve learnability. General usability classifications, such as the one presented by Nielsen (1993), can be applied to a wider range of systems from consumer products to software applications, as it has been left on a more general level than the classification presented in this article.

CONCLUSIONS

In this paper, 18 factors affecting the learnability of a building modeling system have been presented. These factors can be used as a general framework for understanding the learnability of this system. In addition, 64 guidelines for improving learnability have been introduced. By following these guidelines in system development and training, the learnability of the building modeling system can be improved.

Throughout the study, three aspects influencing learnability were addressed: the user interface, conformity to user's expectations, and training. Learnability studies have often concentrated on the effect of the user interface, but I believe that a classification addressing the other two distinct aspects of learnability as well helps to improve the learning process and system learnability as a whole.

The classification of learnability factors and guidelines was based on a body of empirical data collected via several research methods. The classification was created with the grounded theory method that is intended for creating a theory that fits the available set of data.

The classification should have practical relevance to other developers of complex systems as well. The learnability factors and guidelines can be used as a reference when analyzing and improving the learnability of any systems. However, it must be noted that the factors and guidelines are based on the empirical data concerning a building modeling system. Thus, some of the factors and guidelines may not even apply to a system whose scope differs radically from the scope of the building modeling system I studied. Furthermore, the emphasis put on the different factors and guidelines may vary for different systems. However, the grounded theory methodology that was used for analyzing the learnability of a building modeling system can be applied to other systems as well. This would produce corresponding classifications of learnability factors and guidelines that take into account the particularities of each system.

I expect that the results concerning learnability are of interest not only for system developers but also for the body of HCI researchers. Not many classifications of factors affecting the learnability of complex systems have been introduced in the HCI literature. This is true for learnability guidelines as well: Several sets of usability guidelines have been presented in the literature, but sets of learnability guidelines are less common.

In the future, it would be especially interesting to study in more detail the effect of differences between users' mental model and the actual system. Another future research topic would be to validate the learnability factors and guidelines. This could be done by implementing changes to real systems according to the guidelines and measuring the effect of the changes on the performance of new as well as expert users.

REFERENCES

Bevan, N., & Macleod, M. (1994). Usability measurement in context. *Behavior and Information Technology, 13*, 132–145.

Butler, K. A. (1985). Connecting theory and practice: A case study of achieving usability goals. In L. Borman & R. Smith (Eds.), *Proceedings of the SIGCHI* [*Special Interest Group on Computer-Human Interaction*] *conference on human factors in computing systems* (pp. 85–88). New York: ACM Press.

Dix, A., Finlay, J., Abowd, G., & Beale, R. (1998). *Human-computer interaction* (2nd ed.). Hertfordshide, UK: Prentice Hall International.

Dykstra-Erickson, E., & Curbow, D. (1997). The role of user studies in the design of OpenDoc. In S. Coles (Ed.), *Proceedings of the conference on designing interactive systems: Processes, practices, methods, and techniques* (pp. 111–120). New York: ACM Press.

Elliott, G. J., Jones, E., & and Barker, P. (2002). A grounded theory approach to modeling learnability of hypermedia authoring tools. *Interacting with Computers, 14*, 547–574.

Fein, R. M., Olson, G. M., & Olson, J. S. (1993). A mental model can help with learning to operate a complex device. In S. Ashlund, K. Mullet, A. Henderson, E. Hollnagel, & T. White (Eds.), *INTERACT '93 and CHI* [*Computer-Human Interaction*] *'93 conference companion on human factors in computer systems* (pp. 157–158). New York: ACM Press.

Goodwin, N. C. (1987). Functionality and usability. *Communications of the ACM, 30*, 229–233.

Green, A., & Eklundh, K. S. (2003). Designing for learnability in human-robot communication. *IEEE Transactions on Industrial Electronics, 50*, 644–650.

International Organization for Standardization [ISO]. (1998a). Standard ISO 9241: Ergonomic requirements for office work with visual display terminals (VDTs), part 10: Dialogue principles. Geneva: ISO.

International Organization for Standardization [ISO]. (1998b). Standard ISO 9241: Ergonomic requirements for office work with visual display terminals (VDTs), part 11: Guidance on usability. Geneva: ISO.

Karn, K. S., Perry, T. J., & Krolczyk, M. J. (1997). Testing for power usability. In S. Pemberton (Ed.), *CHI* [*Computer-Human Interaction*]*'97 extended abstracts on human factors in computing systems: Looking to the future* (p. 235). New York: ACM Press.

Kellogg, W. A., & Breen, T. J. (1987). Evaluating user and system models: Applying scaling techniques to problems in human-computer interaction. In J. Carroll & P. Tanner (Eds.), *Proceedings of the SIGCHI/GI* [*Special Interest Group on Computer-Human Interaction/Graphics Interface*] *conference on human factors in computing systems and graphics interface* (pp. 303–308). New York: ACM Press.

Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction, 7*, 57–78.

Lin, H. X., Choong, Y.-Y., & Salvendy, G. (1997). A proposed index of usability: A method for comparing the relative usability of different software systems. *Behaviour & Information Technology, 16*, 267–278.

Marton, F., & Booth, S. (1997). *Learning and awareness*. Mahwah, NJ: Lawrence Erlbaum Associates.

Nielsen, J. (1993). *Usability engineering.* San Francisco: Morgan Kaufmann.

Paymans, T. F., Lindenberg, J., & Neerincx, M. (2004). Usability trade-offs for adaptive user interfaces: Ease of use and learnability. In J. Vanderdonckt, N. Nunes, & C. Rich (Eds.), *Proceedings of the 9th international conference on intelligent user interface* (pp. 301–303). New York: ACM Press.

Rieman, J., Lewis, C., Young, R. M., & Polson, P. G. (1994). Why is a raven like a writing desk? Lessons in interface consistency and analogical reasoning from two cognitive architectures. In B. Adelson, S. Dumais, & J. Olson (Eds.), *Proceedings of the SIGCHI* [*Special Interest Group on Computer-Human*

Interaction] *conference on human factors in computing systems: Celebrating interdependence* (pp. 438–444). New York: ACM Press.

Roberts, T. L., & Moran, T. P. (1983). The evaluation of text editors: Methodology and empirical results. *Communications of the ACM*, *26*, 265–283.

Salter, W. J. (1988). Human factors in knowledge acquisition. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 957–968). Amsterdam: Elsevier Science Publishers B.V.

Santos, P. J., & Badre, A. N. (1995). *Discount learnability evaluation.* Graphics, Visualization & Usability Center, Georgia Institute of Technology. Retrieved December 2, 2004, from ftp://ftp.gvu.gatech.edu/pub/gvu/tr/1995/95-30.pdf

Shayo, C., & Olfman, L. (1998). The role of conceptual models in formal software training. In R. Agarwal (Ed.), *Proceedings of the 1998 conference on computer personnel research* (pp. 242–253). New York: ACM Press.

Sinkkonen, I. (2000). *Things that facilitate learning in products.* Unpublished licentiate thesis, Helsinki University of Technology, Espoo, Finland.

Whiteside, J., Jones, S., Levy, P., & Wixon, D. (1985). User performance with command, menu, and iconic interfaces. In L. Borman & R. Smith (Eds.), *Proceedings of the SIGCHI* [*Special Interest Group on Computer-Human Interaction*] *conference on human factors in computing systems* (pp. 185–191). New York: ACM Press.

Author's Note

I thank all the employees at Tekla Corporation who gave their input into the research project. I also thank the participants of the empirical research. In addition, I present thanks to the researchers at the Helsinki University of Technology for the discussions that contributed to the writing of this paper.

All correspondence should be addressed to:
Minttu Linja-aho
Tekla Corporation
P.O.Box 1
FI-02131 Espoo, FINLAND
minttu.linja-aho@tekla.com

Human Technology: An Interdisciplinary Journal on Humans in ICT Environments
ISSN 1795-6889
www.humantechnology.jyu.fi