HELSINKI UNIVERSITY OF TECHNOLOGY

Minttu Linja-aho

# Evaluating and Improving the Learnability of a Building Modeling System

Master's Thesis
Espoo, January 5, 2005

Supervisor: Marko Nieminen, Dr.Sc.(Tech.)
Instructor: Pia Nakari, M.Sc.(Tech.)

HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Automation and Systems Technology

ABSTRACT OF MASTER'S THESIS

| | |
|---|---|
| Author<br>Minttu Linja-aho | Date<br>January 5, 2005<br>Pages<br>121 |
| Title of thesis<br>Evaluating and Improving the Learnability of a Building Modeling System | |
| Professorship<br>User interfaces and usability | Professorship Code<br>T-121 |
| Supervisor<br>Marko Nieminen, prof., Dr.Sc.(Tech.) | |
| Instructor<br>Pia Nakari, M.Sc.(Tech.) | |

The objective of this study is to classify factors affecting learnability and to suggest means of improving the learnability of a building modeling system. The term 'learnability' signifies how quickly and pleasantly a new user can begin efficient and error-free interaction with a system. The learning period for complex systems designed for domain experts is often very long, and therefore learnability improvements would be especially desirable in such systems.

The user interface is an important but not the sole determinant of learnability. This thesis accordingly addresses the effect of the system structure, because the match between the mental models of users and the actual system structure is presumably crucial for learnability. In addition, as most new users of the building modeling system attend a training session, this thesis also addresses the effect of training on learnability.

Several empirical methods were used for examining learnability. A longitudinal study was arranged to obtain information on different phases of the learning process. First, users interacting with the system for the first time were interviewed to obtain information on their mental models. Next, a three-day training session for new users was observed and training material was analyzed. Users' skill levels were assessed immediately after the training and two months later by observing them in a scenario-based learnability test. Users were also asked to fill in a satisfaction questionnaire form. All the research concentrated on the users' core tasks, which had been defined beforehand in user interviews.

Learnability-related phenomena were extracted from the empirical data and they were analyzed to determine factors affecting learnability. Three groups of factors were formed: factors related to the user interface; differences between the mental models of users and the system structure; and training. General learnability guidelines and detailed suggestions for changes in the user interface, system structure and training were proposed on the basis of the learnability factors. These learnability guidelines and the classification of factors affecting learnability are the main contribution of this study. The detailed suggestions can be applied to produce immediate improvements in the learnability of the building modeling system.

Keywords: Learnability, ease of learning, usability, building modeling, complex systems, learning

| | |
|---|---|
| TEKNILLINEN KORKEAKOULU<br>Automaatio- ja systeemitekniikan osasto | DIPLOMITYÖN TIIVISTELMÄ |
| Tekijä<br>Minttu Linja-aho | Päiväys<br>5.1.2005<br>Sivumäärä<br>121 |
| Työn nimi<br>Rakennesuunnittelujärjestelmän opittavuuden analysointi ja kehittäminen | |
| Professuuri<br>Käyttöliittymät ja käytettävyys | Koodi<br>T-121 |
| Työn valvoja<br>Prof., TkT Marko Nieminen | |
| Työn ohjaaja<br>DI Pia Nakari | |



Tämän työn tavoitteena on luokitella rakennesuunnittelujärjestelmän opittavuuteen vaikuttavia tekijöitä ja esittää keinoja parantaa järjestelmän opittavuutta. Opittavuudella viitataan siihen, kuinka nopeaa ja miellyttävää järjestelmän tehokkaan ja virheettömän käytön aloittaminen on. Monimutkaisten asiantuntijajärjestelmien oppimiseen kuluva aika on usein huomattavan pitkä ja siksi opittavuusparannukset ovat erityisen toivottavia.

Käyttöliittymä on olennainen mutta ei ainoa opittavuuteen vaikuttava tekijä. Tässä tutkimuksessa käsitellään myös järjestelmän rakenteen vaikutusta opittavuuteen, koska on oletettavaa, että käyttäjän käsitemallin ja järjestelmän todellisen rakenteen vastaavuus on opittavuuden kannalta tärkeää. Koska suurin osa rakennesuunnittelujärjestelmän uusista käyttäjistä osallistuu koulutusjaksoon, käsitellään lisäksi koulutuksen vaikutusta opittavuuteen.

Opittavuuden tutkimiseen käytettiin useita empiirisiä menetelmiä. Käyttäjiä seurattiin pitkittäistutkimuksella, koska haluttiin tietoa oppimisprosessin eri vaiheista. Ensin haastateltiin käyttäjiä, jotka käyttävät järjestelmää ensimmäistä kertaa, jotta saatiin tietoa heidän järjestelmää koskevista käsitemalleistaan. Seuraavaksi havainnoitiin kolmipäiväistä koulutusjaksoa ja analysoitiin koulutusmateriaalia. Heti koulutuksen jälkeen ja kaksi kuukautta myöhemmin käyttäjien taitotasoa arvioitiin skenaariopohjaisessa opittavuustestissä. Käyttäjiä myös pyydettiin täyttämään miellyttävyyttä koskeva kyselylomake. Kaikissa opittavuustutkimuksen vaiheissa keskityttiin käyttäjien perustehtäviin, jotka oli määritelty ennen edellä mainittuja empiirisiä tutkimuksia haastattelemalla järjestelmän nykyisiä käyttäjiä.

Empiirisillä menetelmillä kerätystä tiedosta löydettyjä opittavuusilmiöitä analysoitiin ja ryhmiteltiin. Tässä työssä esitetään niiden perusteella muodostetut käyttöliittymää, käyttäjien käsitemallien ja järjestelmän todellisen rakenteen vastaavuutta sekä koulutusta koskevat opittavuustekijät. Työssä esitellään myös opittavuustekijöihin perustuva opittavuutta tukevan suunnittelun ohjeistus. Lisäksi työ sisältää yksityiskohtaisia ehdotuksia käyttöliittymässä, järjestelmän rakenteessa ja koulutuksessa tehtävistä muutoksista.

Tämän tutkimuksen tärkein tuotos on oletettavasti opittavuustekijöiden ryhmittely ja opittavuutta tukevan suunnittelun ohjeistus. Yksityiskohtaisten ehdotusten avulla järjestelmän opittavuudessa voidaan tehdä nopeita parannuksia.



Avainsanat: Opittavuus, käytettävyys, rakennesuunnittelu, monimutkaiset järjestelmät, oppiminen

# Acknowledgements

This thesis was written at the Tekla Corporation towards the completion of my studies at the Helsinki University of Technology. I wish to thank all the people that made it possible to engage in this interesting research at Tekla.

I want to thank especially my instructor at Tekla, Pia Nakari, for the advice and support during the research, and my supervisor, professor Marko Nieminen, for important suggestions and feedback. I also wish to thank members of the Usability engineering team, Requirements management team, and Service team at Tekla for discussions that helped me to get new ideas and reflect on and criticize the existing ones. In addition, I am grateful to the participants of the empirical research activities, as well as Sari Kujala and Osmo Tolvanen that read a draft of my thesis and gave constructive feedback.

I also wish to thank my parents that have encouraged me in my interest towards learning new things. I wish to thank my friends as well. And I want to thank my dear husband Vesa for all his love and support.

I want to give the greatest thanks to my Heavenly Father and his son Jesus. They have everything in their hands, and all peace of mind and joy comes from them.

Espoo, January 5, 2005

Minttu Linja-aho

# Table of contents

# 1 Introduction

## 1.1 Why Learnability?

Building modeling systems are complex software applications intended for creating a three-dimensional model containing all the components of a building. A few decades ago, structural engineers drew those components two-dimensionally with paper and pen. Later, computerized drawing systems were introduced but at first they were based on two-dimensional drawing. Nowadays, designers can create a parametric three-dimensional model instead of a symbolic two-dimensional drawing. The information can be transferred in an electronic form to other parties of the building modeling process. All this requires building modeling systems to contain a lot of functionality. The downside of the development is that learning to use the current building modeling systems takes a lot of time.

The term learnability can be used to describe the ease or difficulty of learning to use a building modeling system. Learnability refers to the experience of a new user when he is starting to use the system. It should be possible to learn efficient and error-free interaction quickly and the learning process should be pleasant.

Several researchers in the usability research domain have recognized the importance of learnability in determining usability and system acceptability (e.g. Butler, 1985; Santos & Badre, 1995). Learnability has been found to be correlated with user satisfaction (Lin, Choong, & Salvendy, 1997), and user satisfaction in turn may be a critical factor in determining whether individuals or organizations will adopt a new system as a part of their processes or whether they will stick with their old systems. Therefore, it is desirable to design interfaces that are easy to learn.

Learnability is an especially critical attribute for complex systems, such as current building modeling systems. As stated above, the learning period of complex systems may be very long - from hours to even months. Learnability improvements would enable users to learn to use the system faster and more pleasantly. This would raise user satisfaction, which was already mentioned above, and in addition, raise productivity in the beginning of the learning process and shorten the time to achieving the maximum productiveness.

However, in addition to meeting the learnability criterion, complex systems that are used by domain experts must also be maximally efficient to use (Santos & Badre, 1995). It has been discussed whether these two attributes are contradicting or support each other. Several studies have indicated that learnability and efficiency - and also learnability and usability - 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) remind that experts and novices may have different requirements for a system; abbreviations and shortcuts, for example, will improve the performance of experts but may slow down the learning of a novice. Thus, balancing learnability and efficiency in a user interface is a challenging task.

Even if sometimes compromises are needed to balance different usability attributes such as learnability and efficiency, novice users are an important user group and therefore the learning dimension should be taken into consideration when designing the system. Lin et al. (1997) have written the following statement that states the requirement for learnability and elaborates on the learning process:

> "Well-designed computer software should be easy to learn. Humans can learn through several formats such as rote learning, learning through understanding, or learning by exploration. The learning process will be enhanced and the result will be retained if users are presented with a well-designed, well organized interface." (Lin et al., 1997)

In the quote, Lin et al. (1997) refer to both the characteristics of the learning process and the presentation of the user interface. In general, to understand learnability, it is necessary to consider the whole learning process and the changes that take place in users' mental models, in addition to considering the details of the user interface.

## *1.2 Research Framework*

The need for researching learnability arose from the request to find ways to shorten the learning period of new users of the Tekla Structures system. Tekla Structures is a building modeling system used by structural engineers for designing and fabricating buildings with steel or concrete structures. With Tekla Structures, three-dimensional models containing information on materials, strengths, and other structural parameters can be created. Drawings and reports can be created automatically of the three dimensional model. Drawings and reports are then sent forward to other parties of the construction process.

Tekla Structures is a complex system for domain experts in structural engineering and it includes an expansive number of functions. Its complexity is demonstrated by the fact that there are 220 commands on the first menu level, and in addition, there are dozens of commands on successive menu levels. The complexity of the system leads to long learning times. Most users attend a three-day training session but only a small subset of system functions are learned in that time.

The usability of the Tekla Structures system has been studied earlier and there are usability engineers working at Tekla. However, the learning process of new users has not been studied in detail before. My research addresses the learning issue.

During the research process, I have received information on the previous usability considerations and the building modeling process from usability engineers and requirement management specialists at Tekla. Based on that information, I planned and carried out a series of empirical research sessions and analyzed the results to obtain information on learnability.

In this study, learnability is approached from the perspective of usability research. Learnability and usability can be seen as analogous: A central goal in usability research is to make products easier to use. Correspondingly, in this learnability research, a central goal is to make a product easier to learn. In addition, many usability researchers have recognized learnability to be one of the factors pertaining to usability. In this study, the conceptual framework is largely adopted from the vocabulary used by usability researchers. Many of the observational research methods that are used in this study are known from the usability engineering practice. However, in this study, the usability research perspective is extended by including some theories and methodologies from cognitive and pedagogical sciences.

## *1.3 Objectives*

The objective of this study is to recognize factors that affect the learnability of a building modeling system, Tekla Structures, and to find ways to improve its learnability. In this study, we adopt a broad view of learnability by considering the effect of the user interface, differences between users' mental models and the system structure, and training on learnability. Learnability factors related to these aspects will be researched. Information on the factors affecting learnability will be used to create suggestions for improving the learnability of the user interface. Both general guidelines for improving the learnability of a complex system and detailed suggestions for improving the learnability of the user interface, system structure, and training are presented.

The learnability research concentrates on the core tasks of users. The core tasks are defined before the empirical learnability evaluations by interviewing users of the system. However, some of the results can be applied to other parts of the system as well.

Two main research questions have been formulated for this study. Three focusing questions for the first main research questions have also been formulated. The questions are the following:

1. Which factors affect the learnability of a building modeling system?
    a. Which user interface design issues affect learnability?
    b. How do the differences between mental models of users and the actual system structure affect learnability?
    c. Which characteristics of training sessions support learnability?
2. How can the learnability of a building modeling system be improved?

The following diagram clarifies the empirical research activities that are performed in this study to find answers to the research questions. The arrows contain empirical research activities and the balloons indicate their outcomes. The first two arrows and a balloon are related to the research on core tasks. The following seven arrows indicate the research methods that are used for evaluating learnability. Learnability phenomena will be extracted from the observations, the phenomena will grouped into learnability factors, and suggestions for improving learnability will be based on the grouping of learnability factors.

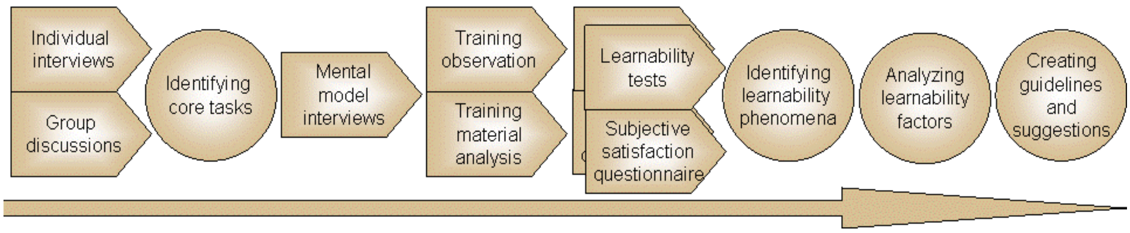



**Figure 1.** Empirical research activities.

In this study, we expect to find some learnability factors that have already been introduced in the literature concerning learnability and some factors that are familiar from usability guidelines, system design principles, or pedagogical theories. However, we expect that our classification of factors will reflect the characteristics of the building modeling system and are therefore unique. We also expect that the learnability factors are a good basis for creating suggestions for improving the learnability of the system.

## *1.4 Structure of the Thesis*

In this chapter (chapter 1), some introductory information on learnability and Tekla Structures is provided. In addition, the research questions are presented. In the next chapter (chapter 2), the research framework and the theoretical framework for learnability will be presented. The chapter starts with a definition of learnability and continues with a review of theories and research results on three issues closely related to learnability. The first issue is learnable user interfaces, the second issue is mental models, and the third issue is the characteristics of the human learning process. In chapter 3, the building modeling process and the Tekla Structures system are presented. Chapter 4 includes descriptions for empirical methods used in this study as well as the results of each method. In chapter 5, a classification of factors affecting learnability is presented. The classification is based on the learnability phenomena that were noted during the empirical research described in chapter 4. Chapter 6 contains suggestions for improving learnability by redesigning parts of the user interface, changing the system structure or restructuring training. In chapter 7, conclusions are made of the empirical methods, factors affecting learnability, learnability improvements, reliability, validity, and generalizability of results, and suggestions for further research.

# 2 Theoretical Framework for Learnability

## 2.1 *What Is Learnability*?

### Definition of Learnability

Several definitions for learnability have been presented in the literature and examples of them are shown in table 1. Some of the definitions consider learnability to be a measurable attribute; others explain it with general terms such as the "experience of a new user". As was said in chpater 1.2, we approach learnability from the perspective of usability research, and therefore most of the definitions have been presented by usability researchers.

In the literature, the terms ease-of-learning and learnability have often been used interchangeably. Both terms can be found in the definitions below. However, in our study, we use only the term learnability.

**Our definition.** We define learnability as follows: *Learnability signifies how quickly and pleasantly a new user can begin efficient and error-free interaction with the system.*

**Other definitions.** Our definition of learnability has been constructed by looking at definitions given by different authors, considering the additional issues we want to include in the definition, and combining these into a definition that we find feasible for the system in question. In the following table, we present definitions of different authors.

**Table 1.** Definitions of learnability.

| Author | Definition presented by the author | Our stand on the definition |
|---|---|---|
| **ISO 9241-11 - standard, SFS-EN 1998** | Part 11 of the ISO 9241 standard defines learnability *through the three attributes of efficiency, effectiveness and satisfaction.* | Our definition is formulated differently that the ISO definition but contains the same elements. The words 'speed' and 'pleasantness' included in our definition are closely related to 'effectiveness' and 'satisfaction' included in the ISO definition. |
| **N. Bevan and M. Macleod, 1994** | N. Bevan and M. Macleod defined learnability to comprise of the usability attributes of *satisfaction, effectiveness, and efficiency that are evaluated in a certain context*, namely the context of the new user. | Bevan and Macleod's definition is close to the ISO definition presented above. As stated above, our definition is formulated differently but contains the same elements. |
| **A. Dix, J. Finlay, G. Abowd, and R. Beale, 1993, p. 131-137** | Alan Dix et al. define learnability in their book *Human-Computer Interaction* to be the *ease with which new users can begin effective interaction and achieve maximal performance.* | This definition is very close to our definition. This definition refers to 'effectiveness' and 'performance' whereas our definition refers to 'efficiency' and 'error-freeness'. We have focused to word 'ease' to 'speed' and 'pleasantness'. |
| **J. Nielsen, 1993, p. 28-30** | Jacob Nielsen writes that ease of learning refers to the *novice user's experience on the initial part of the learning curve.* | For our study, we need a stricter definition than the 'experience' of a new user, and therefore we did not adopt this definition for our study. |
| **T. Paymans, J. Lindenberg, and M. Neerincx, 2004** | In a study presented by Paymans et al. concerning context-aware user interfaces, learnability referred to the *change in users' mental models before and after using the system*. | We have also addressed the change of mental models in our study, but we do not define learnability only in terms of mental models. |
| **G.J. Elliott, E. Jones, and P. Barker, 2002** | G.J. Elliott et al. criticize the practice to define learnability simply in terms of the time required to learn to interact with a system. They propose a *causal model for learnability that comprises of factors affecting learnability and their causal relationships*. Their model will be presented in section 2.2. | In our study, we present a corresponding model for the learnability of building modeling systems. |

**Our operationalization.** Our study is mainly qualitative and therefore we can collect information by observing users and noting issues that affect learnability. However, we also wanted to formulate an operational definition of learnability, in order to be clear on what is meant by the learnability of a system.

Our definition of learnability defines the goal of the learning process to be "effective and problem-free interaction". Effectiveness is defined in the ISO 9241-11 standard as "accuracy and completeness with which users achieve specified goals". Error-freeness is self-explanatory, except that it must be decided what kinds of incidents are counted as errors. These goals of effectiveness and error-freeness led us to evaluating the learning results by counting the number of tasks completed successfully and without errors. We considered an error to be an incident in which the user faced a problem that he could not solve without help, spent extensive mental effort on finding a solution to a problem, needed to undo and redo commands, or resorted to trial-and-error strategy. In addition to counting tasks completed successfully and without errors, we also measured task time, for reference purposes.

The problem with evaluating learning results by counting tasks completed successfully and without errors and recording task time is that when a complex system is studied, it is impossible to perform the measurements before users have received some training. Thus, only the skill level after the training can be addressed with these measures. However, as our study is mainly qualitative, it is enough to perform the measurements only after the training and use other methods to collect qualitative information on learnability before the training.

Our definition also states that the learning process should be fast and pleasant, the first being an objective attribute and the second one subjective. The first attribute, "fast", refers to the skill level – the level of effectiveness and error-freeness – that users have reached in a certain time period. We address this attribute by evaluating users' skill level immediately after the training and two months later. The second attribute, "pleasant", refers to the subjective opinion of users on whether learning the system is annoying or convenient. We use a questionnaire to address this attribute and collect comments presented by users that reflect their subjective opinions.

**Other operationalizations.** In the ISO 9241-11 standard, learnability is operationalized through the three attributes of effectiveness, efficiency, and satisfaction. For each attribute, one or more measures are presented. Effectiveness measures for learnability are the number of functions learned and the percentage of users who manage to learn to a certain criterion. Efficiency should be measured in relation to a specified level of effectiveness. Efficiency measures are time to learn to criterion, time to re-learn to criterion, and relative efficiency while learning. Satisfaction can be measured with a rating scale for ease of learning.

Some other authors have operationalized learnability in terms of errors, task time, error recovery time, and time to mastery. Butler (1985) assigned the learnability an operational definition that contains the time to mastery and error avoidance or recovery. Carrolls and Carrithers (1984) evaluated learnability by counting errors and measuring the time that is needed to perform a task as well as the time that was needed for error recovery. The operationalizations presented by Butler (1985), Carrols and Carrithers (1984), and in the ISO standard are rather close to our definition.

Nielsen (1993, p. 28-30) wrote that learnability can be operationalized as the time that a new user needs to reach a predefined level of proficiency. However, in this study, we cannot measure learnability by defining a level of proficiency and measuring when users have reached it because reaching a feasible level of proficiency would take from days to weeks or even months. Instead, we need to arrange several meetings with users and measure what they have learned until the meeting.

As the learnability measures presented above are rather similar to our operationalization, we can assume that our operationalization is feasible. We must remember, however, that as our study is mainly qualitative, we should not stick too strictly with assessing these measures but our main task is to collect qualitative information on issues affecting learnability. Actually, some qualitative studies on learnability, such as one reported by Elliott et al. (2002), do not present any operationalization of learnability but concentrate on the qualitative observations.

## Learnability as a Usability Attribute

**Relationship of learnability and usability.** To place the concept of learnability into its framework, some information on usability and its upper level concepts is presented here. ISO standards present an often-cited model of usability. According to part 11 of the ISO 9241 standard, usability refers to how efficiently, effectively and pleasantly the user can use the product in a certain environment to reach a certain goal. It is stressed in the standard that the context must always be taken into account when evaluating usability. (SFS-EN, 1998) The following figure presents usability as it is seen in the ISO 9241-11 standard.

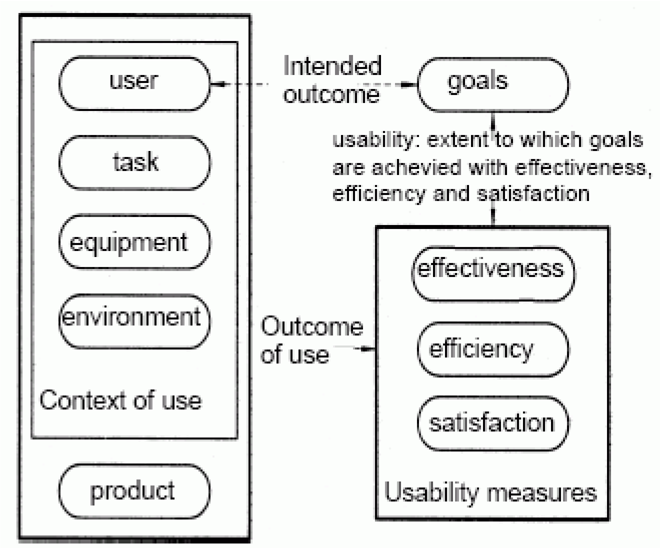



**Figure 2.** Usability model presented in the ISO 9241-11 standard (SFS-EN, 1998).

There are contradicting views of how learnability relates to usability. Some researchers consider ease-of-learning to be a sub concept of ease-of-use. Others see those two as competing attributes that can seldom be fulfilled at the same time. A third group sees ease-of-learning as an attribute that covers the whole usage process; product can be easy-to-learn for beginners but also for experienced users if it constantly assists the experienced user to find new, more efficient ways to work. (Sinkkonen, 2000, pp. 15-16)

In this study, we look at learnability from the perspective that was presented first; we consider learnability to be a sub concept of usability. We consider learnability to relate mainly to the initial learning process whereas usability covers the whole life span of the product. We also recognize that different usability attributes may place contradicting requirements on usability; for example, improving learnability may call for simplicity and improving efficiency may call for more shortcuts, abbreviations, and more functionality. We could also have included into the definition of learnability the aspects of re-learning functions after a period of not using them, or continuous learning that refers to experts learning to use new functions. However, to delimit the subject, we decided to concentrate on the initial learning process.

Elliott et al. (2002) have discussed the relationship of learnability and usability in their publication. They refer to several studies that have indicated 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 adopted to models of learnability as well. However, may other researchers (e.g. Paymans et al., 2004) have noted that sometimes learnability and usability may be contradicting and issues that improve learnability actually reduce usability. Therefore, the conclusion made by Elliott et al. (2002) is an interesting one.

**Learnability as a usability attribute.** Despite the problematic relationship of learnability and usability, learnability has been classified as a sub attribute of usability by several researchers. Jacob Nielsen (2003, p. 26) presents five sub attributes of usability: learnability, efficiency, memorability, errors, and satisfaction. In the SUMI (Software Usability Measurement Inventory) questionnaire, usability has been divided into the sub attributes of learnability, efficiency, affect, helpfulness, and control (Kirakowski, 1996). In another questionnaire, WAMMI (Website Analysis and MeasureMent Inventory), usability has been divided into five attributes: attractiveness, control, efficiency, helpfulness, and learnability (Chambers &

Connor, 2001). Dix et al. (1993, p. 131) in turn divides 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.

**Learnability and product acceptance.** Jacob Nielsen (1993, pp. 23-26) has associated the concept of usability with the wider context of product acceptance. The following figure, adapted from Nielsen (1993, p. 25), clarifies the relationship of factors contributing to system acceptability. Learnability and its top-level concepts have been highlighted in the figure.

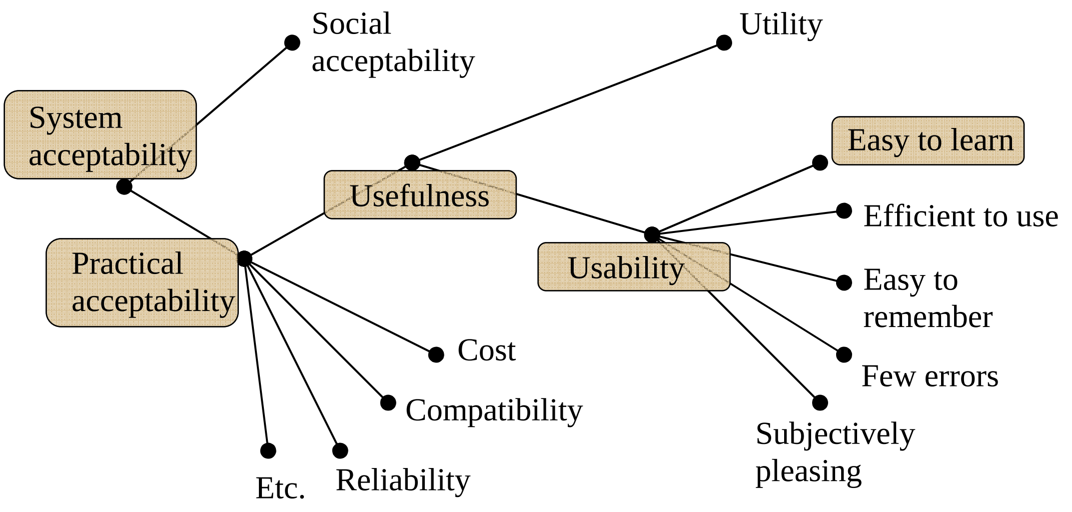



**Figure 3.** Attributes of system acceptability, presented by Nielsen (1993, p. 25)

## Learnability Versus Efficiency

When designing complex systems, it is important to address both the learnability and the efficiency attribute. Users face learnability issues when they start to use the system, but most of them develop into an expert for whom efficiency is the most important learnability attribute.

**Designing for both learnability and efficiency.** Jacob Nielsen (1993, p. 41-42) has stated that fast learning and efficiency are not contradicting goals. On the contrary, a system that is easy to use is often also good for a professional user. For example, adding labels for fields is especially useful for a novice but labels do not cause harm to an expert user either. (Nielsen 1993, p. 42) Sinkkonen, Kuoppala, Parkkinen, and Vastamäki (2002, p. 266) have added that if the functioning of the system is incoherent, terms inconsistent and the user interface unclear, a novice as well as an expert user may make mistakes when using it.

On the other hand, several authors have also speculated on the possible negative effect of learnability improvements on the performance of experts. Goodwin (1987), for example, reminds that novice users do better with a simple interface, but expert users benefit from complexity. Thus, the benefits of complexity must be balanced against the cost of making errors. Another example of a user interface design solution that improved learnability but deteriorated efficiency was found in a study by Paymans et al. (2004) addressing context-aware user interfaces. The outcome of the study was that presenting conceptual information to learners helped them to understand the device better but made them score lower in a performance test.

To avoid negative effects on efficiency when aiming at improving learnability, all design solutions should be compared to the requirements of both novice users and experts. Learnability improvements that cause additional steps into the task sequence or otherwise make the task more time-consuming should not be implemented if the efficiency requirement is important. However, learnability improvements should not be feared either, because may of them will improve the performance of both novices and experts.

**Learnability and efficiency on the learning curve.** The following figure clarifies the goal of designing for both learnability and efficiency. If the system were optimized for only one of those attributes, some part of the learning curve would be unnecessarily low, but if both are taken into account, the learning curve is maximally high.

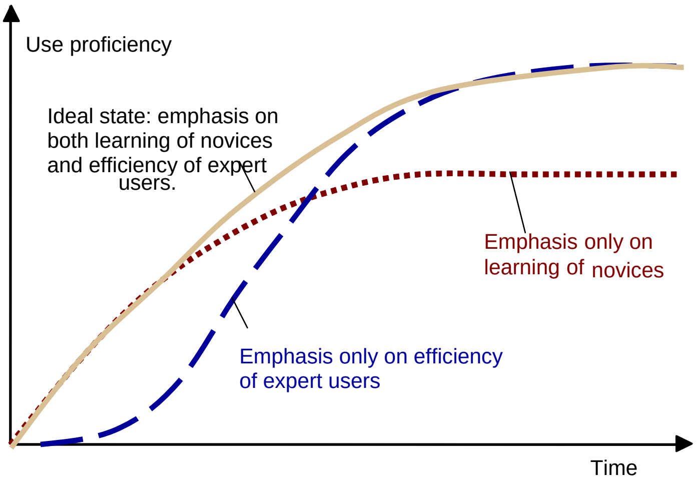



**Figure 4.** Learning curve. Adapted from Nielsen 1993, p. 28 (the solid curve was added).

Learnability can be connected to the left part of the learning curve. For systems having good learnability, the learning curve rises steeply from the beginning, which means that use proficiency develops quickly over time. For systems having poor learnability, the learning curve stays on a low level for a certain period of time, which means that proficiency develops very slowly. Efficiency is usually considered to refer to the right part of the learning curve. Usability can be considered to encompass the whole learning curve, and good usability means that the learning curve is high at all time instances. (Santos & Badre, 1995)

## Aspects of Learnability

Learnability studies have often concentrated on the effect of user interface design on learnability (see e.g. Lin et al., 1997; Elliott et al., 2002; Santos & Badre, 1995). Naturally, the user interface is crucial for learnability as the link between the user and the system is essentially the user interface.

However, in addition to considering the effect of the user interface, the system structure affects learnability on a deeper level as well. The deeper level means the system structure that includes the scope, underlying concepts, and basic functionality of the system. These are often designed before even starting to design the user interface. For software applications that have been available for several years or even more than a decade, the scope, underlying concepts, and basic functionality may have remained almost the same during the years. The user interface has probably followed the trends and standards in user interface design. However, the system structure has a profound impact on the experience of a user that is not familiar with the system. If the system structure differs radically from the systems that the user is familiar with, he faces difficulties when trying to understand and use the new system. This separation between user's expectations and the actual system structure can be explained by using the concept of mental models.

In order to have an understanding of the learnability of a system, it is also necessary to understand the basics of the human learning process because its characteristics will have a profound impact on the learning result. In the case of Tekla Structures, this is even more important, because almost all new users attend a basic training course. Information on the dynamics of the learning process can be utilized when analyzing the training and its effect on the learning results.

In this study, we concentrate on analyzing the three aspects that were mentioned above. The aspects are later referred to as user interface, system structure, and training. The following figure illustrates our approach to learnability.

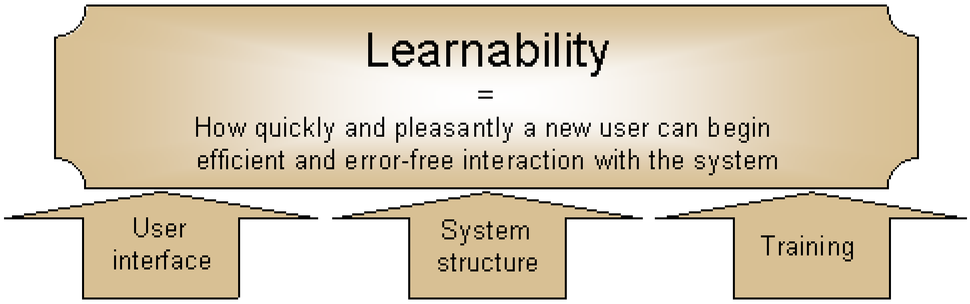



**Figure 5.** Learnability aspects.

The fact that learnability research has often concentrated on the effect of user interface design alone probably arises from the traditions of usability research. Learnability is considered a usability attribute and usability attributes in turn are connected with user interface design. It has been recognized by some researchers (e.g. Lin et al., 1997) that user interface learnability is closely connected to the human learning process as well, or that learning causes changes in mental models of users. However, these issues have not been commonly included in learnability studies but have been researched separately by pedagogical and cognitive scientists. In this study, we aim at taking the user interface, differences between the mental models of users and the system structure, as well as user training equally into account.

## 2.2 Learnable User Interfaces

### Theories of User Interface Learnability

The user interface is an important factor in determining the learnability of a system, even though not the only one, as stated in the previous chapter. Several researchers' conceptions of issues that affect user interface learnability are presented in this chapter.

**Classifications of learnability factors.** Dix et al. (1993) have presented five principles that support learnability. The principles are predictability, synthesizability, familiarity, generalizability, and consistency. Unfortunately, Dix et al. do not tell how these five principles were constructed. Another set of five principles supporting learnability was constructed by Haramundis (2001). Her classification was originally meant to describe learnable instructional documents, but it can be applied to computer systems as well. The adjectives that she considers to describe learnable material are memorable, logical, reconstructible, consistent, and visual. Following either of these sets of principles will help to design learnable user interfaces. The principles are on a very abstract level, however, and applying them to real systems probably requires some expertise on usability issues.

Elliott et al. (2002) present one more abstract conceptualization of learnability, but they bring it to a more concrete level by providing examples of user comments and observations on which the conceptualization has been based. The four factors that they found to determine the learnability of a system were transparency of operation, transparency of purpose, accommodation, and accomplishment. Several sub-factors for each of the four factors were also presented. There are causal relationships between the factors: transparency of operation and purpose lead to accommodation, which is a determinant of the sense of accomplishment.

According to Elliott et al. (2002), their conceptualization of learnability is not necessarily domain-independent as it was acquired by researching only hypermedia authoring systems. However, it is one of the rare conceptualizations of factors affecting learnability and therefore valuable. In this study, we strive to present a corresponding classification for the learnability of a building modeling system.

**Guidelines for learnable user interfaces.** In addition to conceptualizations that present learnability as consisting of a set of sub factors, several researchers have presented guidelines for designing learnable user interfaces. Rieman, Lewis, Young, and Polson (1994) have stressed the importance of consistency in determining the learnability of a user interface. They presented four guidelines for designing user interfaces that follow this consistency principle. Analogies should be used but only if they are inside the context of the program or its class; graphical cues should be provided that indicate the categories that have similar functionality; labels should be designed to link the control to its effect; and clear and immediate feedback should be provided. Green and Eklundh (2003) that studied the learnability of human-robot communication in turn stress the naturalness of interaction in determining learnability. They wrote that different user interface should use similar interaction strategies in order to facilitate easy transfer of learning, immediate feedback that may happen in a conversational sequence is necessary, and lifelike characters should be used by the robot as they enable even a first-time user to understand the messages of the robot.

**Checklists.** Several questionnaires and checklists exist that are designed to support the analysis of system usability or learnability. The questionnaires and checklists may include various user interface items and requirements for how they should be designed to be learnable. In the Purdue Usability Testing Questionnaire (see Lin et al., 1997) for example, learnability has been addressed through seven questions. The questions present a concrete though a rather narrow framework for learnability. The questions for evaluating system learnability are:

- Does it provide clarity of wording?
- Is the data grouping reasonable for easy learning?
- Is the command language layered?
- Is the grouping of menu options logical?
- Is the ordering of menu options logical?
- Are the command names meaningful?
- Does it provide no-penalty learning?

**Connections between theories.** It can be concluded that several conceptualizations of factors affecting learnability exist. They all have issues in common, but as they have been constructed for different domain applications and with different experimental setups, the classifications are partly different. Differences may also arise from different terminology being used for nearly same issues. For example, the learnability factor of predictability presented by Dix et al. (1994) is very close to the transparency of purpose presented by Elliott et al. (2002). It can also be noted that both Rieman et al. (1994) and Green and Eklundh (2003) included immediate feedback in their guidelines. The importance of feedback can be easily reasoned and has been proved in many studies, but we have noticed that in practice, systems often neglect to provide enough feedback, and therefore the need for feedback cannot be stressed too much.

## Elements that Support User Interface Learnability

**Following usability guidelines.** As Elliott et al. (2002) and Whiteside et al. (1985) among others have stated, learnability and usability are congruent. Many researchers have classified learnability as one of the usability attributes (see section 2.1). Therefore, it can be assumed that following common usability principles will also improve the learnability of a system. An example of well-known used usability guidelines are Nielsen's ten heuristics (see Nielsen, 1993, p. 20). Some of the heuristics, such as simple and natural dialogue, speaking the user's language, consistency, feedback, and preventing errors, can be understood to affect learnability, and therefore following the heuristics can be reasoned to improve learnability as well. Shneiderman (1998) presents a corresponding checklist, *eight golden rules* for usability. Consistency, feedback, error prevention were mentioned by him as well as by Jacob Nielsen (1993, p. 20).

**Following learnability guidelines.** Some classifications of learnability factors were presented above. They can be used for analyzing and improving the learnability of a system. Rieman et al. (1994), for example, emphasized consistency as a learnability principle and presented four guidelines for supporting consistency in user interface design. Green and Eklundh (2003) in turn emphasized the naturalness of interaction. They stated that different user interfaces should use similar interaction strategies and that the system should give a sufficient amount of feedback to the user. The four learnability factors presented by Elliott et al. (2002) can also be compared to the user interface design. The user interface elements that do not conform to the factors may need to be redesigned.

Van Welie, van der Veer, and Eliens (1999) have presented a layered model of usability that contains four levels: the usability level, the usage indicator level, the means level, and the knowledge level. According to them, learnability is one of the usage indicators that can be observed when the user is working with a system. They present two means for improving learnability: consistency and task conformance. Consistency means that similar elements are treated in a similar fashion. Task conformance means that the system supports the tasks that users would like to do with it and does it in an understandable manner.

**User support.** In addition to designing user interface elements to follow the guidelines for usability and learnability, several support methods can be used to aid the learning. The support methods may be separate from the user interface itself.

One method for supporting the learnability of the user interface is to integrate user support functions into the user interface elements. There are several possible solutions for this, ranging from messages, tooltips, and balloons to online help systems that are directly accessible from the user interface. Dix et al. (1993, p. 403-405) has addressed several issues that are relevant to integrating user support into the user interface. One of the issues is especially important for the system that this study concerns. As there are big differences in the skill level of novices and experts, there is a danger of the user support suitable for a novice to be obtrusive to an expert. Dix et al. (1993, p. 404) suggests that there should always be the possibility to turn the user support on or off. This idea could be developed further by providing separate advanced and beginner modes that provide different levels of support. McKita (1988) stresses the importance of user analysis in developing support systems that are appropriate for the user group in question.

John M. Carroll and Caroline Carrithers (1984) introduced the concept of "training wheels" for supporting learning. In a training wheels user interface, advanced menu options and commands are not available in the beginning of the learning process. If the user tries to access them, he gets a constructive message that tells that the command is not available on the training system and possibly gives instructions for continuing with the task that the user had started to do. The training wheels interface was observed to shorten the learning time, as the users did not need to spend time with recovering from some common errors related to the advanced functionality of the system.

## Methods for Evaluating User Interface Learnability

Next, we present some research methods that have been used for addressing the learnability of user interfaces. There are studies that aim at measuring learnability with quantitative attributes such as task time and number of errors. Quantitative attributes can be used determining whether the system meets predefined learnability criteria, for comparing different designs, or for comparing software applications. There are also qualitative observational studies that aim at finding the issues that affect the learnability of a system or noting the issues that should be changed to improve learnability.

**Time and error measurements.** In a study by Butler (1985), learnability was operationalized by preparing a test task, equipping users with a manual, and asking them to proceed through a task. Time and errors were measured, and an acceptable level was defined to be less than 180 minutes' average time for completing the task and no users encountering problems that they could not solve by themselves.

In another study presented by Roberts and Moran (1983), learnability was studied by teaching users individually and, after each topic, testing whether users could perform the task individually. The tasks that users could perform were counted. If a user performed a task incompletely or had to look at instructions, half a credit was given. A learning score was calculated by dividing the amount of time taken for the learning session by the number of tasks users could perform.

**Qualitative observations.** If the goal of the experiment is to find issues that support or hinder learning, a qualitative user observation may be the most suitable. The experiments done by Elliott et al. (2002) that were described in the previous section are an example of qualitative observational studies. Another example of a qualitative observational study is the learnability test organized by Dykstra-Erickson and Curbow (1997). They wanted to evaluate the learnability of a document management system to improve its design. They asked ten users to work through 39 tasks and based on the observations, they could identify things that users considered easy or difficult to learn.

Carroll and Carrithers (1984) describe one more learning study. They aimed at assessing the concept of training wheels. They executed the test by bringing the subjects to use word-processing software that they had never used before and asking them to learn it by reading a self-study manual and type in a certain text as fast as possible. Carroll and Carrithers report that their experimental approach was observational concentrating on the qualitative differences between the learning events with and without the training wheels functionality. However, they also measured time and errors and used this information to analyze learning differences further.

**Longitudinal studies.** Jacob Nielsen (1993, pp. 29-30), suggests that system learnability should be evaluated with a longitudinal study containing several observational events. His approach to evaluating learnability was already discussed in section 2.1. Santos and Badre (1995) also recommend evaluating learnability over an extended time period. This is necessary in order to assess a sufficiently long part of the learning curve. Evaluating learnability by observing only first-time users produces results mainly on the intuitiveness of the system and does not give a holistic picture of learnability.

It can be concluded that learnability has usually been researched by either doing quantitative measurements on time and errors or qualitative observations on user behavior, or a combination of these both. Not all researchers have organized longitudinal studies, but it would be advisable.

## *2.3 Mental Models*

### Theory of Mental Models

To make a complete assessment of system learnability, we need to consider also the internal processes of the user and evaluate how closely the system matches users' expectations. According to experts (e.g. Kellogg & Breen, 1988), the closer the actual system structure matches users' internal representation of it, the less errors users will make. We approach the issue of comparing users' expectations and actual system structure through the concept of mental models.

**Concept of mental model.** Mental models are internal representations of entities with which we interact. According to Robert Fein, Gary Olson, and Judith Olson (1993), mental model of a computerized system may contain information on system functionality, components of the system, how each component influences each other, related processes, and their interrelations. Borgman (1999), among others, suggests that the theory of mental models can be used to explain the cognitive mechanism for representing and making inferences about a system when learning to use it.

The concept of mental model was introduced in the beginning of 1980s (Halasz & Moran, 1983). It is nowadays widely used by human-computer interaction experts (Borgman, 1999). The term conceptual model is also sometimes used to refer to the same idea (see e.g. Chandra & Blockley, 1995). At other times, the term conceptual model is used to refer to the underlying system model that should be communicated to the learner (see e.g. Shayo & Olfman, 1998). In this study, we use the term mental model to refer to an internal representation, and the term conceptual model to refer to the underlying system model.

Shayo and Olfman (1998) have identified three purposes for mental models concerning system structure. They base their proposal on a literature review. Firstly, mental models guide the user in planning the behavior of the system. Secondly, they help the user interpreting the behavior of the system. Thirdly, they help the user to form accurate mental models that he can use to perform correctly in a problem situation. If mental models really help in all these issues, which we believe, it is important to support correct mental model formation.

**Learning as a change of the mental model.** Learning can be viewed as a process in which the user processes information and as a result, his mental model changes. According to Chandra and Blockley (1995), mental models are based on knowledge that is obtained from outside sources, on observations and experiences that a human has, or on a combination of these two. Mental models change constantly when knowledge, observations, and experience are gained.

The concept of mental model is especially useful in explaining how humans learn to use complex systems. Such system may contain hundreds of functions and using each function may require dozens of steps. A brute memorization of all the steps would be an enormous cognitive task. However, as the user learns some general operating principles for the system and learns to use certain functions, he will form a mental model that helps in memorization and generalizing the knowledge to new functions. (Halasz & Moran, 1983)

**Erroneous mental models.** The learner has a mental model of the system even before he starts to use the system. However, the mental model is often very imperfect. This hinders the learning process and causes errors, as the user expects the system to perform differently than it actually does. (Kellogg & Breen, 1988) However, users may have incorrect mental models even after they have used the system for a certain period. Firstly, the incorrect assumptions can remain as a part of the mental model if they are not replaced by new information. Secondly, according to Vosniadou (1996), misconceptions can also develop during the learning process. The learner may try to combine incorrect assumptions with contradicting new information. The assumptions can result from previous experiences that are actually not applicable to this situation but the learner assumes they are. When the learner combines the assumptions with new information, he produces a synthetic mental model that explains the contradictions in a wrong way but creates an illusion of a complete mental model.

## Elements that Support Correct Mental Model Formation

**Designing for mental models.** A fundamental method for improving the match between users' mental models and the system structure is to research user needs, preconceptions, and terminology that is familiar to them before designing the system. The system structure, functionality, and terminology should be designed to correspond to users' expectations for it. This will lead to less learning difficulties and less errors during the learning period. (Kellogg & Breen, 1988)

**Communicating the conceptual model to the user.** Sometimes it is not possible to design the system according to the user’s mental model but the underlying conceptual model of an existing system should be conveyed to the user. It is stated in the part 10 of the ISO 9241 standard that the user should be able to obtain information on the model on which the application is based. The following citation is taken from the standard:

> "Rules and underlying concepts which are useful for learning should be made available to the user, thus allowing the user to build up his/her own grouping strategies and rules for memorizing activities." (SFS-EN, 1998)

One method for making the conceptual model of the application explicit is using metaphors from the real world (Nielsen, 1993, s. 127). Well-known examples of this strategy are user interface windows, folders, and desktops. They have been developed to correspond to real world items, even though few people associate them with those any more. (Sinkkonen et al., 2002, s. 252) Also making the relationships between user actions, phenomena, and results visible helps the user to understand the conceptual model of a system. In addition, the user must be given feedback about his actions, in order for him to be able to adjust his conceptual model correctly. (Sinkkonen et al., 2002, p. 287

**Providing explicit conceptual models.** However, there has been discussion on whether users build a mental model spontaneously with the help of hints given by the user interface, or whether they should be provided conceptual explanations that can serve as a basis for a mental model (Borgman, 1999). Several studies have indicated that providing an explicit conceptual model to learners improves learning results. Fein et al. (1993) studied users learning to use a complex ecosystem modeling system. They noticed that it is useful to provide the user textual or graphical information on the conceptual model of the system. Users that read a document describing the hidden interactions in the system before learning to do tasks with it performed better in control tasks than users that learned to use the system with rote memorization of procedures. In addition to comparing the conditions with or without conceptual models, Fein et al. (1993) tested two kinds of models: an explicit model containing some facts about the hidden interactions, and a full model that described the dynamics of the ecosystem in the form of a story. These two models caused no difference in learning results.

Sein and Bostrom (1989) in turn compared abstract models that present the system as a relational structure such as a schematic diagram and analogical models that present the structure of a known object and show how it relates to a new object. They found dependencies between certain learner characteristics (abstract versus concrete learners) and the type of conceptual model that learners preferred. Another study that compared different kinds of conceptual information was done by Shayo and Olfman (1998). They compared conceptual information that was presented in a narrative form to one that was presented as a table containing action verbs and system functions. However, their study did not reveal significant difference between those types of information.

It can be concluded that there are several methods for presenting a conceptual model to the user when he is learning to use the system. If the user interface communicates its structure to the user through analogies, metaphors, textual information, or descriptive visual design, the need for a separate document describing the conceptual model diminishes. As stated above, mental model formation can, however, be supported by providing explicit mental models to the user.

## Methods for Assessing Users' Mental Models

Information on the mental models that users have before using the system or after being trained to use it is beneficial for improving the system and for planning training activities. However, as mental models are internal representations of entities, it is not easy to get information on them.

Multiple techniques have been introduced for studying users' mental models. Next, we present some of the techniques. According to Kellogg and Breen (1986), the technique for deriving the model is critical, as different techniques will produce different information on mental models.

**User observations.** It is possible to use the traditional usability testing techniques such as a scenario-based test with think-aloud protocol for researching mental models. The comments made by the user will reveal some details of his mental model. Errors made by the user can be assessed to find the differences between the user’s mental model and the model corresponding to the system. Users can be observed in their work environment and usage scenarios can be formed with the aim of understanding the goals, concepts, and terminology of the users. Users can be interviewed to hear the terminology they use. (Sinkkonen et al., 2002, p. 245-247)

**Free association.** It is also possible to use specialized tasks to collect information on mental models. Nielsen (1993, p. 127) suggested that users could be asked to associate concepts freely, beginning from the system and getting to terms that are related to it. The concepts that are mentioned after each other are expected to reside close to each other in the user’s mental model.

**Hierarchical clustering analysis.** Kellogg and Breen (1986) suggested a method called hierarchical clustering analysis for studying mental models. In their study concerning a text editing system, participants were asked to group system functions that were written on index cards. The grouping was compared with the actual system structure. The study revealed that the experts' grouping was closer to the system structure than novices' grouping.

**Teachback procedure.** Van der Veer and Bamossy (1990) researched the mental models that students had after being taught to use different operation systems. They had an interesting experimental method: they asked students to explain to an imaginative friend how to perform a certain task with the system. This method provided information on the style and level of representation as well as completeness and correctness of mental models that the students had.

**Quizzes.** Paymans et al. (2004) reported on assessing mental models with a paper-and-pen quiz. They studied the learnability of context-aware user interfaces and wanted to determine the effect of user support on mental model formation. Users were divided in two groups, with and without user support. They were asked a set of questions that measured the degree of understanding the users had about the interface. The questions were asked before and after interacting with the system.

It can be concluded that there are various methods for studying users' mental models. They all aim at making users' thoughts explicit. This requires creativity, because it is not necessarily easy for users to verbalize their thoughts. If information on a certain aspect of their mental models is needed, a research method that addresses the particular issue needs to be designed.

## *2.4 Human Learning Process*

### Theories of Learning

To understand the requirements for a learnable system, efficient training methods, and good instructional documents, we now discuss the learning process of humans. Multiple theories of learning exist, developed by different schools of scientists. Some of the theories overlap partly with each other. The constructivist theory that is based on cognitive science is nowadays widely accepted (Lonka & Lonka, 2001) and it is also a good foundation for analyzing the learning process of Tekla Structures users. Next, we present the foundations of the constructivist learning theory and describe briefly some other learning theories.

In section 2.3, learning was defined as a change in the mental model of a user. The theories of learning described below, e.g. the constructivist theory, and the theory of mental models, are actually different aspects of the same issue. The learning theories described below describe the dynamics of the learning process mainly from a pedagogical point of view, whereas the theory of mental models is closer to cognitive science and looks at the learning process from an internal perspective.

**Constructivism.** Constructivism is based on the cognitive learning theory that emphasizes the internal thought processes of the learner as opposed to the external stimuli emphasized by behaviorism (Sinkkonen et al., 2002, p. 269). Constructivism adds to the cognitive theory by stating that knowledge is never independent of the human that has the knowledge and the situation where the knowledge was acquired. According to constructivism, the learner adopts knowledge and combines it with his previous knowledge to form a more accurate model of the subject. Constructivism contains many different research orientations, such as individual constructivism that stresses mental processes of individuals and social constructivism that stresses group interaction for defining concepts (Marton & Booth, 1997, p. 6-8, 12).

Constructivism has many implications to teaching and pedagogy. The learner is seen as an active information processor. The previous knowledge of the learner is the basis for learning new things and the context of learning affects the learning result. A requirement for effective training is that the instructor takes the previous knowledge of the learners into account by building links between the old and new information. (Sinkkonen et al., 2002, p. 269) The process of accommodating new information is different for each learner and therefore learners are encouraged to develop their own learning strategies. As facts must be combined with existing knowledge, understanding is viewed as a better learning strategy than memorizing, and problem solving is more effective learning strategy than rote learning of facts. As constructivism sees learning and knowledge individualized and relative, different interpretations of facts are accepted and even valued. The goal for all learning is knowledge that can be transferred to different situations. (Tynjälä, 1999, p. 60-67)

**Other learning theories.** One of the older research branches is behaviorism that was popular until the 1950'. It viewed learning as a result of external factors and stated that learning could be controlled with reinforcement and punishment (Marton & Booth, 1997, p. 5). One of the problems of the behaviorist conception is that it sees learning mainly as transformation of information and not active processing of it (Tynjälä, 1999, p. 29-31).

The information processing theory started to get popular in 1950's. It views the human learning process comparable to computers' information processing functionality. The theory stresses the path of information through the memory stages: first information goes to the sensory memory, then some of it goes to working memory for further elaboration, and sufficient repetition or active processing may cause it to be stored in long-term memory. The information processing theory has stressed repetition as an effective teaching method. The contradiction between the repetition method and most teaching events is that in the latter, the aim should be not to learn single facts by repeating them but to connect the material to existing knowledge so that it can be applied in a wide variety of situations. (Tynjälä, 1999, p. 31-37)

Recently, humanistic psychology has brought its views to learning theories. Humanistic psychology contains many orientations. In general, humanistic psychology stresses the particularity of each person and the goal of self-fulfillment. It suggests that the learning process should contain creative exploration and self-reflection of the learner. (Sinkkonen et al., 2002, p. 269)

## Elements of Skill Learning

Learning skills differs from learning knowledge in many ways. In this section, we discuss some elements that are necessary in the skill learning process: acquiring conceptual information, practicing procedural skills, and practicing error recovery. There are many other issues that could be discussed here too, but on the basis of a literature review, we selected these three central issues.

John R. Anderson has divided skill acquisition into three phases that support the notion of conceptual knowledge and procedural skills being interrelated. He defined the first phase of skill acquisition to be the cognitive stage. The learner stores to memory a set of facts that are relevant to the skill. In the second phase that is called associative stage, the learner starts to form the facts into a procedural model. The procedural model contains step-by-step instructions for performing a certain action. In the third phase that is called autonomous phase, the procedure becomes more automated and rapid and in the end require very little processing resources. In short, conceptual knowledge develops into an efficient skill when it is practiced. (Anderson, 1980, p. 273-275)

**Acquiring conceptual information.** The constructivist conception of learning was presented above: learning means acquiring new information and combining it with previous knowledge to form a revised model of the subject to be learned. As we have taken this theory as the basis of our research, we have to consider the fact that learning requires active processing of information to adopt it as a part of the mental model. For skill learning, this means that mere repetition of a procedure is not enough but the conceptual background needs to be understood as well. Many authors have recognized this issue (see e.g. Shayo & Olfman, 1998; Everingham & Brown, 1986)

Combining conceptual knowledge and procedural skills is especially important in learning information technology. Using information technology generally requires remembering a lot of terminology, keyboard commands, and action sequences. It is much easier for a human to remember details that are related to a thematic entity and connected to existing knowledge than to remember disconnected facts. (Lonka & Lonka, 1991, p. 99-100)

One type of conceptual information concerns the techniques for basic interaction with the system. To start a successful learning process, learners need to know how to communicate with the system and how to interpret the responses. This information should be understood on a conceptual level even though it should also be practiced with the real system. The techniques for basic interaction are intuitive for an expert user but not for a novice. (Vanderlinden, Cocklin, & McKita, 1988)

**Practicing procedural skills.** From the phases of skills acquisition that John R. Anderson defined (see above) we can deduce that skills must be practiced in order for them to fully develop. Thus, to learn to use a system, the skills must be practiced by doing tasks on a computer. Predefined exercises can be used to guide practicing.

A problem with doing exercises in general is that if the task always has similar structure, the user may learn to solve only problems that are structured in a certain way. The problems that he will encounter in a real work situation will be profoundly different, however. To avoid this, different kinds of exercises could be used and the exercises should be designed to resemble real work situations as much as possible. (Koli & Silander, 2002, p. 37)

One of the traditional exercise types consists of step-by-step instructions that guide the learner in doing a task. Another approach is the explorative learning strategy with a relatively free-formed learning process. Next, we describe these two approaches in more detail. In reality, the learning strategy may naturally be situated somewhere on the continuum between the explorative learning strategy and structured learning with step-by-step instructions, or these two can be mixed.

According to Wright (1988; in Helander p. 636), step-by-step instructions can serve as an example that the user remembers and applies later. Everingham and Brown (1986), for example, report on successfully using the step-by-step instruction method. However, step-by-step instructions can also lead to non-optimal learning strategies. As Gay Vanderlinden and his research colleagues have observed, "most users simply follow them by rote, passively, and, as a result, learn how to successfully use the tutorial, not the system" (Vanderlinden et al., 1988). To avoid this, learners should be encouraged to analyze their actions and work for building a complete understanding of the system

In the other end of the continuum, we have the explorative learning strategy. John Rieman (1996) defined the term 'exploratory learning' as a process in which the user investigates the system on his own initiative, often in pursuit of a real or artificial task. John M. Carroll and his research colleagues have done research on exploratory learning and have developed a material type called 'minimal manual' (see Carroll, 1997). Support for guided exploration and error recognition and recovery are key components in minimal manuals. Minimal manuals are usually much shorter than traditional manuals. However, there are also shortcomings in the exploratory learning strategy and guided exploration material. For example, as the guided exploration material does not contain very detailed information, some users have been observed to fill the gaps with their incorrect expectations (Carroll, 1997). In addition, if the instructional material contains only short directions for self-exploration like the minimal manual does, learners may need help from another person or documentation. (Koivulahti-Ojala, 2001, p. 86-88).

**Practicing error recovery.** Novice and even expert users frequently face problems and make errors while interacting with a system. It is often considered as a desirable goal to avoid errors when learning to use a system (Frese, Brodbeck, Heinbokel, Mooser, Schleiffenbaum, &

Thierman, 1991). However, avoiding errors is not necessarily the best learning strategy but learners should acquire the abilities to cope with errors.

Frese et al. (1991) have done a study in which they compared the performance of users that were guided during the learning process to avoid errors and users that had practiced to manage errors. Error management had been practiced by doing a series of tasks that were relatively difficult and almost unavoidably led to making errors, and then working to fix the errors. In a performance test after the learning session, it appeared that users that had practiced to manage errors could complete a larger percentage of tasks than the other group.

According to Lazar and Norcio (2003), conceptual information can also help the user to recover from errors. In addition, exploring the system without step-by-step instructions may be useful for the same purpose, as it encourages learners to solve problems on their own or by referring to available material.

## Supporting the Learning Process with Training

**Elements of skill learning.** The three elements of skill learning must be present in all training sessions. In the following three paragraphs, we discuss how these elements can be integrated into the training course in practice.

The first element, conceptual information, must be delivered to learners by the instructor. Different types of conceptual information were described in section 2.3 and a suitable one needs to be chosen for the training. The conceptual information also needs to be integrated with other components of the training in a feasible way. Everingham and Brown (1986) gave in their article a practical account of computer training arrangements in the Michigan State University. They noted that conveying basic concepts to students is necessary even though not always easy. In the training they described, a brief overview of each command was always given before demonstrating a command in practice, and hands-on exercises were occasionally suspended to deliver conceptual information. Training material also contained explanations for concepts.

The second element, practicing procedural information, is usually best supported by providing the learners with exercises. The best exercise type may be found somewhere between the two extremes of step-by-step instructions and exploratory learning strategy (see descriptions above). The level of guidance that is needed depends on the characteristics of the learners, the subject to be learned, and the nature of the training course.

It is also useful to practice error management in training. As was mentioned in the previous section, various methods can be used to do this. In a study concerning the effectiveness of training methods for teaching users to recover from errors, Lazar and Norcio (2003) noticed that error management training that concentrates merely on managing errors is not necessarily the best training strategy, but including some activities related to managing errors leads to better learning results. Lazar and Norcio (2003) found the guided exploration training that does not aim at avoiding errors but does not concentrate merely on errors either to produce best learning results. In addition to including problem solving tasks in the training, error recovery can be supported by e.g. including rescue information in training material, introducing all the reference material that is available, presenting methods to find relevant information, and explaining the most common causes of errors and ways to recover from them.

**Focus on task sequences.** The organization of training topics should be logical and proceed from easy ones to more difficult. According to Koivulahti-Ojala (2001, p. 113), training should start from the basic tasks and advance to the tasks that are dependent on the basic tasks. Users must be able to understand the material on the basis of what they have learned earlier in the training. In addition, training should be organized functionally rather than structurally, which means that it draws from the task sequence of the user and not the internal structure of the system. This sounds self-evident but in practice, it has often been forgotten.

**Meeting the needs of users.** It is important that training content corresponds to tasks and needs of users. McKita (1988) stresses the importance of user analysis (gathering demographic data) and task analysis (identifying major tasks and learning phases) for planning the content of instructional documents or training. Ryan Nelson, Ellen Whitener, and Henry Philcox (1993) have presented a framework for assessing the training needs of individuals, subunits, and organizations. This framework suggests that the characteristics of each learner, their tasks, and the organizational culture should be addressed when planning the contents of training.

The importance of assessing user needs is especially true when it comes to adult learners. Sheila Kieran-Greenbush (1991) has noted that an important characteristic of adult learners is self-directedness: adult learners attend to what they feel relevant to their needs and neglect issues that are not salient to them. If user needs have not been researched, adult learners will most probably note the discrepancies between their needs and training contents.

Shayo and Olfman (1993) have done research on the effect of motivation on software training and software usage after the training. Interviews that they conducted with 19 users led to several recommendations for trainers. One of the recommendations was that users should be involved in defining the learning goals because that will improve the match between their expectations and the actual course contents and lead to a rise in motivation. As noted above, adult learners may be so confident of their needs that they will skip the issues that they feel are irrelevant to them (Nelson et al., 1993). If the learners feel that the training contents are consistent with the requirements of their job, they will also more probably continue using the system after the training (Shayo & Olfman, 1993). The motivational factors may be especially significant if the learner can either stick with an old system or move to a new one after receiving training on the new system.

## Supporting the Learning Process with Instructional Documents

Several classifications for types of instructional documents have been presented and we introduce one possible classification here. Our classification divides materials into these five groups:

- printed documentation,
- electronic documentation,
- printed tutorials,
- electronic tutorials, and
- context-sensitive help.

In practice, different material types often overlap; for example, context-sensitive help may be a part of a complete online documentation. However, it is useful to compare the existing selection of material that is available for a system to the classifications to get ideas on what kind of documentation might be missing.

**Printed documentation.** By printed documentation, we mean a reference document describing the functionality of the system. Traditionally, as technical devices and software applications were purchased, the only instructions coming along with them was the printed documentation, commonly known as a manual. Manuals are suitable for searching information on a certain aspect of the system or solving problems that have occurred during usage. (Dix et al., 1993, p.407-408) System features are often described in the order of presentation in the user interface and processes and tasks get little attention. That makes it difficult to learn to perform real tasks with the printed documentation. Printed documentation also seldom contains exercises that are necessary for skill learning. (Koivulahti-Ojala, 2001, p. 110)

**Electronic documentation.** In the computer era, printed manuals have been widely replaced by their electronic counterparts. Electronic documentation is always available on the computer and cannot be lost. (Wright, 1988, p. 636-638 in Helander) However, mere transferring of printed manuals into online form without making any changes is not advisable even though it has been the most common solution of software vendors (McKita, 1988). To fully exploit the possibilities of online documentation, hypertext, various media types, and annotation tools should be added to the documentation (Dix et al., 1993, p. 407-408).

**Printed tutorial.** Tutorials differ from printed documentation in that they are meant for classroom training and self-instruction. They can also be referred to later when the user wants to revise information that he has previously learned. According to Wright (1988, in Helander p. 636), printed tutorials commonly consist of step-by-step instructions for doing task sequences.

**Electronic tutorial.** The simplest electronic tutorials are essentially printed tutorials transferred to electronic form, possibly with some links between related issues added. The most sophisticated electronic tutorials in turn may contain hypertext information, rich content with several media types, alternative modules, interactive exercises, and personalized presentation. These sophisticated tutorials are often referred to with the term e-learning. Between these two extremes, there is a lot of space for variation. (Vanderlinden et al., 1988)

**Context-sensitive help.** Context-sensitive help systems use information on user actions and the interface state to generate appropriate help messages. Context-sensitive help systems are designed to assist users when they are having problems with the task they are trying to do, as opposed to training material that is usually designed to teach the user a new skill. (Capobianco 2003) Context-sensitive help may give information on different levels: for example, a short description of an element that the user is pointing at, description of a dialog box the user has opened, or extended information on a specific task. The information may be presented in a separate window or integrated into the user interface. These all levels are necessary as help may be needed in different situations and learning styles of the users vary. (Preece, 1994, p. 312-313)

## Methods for Evaluating Learning Results

The effect of training on the skill level of users can be measured simply by comparing the skill level after the training to the skill level before the training. Learnability evaluation methods, such as observation sessions with measurements for time and errors, can be used to estimate the skill level. In this chapter, we present some methods that have been used by researchers for evaluating specifically the impact of training.

**Task completion and time.** The purpose of the study organized by Lazar and Norcio (2003) was to compare the effectiveness of three training methods on their ability to find information on the Web: conceptual training, error, training, exploratory training, and traditional training. Participants of the study first attended a three-hour training session, each treatment group receiving one of the four types of training. After that, the participants were given a list of information gathering tasks and one hour to complete them. As soon as they had completed the tasks, they turned in an answer sheet to the instructor. The dependent variables addressed in the study were task performance, which meant the correctness of answers, and task time, which meant the time that was spent on doing the tasks.

Frese et al. (1991) also used task completion and task time as a measure of the effectiveness of the training. They compared a traditional training method that aims at teaching correct task sequences and error management training that additionally aims at teaching error management skills. Users were trained with the specific training method for six hours and after that tested for two hours. In the test, the observer judged how efficiently and correctly users could do tasks that had been practiced in the training and tasks that had not been practiced. In addition, for certain tasks, the speed of completion, the number of errors, and the time that was required for correcting the errors was measured.

**Free recall.** The study of Frese et al. (1991), which was described above, also contained a free recall test. The subjects were asked to state all the commands that they still remembered and explain for what they were used. The correct answers were counted.

**Change in mental models.** Olfman and Shayo (1997) in turn compared the effect of two types of training tasks (one concentrating on the application only and the second connecting it to other relevant applications) and the number of software packages demonstrated (one or two) on the learning results and the ability to transfer skills to other similar applications. They assessed the pre-existing mental model of subjects before the training with quizzes in which subjects had to both explain some principles of the application and answer 'true-false' questions. In addition, the background of subjects and self-assessment of skills were addressed with questionnaires before the training. The training consisted of a video shown to the participants and a demonstration made by the instructor. After the training, the same mental model test was administered that was also run before the training. In addition, the self-assessment of skills was made again. The learning performance was operationalized in terms of changes in subjects' mental models before and after the training.

**Interviews.** Shayo and Olfman (1993) used the interview method to collect qualitative information on the issues affecting the effectiveness of training. The subjects of the study were employees who do part of their job with a computer and they were interviewed about their latest experience of software training. The interview questions concerned users' goals and intentions, self-efficacy and expectancy, individual characteristics, feedback, and support. Recommendations for trainers were derived from the comments that users made in the interviews.

It can be seen that several methods have been used for evaluating the effectiveness of training. Actually, they do not differ very much from the methods that are used in usability testing. However, when the effectiveness of a certain training method is studied, another method is often taken as a reference. When studying the usability of a system, a reference system is not needed if the purpose is to find the problematic elements in the user interface. The problematic elements can be directly inferred by observing users. The issues that make training effective or ineffective cannot be seen as easily. The effect of certain elements in the training is not always immediate but may produce long-term results.

# 3 Building Modeling Systems

## *3.1 Building Modeling Process*

Next, we introduce briefly the building modeling process in which the Tekla Structures system is used. The process involves several parties such as architects, structural engineers, electrical engineers, and HVAC (Heating, Ventilating, Air Conditioning) engineers. The building modeling process starts from the preliminary requirements for the building and continues until the building is being constructed. The following figure shows a typical construction site. The

Tekla Structures system was used for the modeling of the steel and concrete structures on that particular site.


**Figure 6.** Construction site for the shopping center Sello in Leppävaara, Espoo.
(Photograph: Finnmap Consulting Oy, 2004)

The building modeling process is strictly regulated and there are rules that define the documents that need to be produced in different phases of the process. The phases and the required documents are introduced in a booklet called Scope of work in structural design (RT 10-10577, 1995) that has been approved by several associations in the structural design branch and is followed by structural design offices.

According to the aforementioned document, the building modeling process should start with a requirements gathering phase. In this phase, a preliminary analysis of the existing and required premises and costs are made. After the requirements gathering phase, a project planning phase starts and the targeted scale, schedule, costs, and quality of the project are defined. Structural engineers are usually not involved in these two phases. The third phase is called the conceptual design phase. In this phase, strength calculations are made for the preliminary plans, appropriateness of the architectural solutions is estimated, and different solutions for the skeletal structures are compared. Several different software applications can be used this phase. It is possible to use a 3D modeling software but also to use other tools in this phase and start 3D modeling later. After the design conceptual design phase, the detailed design phase is started. In this phase, detailed plans for foundations, skeletal structures, and fabrication are created. (RT 10-10577, 1995) During the whole process, information is continuously exchanged between project parties in the form of drawings and reports. The figure below shows an example of a drawing of a concrete column.

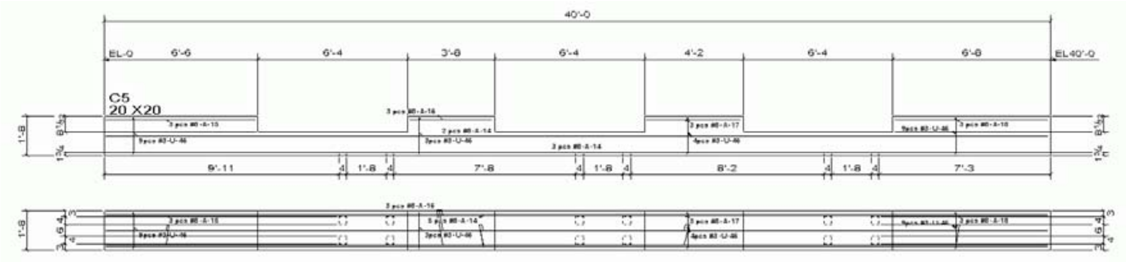



**Figure 7.** Drawing showing a concrete column.

The tasks of structural engineers are often divided among several employees. Senior structural engineers may do the preliminary analysis and plan the design, and a junior worker may produce the model according to the plans.

Research activities aimed at getting a more detailed picture of the phases of the structural design process are introduced in chapter 4.3.

A recent change that has a profound impact on the work processes of structural engineers is the emergence of three-dimensional modeling software. Until now, most structural engineers have used two-dimensional drawing software, often referred to as CAD software, for doing their work. Currently, the structural engineering offices are in the migration phase in which some of their employees have started to use three-dimensional modeling software but others are still using the old drawing software.

The use of two kinds of tools poses challenges to the exchange of information. The software applications need to support data exchange with several other applications. The work of structural engineers involves a lot of collaboration, which makes this even more important.

Change from the two dimensional drawing paradigm to the three dimensional modeling paradigm also makes the learning process challenging. As the nature of the software applications is different, many operations are performed in a different way and many new concepts and operations are introduced in the three-dimensional modeling software.

## 3.2 Tekla Structures Building Modeling System

In this section, the Tekla Structures building modeling system and current training arrangements for teaching its use will be introduced.

Tekla Structures is a building modeling system that is used for steel and concrete design and fabrication. The system can be used in different phases of the building modeling process from the structural analysis and dimensioning to the modeling of details. Tekla Structures is primarily intended for structural engineers, but it supports exchange of information with other parties. The job of structural engineers involves creating a three-dimensional structural model that contains steel or concrete parts and connections as well as material properties and other technical information.

Tekla Structures has been developed by the Building & Construction unit of Tekla Corporation. Tekla Structures is an expanded version of the previous Tekla Xsteel modeling system.

Tekla Structures runs in the Windows environment on a personal computer. Its user interface includes a drawing area, menus, and icons. Below is a screenshot showing a typical interface state.

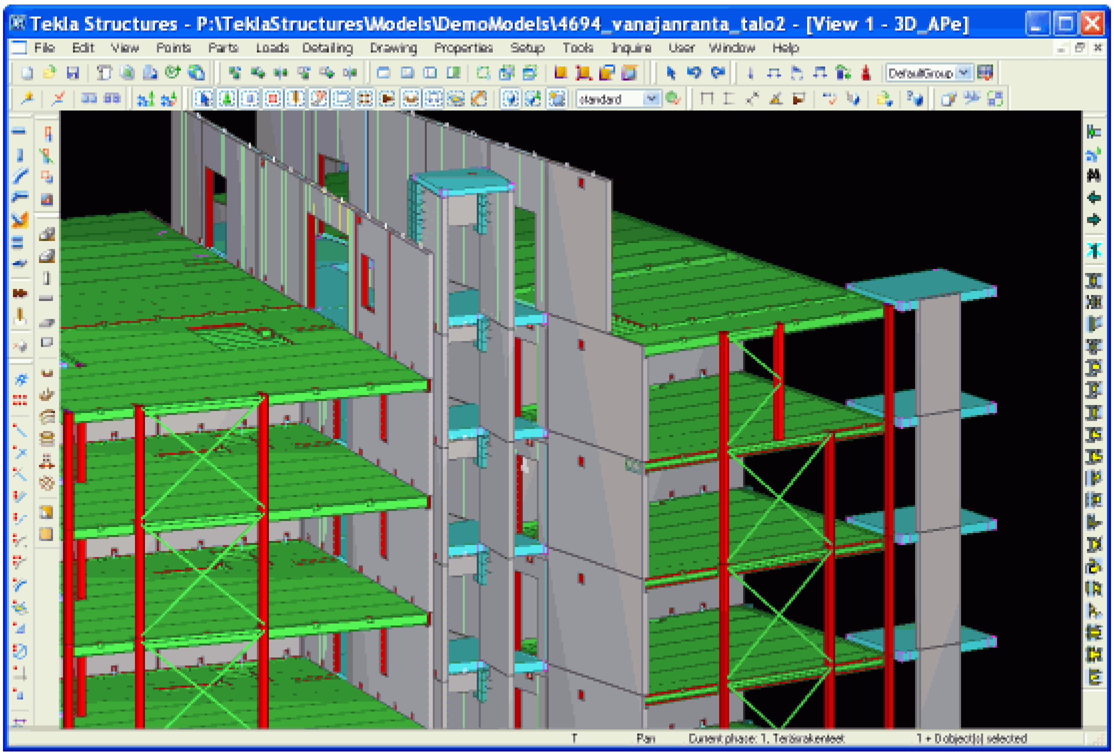



**Figure 8.** User interface of the Tekla Structures system.
(Model: Antti Pekkanen, A-insinöörit, 2003)

The drawing area shows the current model three-dimensionally. Two-dimensional views can also be created from the desired angles and locations. The user can zoom, move, and rotate the model by using the left mouse button and the scrolling wheel.

By clicking the icons on the left, the user can add several types of concrete and steel parts into the model. Some of the icons on the left also enable creating reference points and changing the shape of the existing parts. The icons on the right enable the user to create connections or reinforcements between the parts. Object parameters, such as dimensions and materials of parts, connections, and reinforcements, can be adjusted through a dialog box that is opened by double-clicking an object or an icon.

The icons on the top correspond to different tools and operations such as creating drawings or views, creating loads, moving or rotating parts, adjusting snap settings, or measuring distances, to name a few. All the commands are also available in the menus, and the user can determine himself which menu commands are visible as icons.

As there are a lot of icons and menu options, learning to use the system requires a lot of effort. Improving the learnability of the system would result in a longed-for reduction of learning time.

To support learning, Tekla organizes a training course for new users of the system. The duration of the training varies in different countries, but in Finland, it is three days. In the training, the basic skills for using the system are taught by going through a training material folder. The folder contains detailed step-by-step instructions for constructing a model with steel and concrete parts and creating different kinds of drawings and reports. After the course, the users should be able to start working on a real model in their office. However, the system is very complex and therefore the users are able to learn only a small subset of its features during the training.

# 4 Empirical Learnability Evaluation

## 4.1 Subjects of the Study

To concentrate the learnability evaluation on the core tasks of users, some background research was made before the actual learnability evaluations. Three users that had attended the Tekla Structures training one month earlier, four users that were currently attending the training, and several employees that had worked with customer service several years participated in the background research.

The purpose of the actual learnability evaluation was to assess the initial learning experience of users and therefore, novice users were chosen as subjects. Six users had registered to attend a certain basic training course and their willingness to attend a learnability study was requested. All of them were willing to participate. In the first meeting, they were also asked to sign a written consent for participation (see appendix A).

The subjects had worked on the building modeling branch from 2 months to 28 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 it for one day only.

Below is a table with information on the subjects. Later, the abbreviations U1, U2, U3, U4, U5, and U6 are used to refer to the subjects. To keep anonymity, the subjects are presented in the following table in a random order. Only the participants of the actual learnability evaluations, not the participants of the background research, are included.

**Table 2.** Subjects of the learnability evaluation.

| Profession | Experience with building modeling | Experience with Tekla Structures |
|---|---|---|
| Construction engineer | 2 years | none |
| Construction engineer | 2 years | none |
| Structural engineer | 2 months | 1 day |
| Structural engineer | 28 years | none |
| Designer | 25 years | none |
| Technical drawer | 5 months | none |

## 4.2 Organization of Research Activities

Learnability research was spread over a three months' period so that information on the different phases of the learning process could be obtained. The importance of using a longitudinal study when researching learnability was mentioned in chapter 2.2.

At first, information on core tasks of users was collected by interviewing three users of the Tekla Structures system over telephone, one Tekla employee in person, and arranging three informal group discussions.

Next, users were interviewed to examine their mental models concerning the Tekla Structures system. After that, a three-day training course was observed and the material used at the course was analyzed. Immediately after the course and two months later, users were observed doing a small scenario-based test task and asked to fill in a subjective satisfaction questionnaire.

Unfortunately, one of the six subjects of the learnability evaluation could only attend the research activities that were arranged right before or after the training. He had to skip the second learnability test and the subjective satisfaction questionnaire. The reason for this was the long geographical distance of the subject's office and Tekla headquarters. All other subjects attended all the research activities.

The research activities are summarized in the following table.

**Table 3.** Schedule of research activities.

| Activity | Duration of meetings / observations | Dates |
|---|---|---|
| **Individual interviews** | 4 * 20 min | June 9 – 14, 2004 |
| **Group discussions** | 3 * 20 min | August 15 – 16, 2004 |
| **Mental model interviews** | 6* 1,5 hours | August 13 – 24, 2004 |
| **Training observation and training material analysis** | 3 days | August 24 – 26, 2004 |
| **Learnability tests and subjective satisfaction questionnaires** | 6 * 1,5 hours | August 27 – September 6, 2004 |
| **Learnability tests and subjective satisfaction questionnaires** | 5 * 1,5 hours | October 28 – November 10, 2004 |

## 4.3 Research on Core Tasks

### Method

As Tekla Structures is a complex system designed for domain experts, it includes a lot of functionality, not all of which can be covered in a single learnability study. In order to concentrate on the tasks that are central to new users of the Tekla Structures system, it was necessary to gather information on the core tasks of the users before designing the details of the forthcoming research activities. Information on core tasks was gathered by interviewing three users of the Tekla Structures system over telephone, one employee of Tekla whose work involves customer service, and conducting three informal group discussions during the breaks on a Tekla Structures basic training course. These activites have been highlighted in the following figure that illustrates the progress of our research.

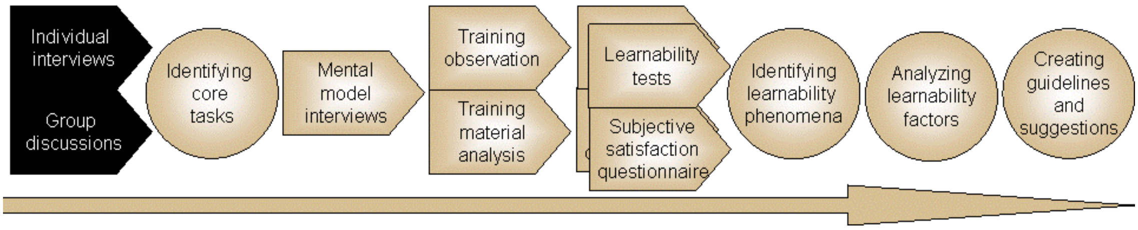



**Figure 9.** Progress of the research activities: Interviews.

An interview form with 21 questions was prepared for the telephone interviews. The questions covered the background of users, the characteristics of their work process, and their opinions on the Tekla Structures system as a tool for performing their tasks. The questions were applied from an interview template developed internally by the Tekla Usability Engineering team (see Tekla Oyj, 2004). The themes of the interview followed the structure outlined by Kujala, Kauppinen, Nakari, and Rekola (2003). They presented a field study method that is intended for researching user needs and is simple and flexible enough in order to be adopted in organizations. The interview form is presented in appendix B. The telephone interview session lasted from 15 to 20 minutes.

The group discussion sessions were free-formed with no pre-planned questions. The researcher joined a group of four participants during a break in the basic training. Information that could be utilized in defining the core tasks was collected by asking questions about the normal workflow of users. In addition, their previous experience with modeling software applications as well as opinions on the Tekla Structures system and the training course were addressed. Discussion on these issues proceeded even without the researcher intervening in it, because the issues were of common interest for training participants. Each interview session lasted about 20 minutes.

One Tekla employee was interviewed with the template that was developed for telephone interviews. He had several years' experience in customer service and thus was familiar with the tasks that belong to customers' work processes. Information on those tasks was also collected from other Tekla employees but without using an interview template.

## Results

The following table summarizes the themes that were extracted from all the individual interviews and group discussions.

**Table 4.** Summary of individual interviews and group discussions.

| Theme | Comments from users |
|---|---|
| **Primary work tasks of users** | The work includes structural planning, creation of drawings or models, strength calculations and other necessary analysis, site meetings, and communication with other project parties. |
| **Data that is available when modeling is begun** | In some structural engineering offices, designers get a detailed room plan and the associated dimensions from an architect, often as a CAD drawing. Designers create the model on the basis of those visualizations. They get the HVAC plan and electricity plan from other parties and discuss with them about how different plans match together.<br>In other structural engineering offices, customers tell only the main dimensions of the building and the designers create the model with a rather small amount of information. If an industrial building is being designed and there are no special requirements, the designers may use existing standard designs or use an existing model as a template and make the necessary changes into it. |
| **Phases of the modeling process** | The modeling process starts from doing analyses for dimensioning. Separate analysis software is often used for this. After that, the model is created with Tekla Structures or AutoCAD. The modeling process may be divided in several phases: drafting phase, project planning phase, and assembly planning phase. The details of the modeling process depend on the software application that is being used. However, the model may contain concrete and steel parts, reinforcements, and connections, depending on the building that is being modeled. A lot of time is spent with modifying the details of the model and the drawings to look exactly how they should. |
| **End result that is given forward** | The designer sends structural drawings, detail drawings, assembly drawings, single drawings and calculations to other parties of the project. These are to guide the manufacturing of concrete or steel parts, construction activities at the site, and coordination between the designers. |
| **Change management** | Changes are very frequent in the building modeling process and they take a lot of time. Designers get change requests from other project parties during the modeling process. |

## Discussion

On the basis of the results, 15 core tasks that the user needs to complete in practically all modeling projects were listed. The following figure shows our progress in the series of research activities.

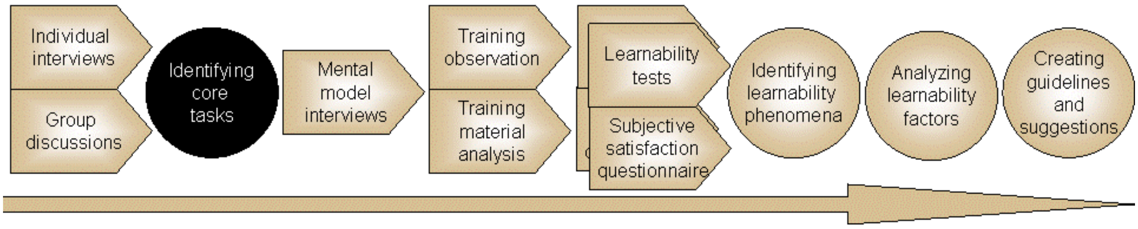



**Figure 10.** Progress of the research activities: core tasks.

The core tasks are listed below and described in some more detail after the list. The listing was used as a basis for designing the details of each learnability evaluation method described in the next sections.

1. Creating views
2. Creating grids
3. Creating concrete or steel parts (columns, beams, slabs etc.)
4. Modifying concrete or steel parts
5. Creating reinforcements
6. Creating connections
7. Saving components (reinforcements, connections etc.)
8. Creating numbering
9. Creating drawings
10. Updating drawings
11. Modifying drawings
12. Creating reports
13. Exporting/importing data to other applications
14. Specifying model properties
15. Modifying catalogs

Creating a new model usually starts with creating a grid along which concrete or steel parts can be placed (task 1), and creating views that show the model in 3D or plane (2). Users did not usually mention these two tasks in the interviews, but as creating views and grids is an essential precondition for starting to create concrete or steel parts, they were included in the list of core tasks.

After creating views and grids, concrete and steel parts are created (3). Concrete parts need to be reinforced (5) and connections need to be created between members (6). These were mentioned in the interviews as an essential part of the modeling process. Components, such as reinforcements and connections, can be saved to be available in later projects (7). This task was derived from the comments that indicated that users often utilize elements that have been created in earlier projects.

According to users, changes in the model are frequently requested by other parties of the construction process and therefore the task of modifying concrete or steel parts (4) was included in the core tasks.

When the model is ready, and also during the modeling process, drawings (9) and reports (12) are created. They are sent forward to engineering works and construction site. Drawings are produced automatically but they usually need to be modified by the designer of the model (11).

If the model changes, the designer can start the automatic updating process for drawings (10). All these tasks were derived from comments made in individual interviews and group discussions. However, before creating drawings or reports, the designer needs to assign numbers for all parts (5). This was not mentioned by the users as other modeling software differ in how numbers are assigned and modified. However, for the Tekla Structures system, numbering is required before creating drawings, and therefore numbering was taken as one of the core tasks.

In addition to sending drawings and reports to other project parties, the model sometimes needs to be exported and imported to and from other file formats (13). The need for communication with other project parties was mentioned by users.

The modeling process also requires defining project properties (14) and managing material and profile catalogs (15). Project properties mean information on the designer and project but also on variables and components used in the project. Catalogs contain the information on the available materials and part profiles. Defining project properties and material catalogs was not explicitly mentioned by users, possibly because these are administrative activities and users did not see them as a part of the modeling process. However, these activities are necessary in order to manage the properties of the model.

The core tasks make it possible to concentrate on the most central tasks in the learnability research. However, it will be possible to generalize some of the results that are acquired by researching these tasks to other parts of the system as well.

## 4.4 Mental Model Interview

### Method

Learnability research was started by interviewing users that were going to attend the basic training. The purpose of this research method was to acquire information on the mental models that users have of the system before interacting with it. This information is useful because the differences between the mental models of users and the system structure may explain the problems that users have later when interacting with the system. This issue was already discussed in section 2.3.

A similar interview method was used Dykstra-Erickson and Curbow (1997). They studied the learnability of a document management platform called OpenDoc. They used user interface prototypes and interview protocol to study users' expectations on how to use the system features.

The following figure shows the position of the mental model interviews among the other research activities.

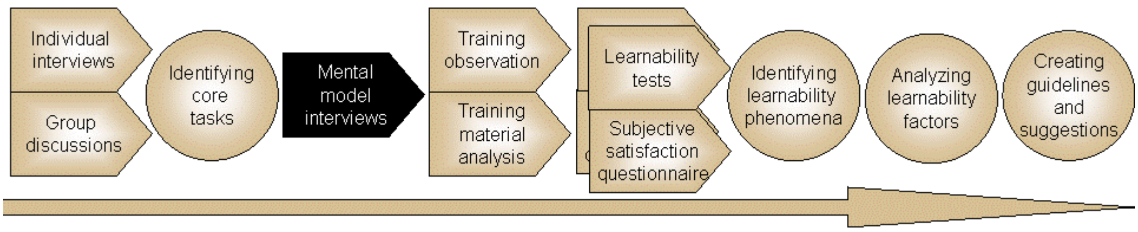



**Figure 11.** Progress of the research activities: mental model interviews.

In the beginning of the interview session, the user was shown the Tekla Structures interface and a new document template as in the figure below. The user was asked questions about interface elements such as icons. After asking questions about the basic state of the user interface, the user was asked how he would perform some basic modeling tasks. The user was allowed to test the procedure he had suggested briefly and if it was not successful, the correct operation was shown. Before the training, users were not asked to perform real modeling tasks, because using the Tekla Structures system is rather difficult without any training.

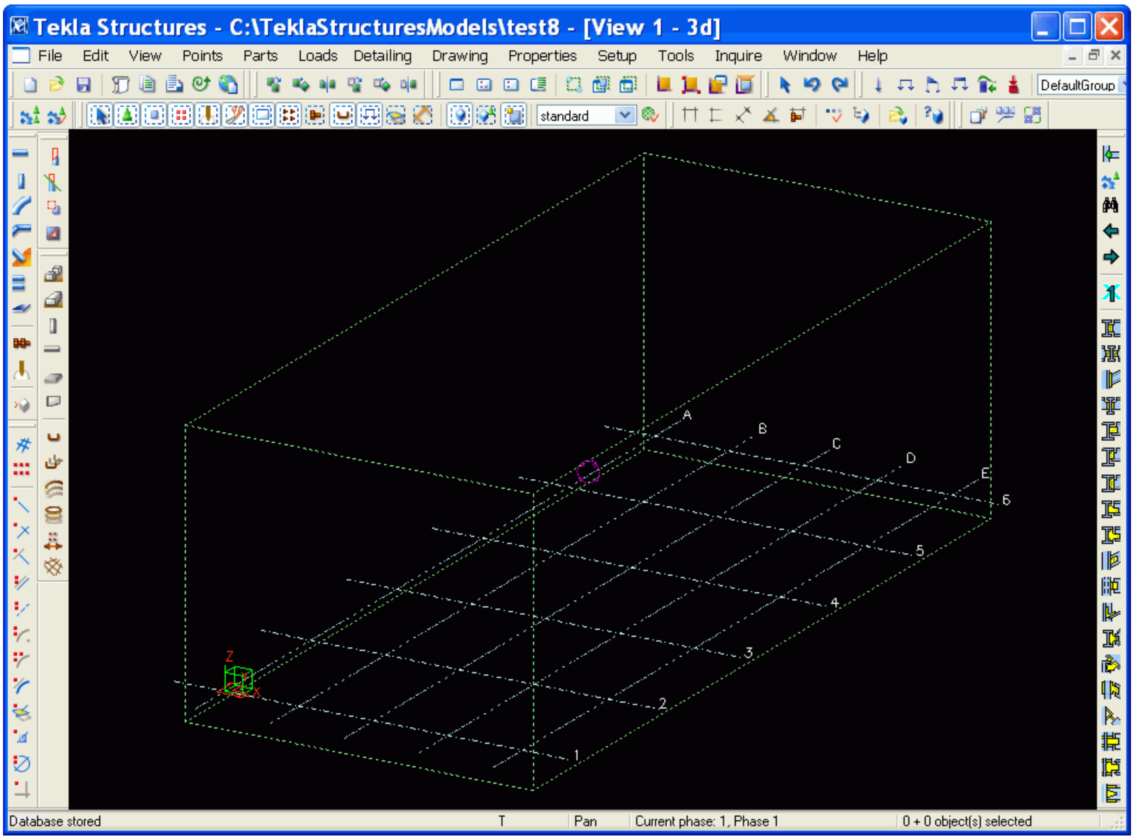


**Figure 12.** The Tekla Structures interface as introduced in the mental model interview.

Below are samples of the interview questions. All interview questions are listed in appendix C.

- Which icons seem familiar to you? What do you think the other icons represent?
- What do you expect to be the biggest differences compared to the software you used before?
- What do you expect the items that you see to be?
- 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 the drawings?
- Where do you expect to save material properties, part profiles and other project specific information?

The duration of the interview was approximately 45 minutes. In some interviews, there was not enough time to cover all of the questions listed in appendix C, but in those situations, a representative selection was chosen by the interviewer.

Mental model interviews were audio recorded and comments were translated to a written form after the interview. This enabled the researcher to analyze the comments in detail. The comments that revealed differences between the mental models of users and the system structure were given special attention. In addition, comments that indicated the mental models of users and the system structure to correspond to each other were noted. In addition, design suggestions that users presented were written down.

## Results

In total, 41 learnability phenomena were extracted from the interview notes. Examples of those phenomena are presented in appendix D. The following table contains a summary of the interview results related to each core task.

**Table 5.** Mental model interview results.

| Core task | Interview |
|---|---|
| **Creating views** | The concept of view was not clear to users as it is different than in two-dimensional drawing software. |
| **Creating grids** | The coordinate system was intuitive to users. However, some users had problems with understanding how grid dimensions should be entered. |
| **Creating concrete or steel parts** | Users could easily place parts in the model without any training. |
| **Modifying concrete or steel parts** | Users had problems with mirroring parts. Some users did not know how to inspect the properties of a part. |
| **Creating reinforcements** | Some users thought that reinforcements can be created on the part properties dialog. They did not easily find the correct method for creating reinforcements. |
| **Creating connections** | Users considered finding suitable connections and difficult. |
| **Saving components** | Users expected that storing connections for later use was simpler than it actually is. They used the term save whereas the current term is "Define custom component". |
| **Creating numbering** | Users did not understand the reason why all parts need to be numbered before creating drawings. |
| **Creating drawings** | Users assumed correctly that drawings are created with templates. |
| **Updating drawings** | The task sequence for updating drawings was intuitive for some users. However, some users did not understand that with this software, drawings and lists are updated automatically according to the model. In traditional drawing, users had to handle all drawings and lists separately. |
| **Modifying drawings** | Users tried to edit the part mark text directly by clicking on it. |
| **Creating reports** | Users said that report templates contain all the necessary information. However, users were confused with the number of report types available. The titles of the report types did not clarify the scope of the report. Users said they would need only a few report types. In addition, users used the term list, not report. |
| **Exporting/importing data to other applications** | Users' expectations for export and import features were consistent with the existing features. The features will probably fulfill their data exchange needs. |
| **Specifying model properties** | One of the users was wondering about how settings affect different files. |
| **Modifying catalogs** | - |
| **General** | Users recognized or could guess the meaning of most icons. The meaning of buttons OK, Apply, Modify, and Cancel was unclear to users. |

### Discussion

Users had rather detailed assumptions about system structure, but as could be expected, their assumptions were partly incorrect. Misconceptions were revealed in the mental model interview.

It was clearly seen in the mental model interview that users based their expectations mainly on the software application they had used earlier. The concept of view is an example of an issue that was difficult to understand, as it was not used in two-dimensional drawing. Users also often mentioned how the requested operation, such as mirroring or modifying part marks, was performed with the software application they were familiar with and expected Tekla Structures to work similarly. This could be anticipated on the basis of the theory of mental models. As was mentioned in section 2.3, humans base their mental models on their previous experiences, which in this case mean experiences with two-dimensional drawing software. Before the training, users had hardly any information other than their previous experiences. They have heard some facts about 3D modeling; however, the information did not cover interface details or task sequences but only general principles of 3D modeling.

It was observed that users could guess the functionality and 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. On the other hand, they could not proceed with the more complex functions such as control connection parameters or drawing layouts without instructions.

Mental model interviews produced a lot of information on users' mental models concerning the system structure and the user interface. This information can be used for making the system correspond to users' expectations better, which in turn will make it easier to learn (see section 2.3). For example, if a remarkable portion of users expects the system to function in a certain way but currently it works differently, there may be a need to change the system to function as users expected. Alternatively, the functionality should be communicated more clearly to users through user interface design and documentation.

The learnability phenomena collected in the mental model interview are essential for defining the learnability factors related to the differences between users' mental models and the system structure. Some of the information can also be utilized for defining the learnability factors related to the user interface and training.

## *4.5 Classroom Training Observation*

### Methods

A basic training course organized for new users of Tekla Structures was observed to acquire information on the learning event itself. The purpose was to see how different features of the system are learned, which functions users consider difficult to learn, and what kind of problems users face when learning to use the system.

In sections 2.2 and 2.4, we presented several observational methods aimed at evaluating learnability of a system or effectiveness of training. In most of them, users were observed when learning to use the system independently. The method of observing training sessions has not been widely used for studying learnability, possibly because formal training sessions are not organized for many software applications but learning takes place informally in organizations. However, training observation was mentioned in the workshop notes of the Computer-Human Interaction 1997 conference to be a good method for collecting learnability data (Karn, Perry,

& Krolczyk, 1997). As training sessions are regularly organized for new Tekla Structures users, we considered training observation to be an easily arranged yet efficient method for evaluating learnability. Most essential system functions are covered in the training, and a three-day-long session enabled us to address more issues than a one-and-a-half hour learnability test.

The following figure shows the position of the training observation in the series of research activities.

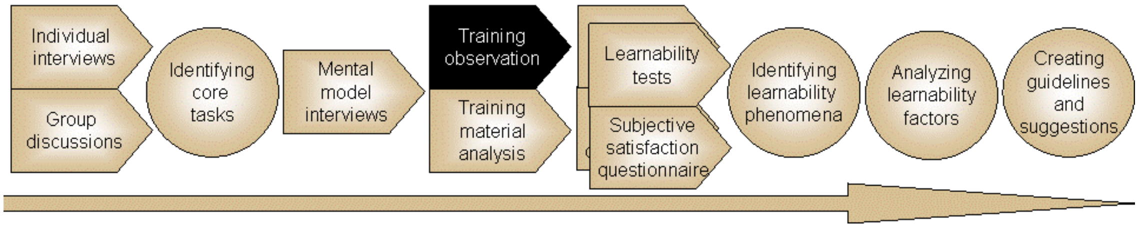



**Figure 13.** Progress of the research activities: training observation.

The observed training course lasted three days. The training group consisted of the six users that had been interviewed also before the training.

The researcher observed the six participants while they were using the system. The observer filled in an observation template containing the fields presented below (for the observation form, see appendix E). The issues mentioned on the observation template were noted for each core task. Fields 6 and 7 are related to training material and results related to those fields are presented in section 4.6.

1. Training topics
2. Teaching methods
3. Time that was spent with each core task
4. Concepts that were explained
5. Concepts that were not explained
6. Chapters in training material
7. References to help material
8. Questions from participants
9. Behavior of participants

## Results

Over 1000 rows on the observation template were filled in during the training from which 111 learnability phenomena could be extracted. 289 questions presented by the users were recorded and they proved to be especially useful for analyzing learnability. However, because of the wealth of information, not all results can be presented here.

A wide variety of topics was covered in the training. They were all related to core tasks. During the three-day training, a model that is close to a real work task was constructed. The model is shown in the figure below.

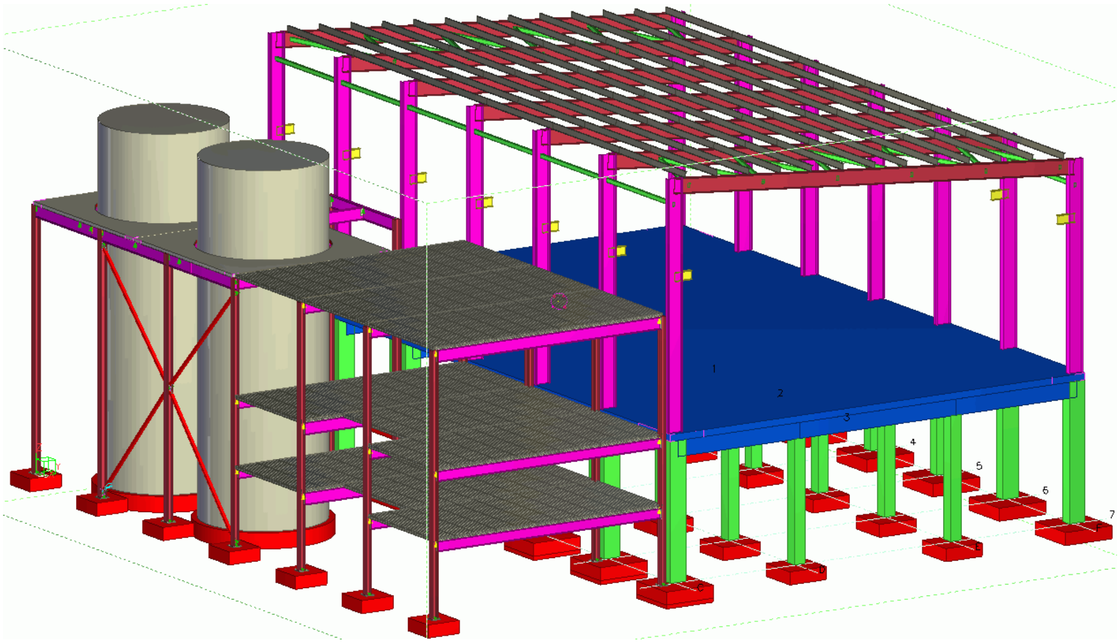


**Figure 14.** The model that was constructed during the three days of training.

For all the tasks, a similar teaching method was used. First, the instructor demonstrated the use of a command on his computer. The computer screen was projected on a wall. After that, users applied the command to the model they were creating on their own computer. There were detailed instructions for using the command in the training material folder, and most users followed the instructions carefully.

The observed group of users was very active and they asked many questions. Users did not always understand all steps of the demonstration and they asked clarifying questions from the instructor. Sometimes there was a confusing element in the user interface whose meaning users asked. Users also faced various problems when doing exercises on the computer and needed help from the instructor to solve these problems. It was noticed that some problems that users faced originated from the fact that they did not understand the meaning of certain commands or objects. Sometimes those concepts had not been explained by the instructor.

Of all the core tasks, most time (282 min) was spent with modifying drawings. A lot of time was also spent with creating concrete or steel parts (148 min) and creating reinforcements (142 min). Exporting and importing data to other applications and specifying model properties were not covered at all in the training. Only a little time was spent with updating drawings (5 min).

The following table contains a summary of the behavior of users and problems they faced when practicing each core task. Representative examples of comments users made during the training are presented in appendix F.

**Table 6.** Summary of user behavior when learning each core task.

| Core task | Behavior |
|---|---|
| **Creating views** | Users had some problems with creating views and some problems with defining the visibility of objects. |
| **Creating grids** | Users could create grids rather easily even though some of them had problems entering values to the grid dialog. |
| **Creating concrete or steel parts** | Users could create parts fast but had some problems with defining part properties and interacting with the model. |
| **Modifying concrete or steel parts** | Users had problems moving and resizing parts. Handling points (which is needed in almost all modify operations) was also unclear to users. |
| **Creating reinforcements** | Users had a lot of problems with finding suitable reinforcing macros, selecting points to place the reinforcements, and defining the properties of reinforcements. |
| **Creating connections** | Users had a lot of problems with finding suitable connections and defining rules for AutoConnections. |
| **Saving components** | Users could create a custom connection rather easily but had problems editing it later. |
| **Creating numbering** | Users said after the demonstration that it is still unclear to them how to define numbering settings. |
| **Creating drawings** | Users succeeded with creating drawings rather well after the demonstration even though they had some questions about it. |
| **Updating drawings** | Users succeeded with updating drawings rather well after the demonstration even though they had some questions about it. |
| **Modifying drawings** | Users had a lot of problems with using the drawing classifier, modifying drawing layout, setting the visibility of objects, and modifying part marks. |
| **Creating reports** | Users succeeded with creating reports rather well after the demonstration. |
| **Exporting/importing data to other applications** | This was not covered in training. |
| **Specifying model properties** | This was not covered in training. |
| **Modifying catalogs** | Users succeeded with modifying catalogs rather well after the demonstration even though they had some questions about it. |

## Discussion

Firstly, the training observation method enabled us to collect learnability phenomena related to the parts of the user interface or the system in general that were difficult to learn. The list of difficult issues contained both user interface details and complete task sequences. A three-day long training observation enabled us to get information on a larger number of tasks than in a one-or-two-hour usability test.

Secondly, the training observation method enabled us to observe the training arrangements and users' response to them. Training is an essential part in the learning process of Tekla Structures users, and to assess learnability, we need to assess training as well.

It can be concluded that the learnability phenomena found in the training observation contribute to finding the learnability factors related to training, user interface, and system structure. After distinguishing the learnability factors, we can use the collected information for creating suggestions for improving learnability.

Changing the teaching methods and contents of the training would help users to achieve better learning results. Redesigning the user interface elements that training observation indicated to be difficult to learn would produce long-term improvements for learnability. In addition to redesigning user interface details, the task sequences that users considered difficult to learn should be redesigned. If the reason for being difficult is not known, the task sequence should be studied in more detail with additional learnability tests with users.

## 4.6 Training Material Analysis

### Method

Training material was reviewed before the training and material usage was observed during the training session. The purpose of the pre-training review was to check all the available material and to evaluate its appropriateness by comparing it against the latest research results on training material and training methods (see section 2.4). The material usage was observed in the training because this was expected to provide information on whether the training material is suitable for the learning needs of new Tekla Structures users.

The following figure illustrates the position of the training material analysis among the research activities.

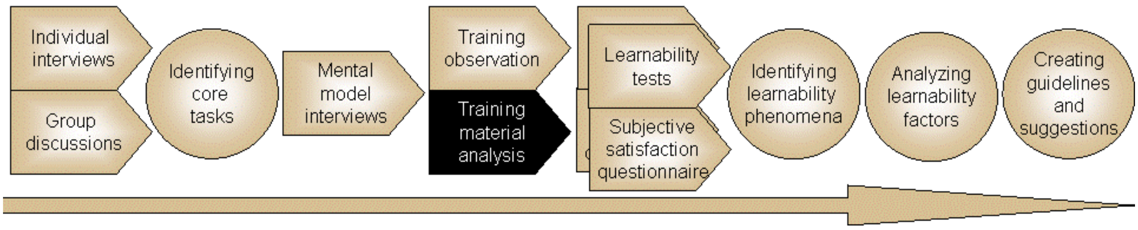



**Figure 15.** Progress of the research activities: training material analysis.

### Results

In the training material analysis, 11 learnability phenomena were found. This is a rather small number compared to the number of phenomena found with other methods. On the other hand, phenomena found in the training material analysis were rather broad and not just observations on the details of the material.

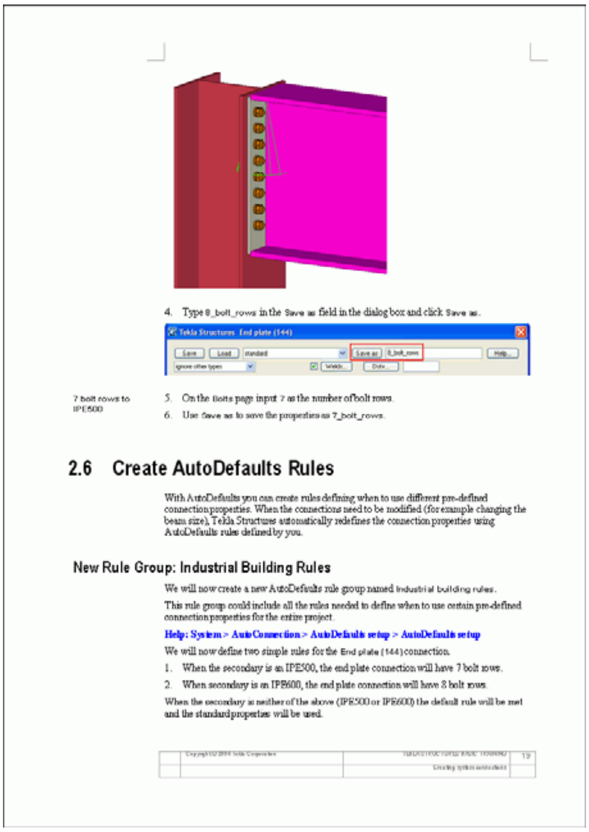
In the following table, material types that are available for Tekla Structures are summarized. The material types were described in section 2.4.

**Table 7.** Availability of material types.

| Material type | Availability |
|---|---|
| **Printed documentation** | No |
| **Electronic documentation** | Yes |
| **Printed tutorial** | Yes |
| **Electronic tutorial** | Yes |
| **Context-sensitive help** | Partly |

There was printed documentation available earlier, but it has been replaced by electronic documentation that is also easier to maintain and update. The tutorial is available in both printed and electronic form. There are also some elements of context-sensitive help available: when the user clicks the F1 button, information on the active dialog is displayed in the help window. However, for other user interface elements than certain dialogs, there is no context-sensitive help available.

In the training, the printed tutorial is used as the primary material. The following figure shows one page of the printed tutorial. Each participant has a tutorial folder on his desk and goes through the exercises it contains. Participants get the same tutorial in PDF format on a CD-ROM and it is available on the hard disk of computers in the training classroom. Thus, it is also possible to use the electronic version of tutorial in the training. However, all training participants were observed to use primarily the printed version of the tutorial.


4. Type 8_bolt_rows in the Save as field in the dialog box and click Save as.

7 bolt rows to IPE500

5. On the Bolts page input 7 as the number of bolt rows.
6. Use Save as to save the properties as 7_bolt_rows.

2.6 Create AutoDefaults Rules

With AutoDefaults you can create rules defining when to use different pre-defined connection properties. When the connections need to be modified (for example changing the beam size), Tekla Structures automatically redefines the connection properties using AutoDefaults rules defined by you.

New Rule Group: Industrial Building Rules

We will now create a new AutoDefaults rule group named Industrial building rules.

This rule group could include all the rules needed to define when to use certain pre-defined connection properties for the entire project.

Help: System > AutoConnection > AutoDefaults setup > AutoDefaults setup

We will now define two simple rules for the End plate (144) connection.

1. When the secondary is an IPE500, the end plate connection will have 7 bolt rows.
2. When secondary is an IPE600, the end plate connection will have 8 bolt rows.

When the secondary is neither of the above (IPE500 or IPE600) the default rule will be met and the standard properties will be used.

19



**Figure 16.** Sample page of the printed tutorial.

The printed tutorial contains 452 pages. 363 pages of it form the concrete design training package that was used in the observed training. The instructor may also customize the training contents according to user needs. Some instructors, for example, skip some of the first chapters to reserve enough time for presenting the advanced features. In the observed training, two chapters of the concrete training material were skipped (29 + 34 pages) and 300 pages of material were covered.

The tutorial also contains 130 references to electronic documentation. The references are presented as a path that indicates the location of the referred subject in the documentation hierarchy. If the user is reading the electronic version of documentation, he may click on the reference link and be led directly to the right page in the documentation. The links to electronic documentation are mainly intended for self-study. In the training, none of the trainees checked the links while doing the exercises. It is not even feasible to assume that learners would familiarize themselves with all the 130 links during the three-day training.

The tutorial is organized around building a real model. The main building block of the tutorial is step-by-step instructions that contain screenshots and textual instructions. The instructions gradually guide the user to construct the model. The tutorial also contains some conceptual explanations in the beginning of lessons and tips for improving performance and avoiding certain undesirable states. There is no table of contents or index in the tutorial.

### Discussion

The training material analysis revealed several learnability phenomena that may support or hinder the learning process. It should be noted that also other issues than ones that had been found by observing users were classified as learnability phenomena in this study. For example, it was noticed that the amount of material is very large. In addition, the amount of material that is available for each of the core tasks differs a lot, and there may be a need to balance the amount of material. The organization of material and the type of instructions that are given may need to be reconsidered. These issues will be discussed in more detail in sections 5.4 and 6.4. Issues found in the training material assessment contribute to the formation of learnability factors related to training.

## 4.7 Scenario-Based Learnability Test

### Method

Scenario-based learnability tests were organized right after the training and two months later. The purpose was to assess the outcome of the training and the self-learning phase after it. Mainly qualitative information on the issues affecting learnability was of interest, but also performance measures were included.

Elliott et al. (2002) and Roberts and Moran (1997), for example, used similar observational methods for evaluating learnability. Naturally, the selection of tasks was different as the system is question was different as well. Corresponding methods have been used by numerous other researchers for evaluating usability. However, based on the literature review, we concluded that those methods are suitable for evaluating learnability as long as the subjects are novice users. Elements of learnability evaluation methods that were presented in sections 2.1 – 2.2 such as collecting qualitative information and measuring time and errors are present in the usability test method. Jacob Nielsen (1993, p. 165-206) has written profound instructions for running scenario-based usability tests, and these instructions have been used to guide our test sessions too.

The following figure indicates our position in the chain of research activities.

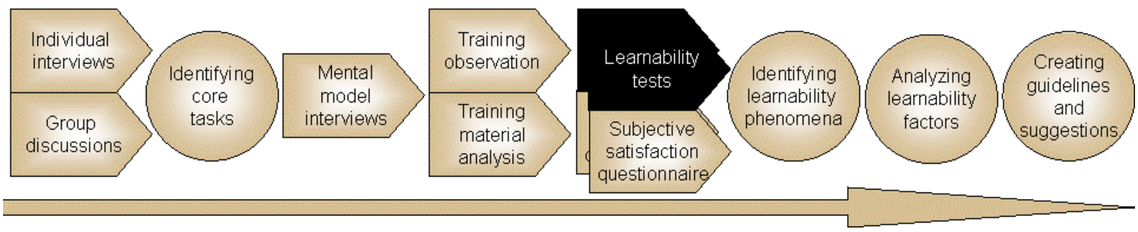



**Figure 17.** Progress of the research activities: scenario-based learnability test.

The test tasks were designed to contain all the core tasks that were presented in section 4.3. The aim was to create tasks that resemble a real work situation as well as possible but are simple enough to enable performing them in one hour. The relationship of the 19 test tasks and the core tasks are mapped in appendix G. The tasks led to constructing a simple model with the Tekla Structures system. The following drawing was given to the user in the beginning of the test session and it shows the model that had to be constructed.

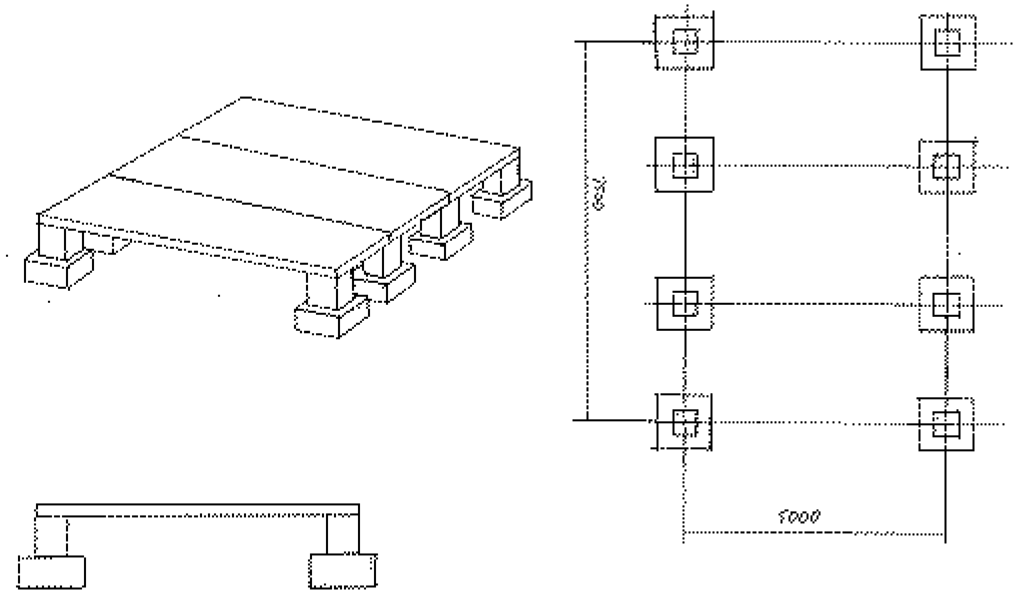



**Figure 18.** The model that users had to construct.

Scenarios describing the test tasks were presented to the user gradually during the modeling process. The scenarios are presented in appendix H. At first, the user was asked to create concrete parts for the foundation. Secondly, he was asked to reinforce one of the parts, and as the user proceeded, he was asked to create drawings, reports, and exported files that are produced as an outcome of a real modeling process as well.

The learnability tests immediately after the training and two months later contained the same tasks with slightly different parameters such as dimensions of the building. In the first learnability test, the user was told that the model represents the foundations of a garage, and in the second learnability test, the model was told to represent the foundations of a storage hall.

Each learnability test was about one hour long. The test was organized in each user's office, in a meeting room or at the user's desk. The user did a test task on a computer and the researcher observed his behavior and took notes on the steps he performed, errors he made, time that was taken for performing the tasks, and his comments. The observation form is presented in appendix I. Test sessions were also recorded on tape. During the learnability test, the user was asked to explain his operations and expectations. This is generally known as the think aloud protocol (see Dix et al., 1993, p. 385-386). The problems that the user faced were collected and the information was used for determining the learnability factors that will be presented later.

The purpose was also to compare the results of the learnability tests to the results of the training observation and training material analysis. Especially the comparison of learnability problems related to each core tasks and the thoroughness of processing the task was of interest.

## Results

137 learnability phenomena were identified in the learnability tests. This is slightly more than in training observations, and much more than in mental model interviews or training material analysis.

The following figure shows an example of a model constructed by one of the users in the first learnability test.

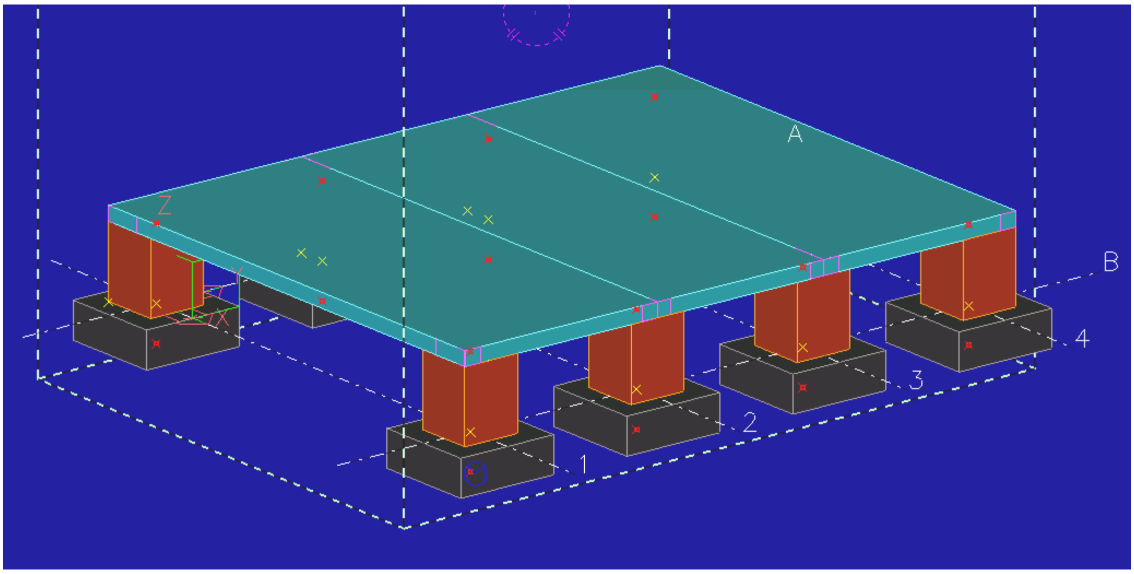


**Figure 19.** The model constructed by U4 in the first learnability test.

Even if the tasks had been simplified as much as possible, not all users could perform all the tasks in one hour, but a some of them had to be left our during the test. In the first learnability test, users completed from 9 to 15 tasks each, and in the second learnability test, they performed from 10 to 16 tasks. Users could complete 12 tasks in both learnability tests on average.

The administrator of the test kept track on the tasks that different users had completed and selected suitable tasks for every user so that all the tasks were completed by a sufficient number of users. Some tasks were completed by all users because they are so central in the modeling process and some are even obligatory before the user can proceed to any other task. Two of the tasks had been classified as advanced tasks for fast users, but they were not used in the test.

All users faced several problems during the learnability test. Sometimes they could solve a problem by experimenting with different operations, but sometimes they needed to ask help from the instructor. There were certain problems that were faced by a remarkable portion of users, sometimes even many times during one learnability test. In the following table, the problems that five or six users in total faced in either of the two learnability tests are presented. More problems are presented in appendix J.

**Table 8.** Examples of problems observed in the learnability tests.

| Core task | Examples of problems | Users experiencing the problem | |
|---|---|---|---|
| | | First learnability test | Second learnability test |
| **Creating grids** | Users were not able to enter grid dimensions to the fields on the Grid properties dialog correctly. When users wanted to create three grid lines with the spacing of 5000, they entered "0 3*5000" to the grid properties dialog, which actually produced four grid lines. | U2, U3, U5, U6 | U3, U4, U5 |
| **Creating concrete or steel parts** | Users did not know how to define the snap settings that they needed. They needed to snap to all points or only grid lines but did not know how to do it. | U1, U2, U3, U5 | U4 |
| **Modifying concrete or steel parts** | Users sometimes selected several parts, double-clicked one of them, and thought that changes they made in the dialog would affect all the parts that they had selected in the beginning. However, if several parts are selected and after that, one of them is double-clicked, the selection is applied only to the part that was double-clicked. | U1, U2 | U3, U4, U5 |
| **Creating reinforcements** | Users had problems finding suitable reinforcements and connections. Users were not familiar with the names of the reinforcements and connections and therefore it was difficult to select a reinforcement from the list of names. Users sometimes entered the search term reinforcement, but as not all reinforcements contain the word in their name, a suitable reinforcement was not found. | U1, U2, U3, U4, U5, U6 | U2, U4 |
| **Saving components** | Users could not choose a correct type for custom components (part, detail, connection, or seam). | U1, U5, U6 | U1, U2, U4 |
| **Creating numbering** | Users often forgot to run numbering before creating drawings. The warning message was shown. | U2, U3, U4, U5 | U1, U2, U3 |
| **Exporting / importing data to other applications** | When users wanted to export the model to AutoCAD, they often chose the option Export > CAD drawing as the name suggests that it will create drawings that are suitable for that. However, they would probably need to create DXF drawings in most situations, which is a separate menu item. | U2, U4, U5 | U1, U3, U4 |

In addition to collecting a list of learnability problems, several other learnability phenomena such as observations concerning things that support learnability as well as users' suggestions for improving the system or training were collected.

In addition to collecting qualitative information on learnability phenomena, user performance was measured with the criteria suggested by Capobianco (2003). Capobianco's criteria conform very well to our operationalization of learnability (see section 2.1). In both learnability tests, average values for these three variables were calculated for each task:

- the percentage of users that could carry out a task without asking help from the test instructor,
- the percentage of users that could carry out a task optimally, which means that the requested end result was achieved without asking help from the instructor, looking at the help pages, undoing and redoing commands, extensive mental effort, or resorting to trial-and-error strategy, and
- average execution time.

The percentage of users that could perform each task optimally or without help and the average times for performing each task are presented in appendix K. A graph of the results is included in appendix L. The performance of users varied considerably between different tasks and from the first learnability test to the second one. The task of saving a reinforcement for later use, for example, could not be done optimally or without instructions by any user in either of the learnability tests. The task of modifying material catalogs could be performed optimally by 33% of users in the first learnability test, but by none in the second learnability test, and all users needed instructions. In addition, all users needed help with creating reinforcements and modifying the model. Users performed rather well with creating grids, modifying the pad footings, and modifying concrete slab properties, with some exceptions.

In the second learnability test session arranged two months after the training, there was a clear difference in the performance of users that had used the system between the two learnability tests and users that had not used it. The performance of these groups in the second learnability test is presented in the following table. The number of users in these two groups is very small, so the comparison does not have statistical significance.

In the following two tables, some central figures from the learnability tests that were done right after the training and two months later are presented. The results are marked with color-coding. The meaning of the colors is presented below.

| |
|---|
| **Light gray:** percentage of users 70 – 100% or average time 0 – 3 min |
| **Dark grey:** percentage of users 30 – 70% or average time 3 – 7 min |
| **Black:** percentage of users 0 – 30% or average time 7 – 10 min |

**Table 9.** Performance of users in learnability tests.

| | **Percentage of tasks performed without instructions** | **Percentage of tasks performed without instructions** | **Percentage of tasks performed without instructions** |
|---|---|---|---|
| **Learnability test immediately after the training** | | | |
| Average for all users | 43 % | 12 % | 5:03 |
| **Learnability test two months later** | | | |
| Average for all users | 63 % | 26 % | 4:58 |
| Average for users that had used the system (2 users) | 71 % | 46 % | 4:03 |
| Average for users that had not used the system (3 users) | 58 % | 6 % | 5:39 |

## Discussion

The test was successful in that it pointed out a remarkable number of learnability phenomena such as learnability problems, things supporting learnability, and suggestions and comments from users. By addressing these issues, learnability of the system can be improved.

The weakness of this experimental design is that the observed learnability phenomena are only related to the test tasks even though the system contains numerous functions that were not touched by the test tasks.

The performance of users demonstrates the fact that Tekla Structures is a complex system and mastery over it cannot be achieved during a three-day training course. In the first learnability test, users could be performed on average 43% of tasks without without instructions, but only 12% of tasks optimally. In the second learnability test, the figures were 63% and 27%. The figures show that the learnability of the system as well as training courses and training material need to be improved.

Results of the learnability tests were compared with the observations made during the training. Connections were found between the learnability problems noted in the learnability tests and the time spent with each core task in training and concepts that were or were not explained by the instructor. It was also noted that even in the first learnability test arranged immediately after training, users did not remember even close to everything that was taught in training. They may have processed the information on a surface level that does not lead to proper memorization. Connections were also found between the learnability problems and

The results of the learnability tests were also compared with the results of the training material assessment. Connections were found between the amount of material that was available and the learnability problems related to each core task. There were some tasks with only five to seven pages of training material available, and furthermore, some of this material may have been skipped in training. On the other hand, for certain tasks, there were even 60 pages of material available, which also led to spending a long time with these tasks in training.

## 4.8 Subjective Satisfaction Questionnaire

### Method

In the end of the learnability test sessions immediately after the training and two months later, users were asked to fill in a 2-page learnability questionnaire. The purpose was to collect subjective opinions of users on issues that affect learnability. The need for assessing subjective opinions can also be seen from our definition of learnability that contains the word *pleasantly*.

The use of a questionnaire for measuring the subjective satisfaction dimension of learnability has been suggested in the ISO 9241-11 standard (SFS-EN, 1998). The ISO standard includes references to some well-known satisfaction questionnaires such as QUIS (Questionnaire for User Interface Satisfaction) developed by Chin, Diehl, and Norman (1988) and the SUMI (Software Usability Measurement Inventory) questionnaire developed by J. Kirakowski (1996). The QUIS questionnaire includes some questions related to learning, and the SUMI questionnaire contains a set of questions related to learnability. Some of these questions were modified and incorporated into the questionnaire form used in our study. Questions from one of the most famous questionnaires that is known as the IBM Computer Usability Satisfaction Questionnaire (see Lewis, 1995) were also adapted to this study. The well-known questionnaires have been checked for validity and reliability, and therefore it is recommendable to use them as a basis for our questionnaire.

The following figure shows the position of this research method among other research activities.

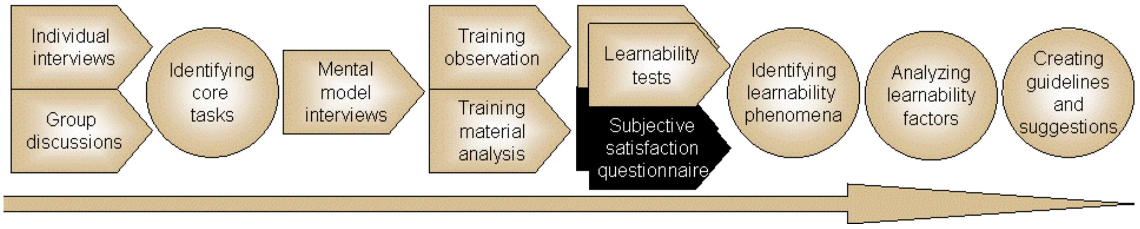



**Figure 20.** Progress of the research activities: subjective satisfaction questionnaire.

The questionnaire used in this study was divided into four sections: general questions, learnability of the user interface, material and training, and function specific questions. The first two groups of questions gave a picture of how satisfied users are with the learning process. Questions on material and training provided information on which support methods users consider being the most important. The purpose of the function specific questions was to see which features users considered most difficult to learn. The questionnaire contained 30 questions altogether. A five-point Likert scale (see Lewis, 1995) with an adjective in both ends was used. For function specific questions, the alternative "I cannot do it" was added because it is not feasible for the user to estimate the difficulty of a function with which he is not familiar. The questionnaire form is presented in appendix M.

Questionnaire answers were scored from 1 to 5 so that an average grade could be calculated for each question. The lowest score corresponds to negative adjectives such as "difficult" or "useless" and the highest score corresponds to positive adjectives such as "easy" or "useful". The answers "I cannot do it" and "I have not used" were given the score of 0.

The aim was also to compare the results of the questionnaire to the results of the learnability tests. It was of interest whether the subjective assessment of the difficulty of a task correlated with the performance of users with the corresponding task in the learnability test.

## Results

The average grades calculated from all answers are presented in appendix N. The appendix includes both the grades that users gave immediately after the course. In this section, we introduce some important learnability phenomena that the questionnaire results pointed out. 18 phenomena altogether were extracted from the subjective satisfaction questionnaire results.

Average scores ranged from 1 to 4.7, which means that there are clear distinctions between different items. The scores given in the two phases (immediately after the training and two months later) differed from each other slightly but not radically.

Despite learning difficulties, users gave an excellent score (4.2 in both phases) to the item asking if the system corresponds to their expectations. It would require further investigation what were the most important expectations of users that determined the score that they gave.

Questions related to the learnability of the user interface got scores from 2.8 to 4.0. Remembering names and use of commands got relatively low scores (2.8 immediately after the training and 3.0 two months later).

Questions concerning material and training got scores ranging from 2.8 to 4.7. In the first phase, training and training material got very high scores (4.7 and 4.5). Instructions on the computer screen got the score of 4.7; this is surprising, as currently there are not much instructions available in the interface. In the second phase, the scores for the items mentioned above were considerably lower. This may be due to the fact that users had not used the materials very much after the training. Training material CD and context-sensitive help received relatively low scores (2.8 and 1.2, and 3.0 and 1.0). There were rather many users that answered they had not used some of the material types. In the training, users may find it easier to ask help from instructor as compared to reading help pages, and after the training, they may ask help from colleagues.

Function-specific questions received scores ranging from 1.0 to 4.7. Exporting and importing data, specifying model properties, and modifying material and profile catalogs received very little attention in the training, which may be the reason for the low scores. On the contrary, creating grids and creating concrete and steel parts received an excellent score in the questionnaire. It was also noted in user learnability tests that users could perform these basic operations rather well.

### Discussion

The fact that users considered remembering names and use of commands difficult indicates that the system requires too much memorization from a novice user. The reason for this may be for example that terminology differs from what users are accustomed with and that the software language is English that is not the native language of users. Based on the questionnaire, special attention should be paid to the amount of information that the user is required to remember. Making all the information visible in the user interface would reduce users' memory load.

The fact that users were very content with the current training and printed training material is good from the perspective from user satisfaction, but it does not indicate whether the teaching methods lead to best possible learning results.

The user interface or operating logics of the functions that were rated as difficult to learn should be redesigned. In addition, special attention should be paid to these issues in training.

It is possible, of course, that users have not understood the terminology of the questions correctly. Especially the term context-sensitive help may be unfamiliar to users, which was the reason for including the hint "opens with F1 button" in parenthesis. This reduces the reliability of the results.

When the results of the function-specific questions in the subjective satisfaction questionnaire were compared with the results of the learnability test, they were noticed to be rather well in line. For example, creating grids and concrete and steel parts received the highest score in the questionnaire and these tasks could be performed rather well in the learnability test too. Exporting and importing data as well as specifying model properties was rated difficult and could not be performed very well in the learnability test either. Other parts of the questionnaire also revealed some of the users' opinions that could not be known just by observing users. However, as the definition of learnability presented in section 2.1 contains the word 'pleasantly', it is important to let the users to evaluate the learnability themselves.

The learnability phenomena collected from the questionnaire results will be utilized for finding the learnability factors related to the user interface, system structure, and training.

# 5 Classification of Factors Influencing Learnability

## *5.1 From Learnability Phenomena to Learnability Factors*

After completing all the empirical research, there was a large amount of data available. To present the data found with different research methods in a consistent form, all learnability phenomena that had been found with different research methods were collected into a large table. The research activity that produced each phenomenon was mentioned in the table. If applicable, the number of users that the phenomenon applied to as well as the related core task were also mentioned. The table contained 237 rows altogether, each containing one learnability phenomenon. Because of the large number of phenomenon, they are not presented here, but sample rows of the table containing the learnability phenomena are presented in appendix O.

The following figure shows our position in the chain of research activities: we have completed all the empirical research activities and have a set of learnability phenomena to be analyzed further. We simplified the figure by presenting "identifying learnability phenomena" as one phase after all the research activities, but actually, the phenomena were identified from the data collected with each method right after completing the activities related to that method.

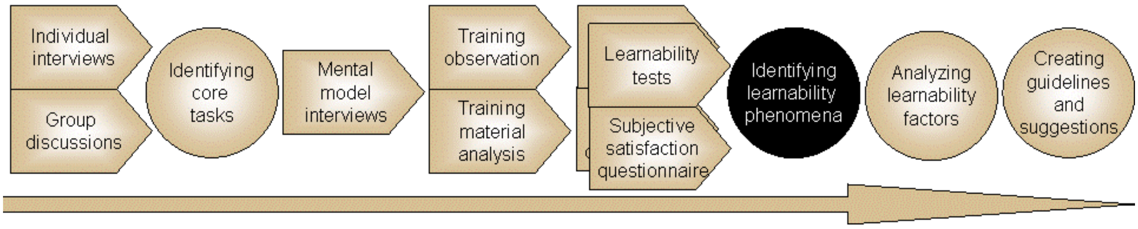



**Figure 21.** Progress of the research activities: learnability phenomena.

Next, the learnability phenomena were grouped in order to find a set of learnability factors that cover all the observations. We used a variation of the grounded theory method to group the phenomena. The grounded theory method starts with an unorganized set of data and without no predefined theoretical framework and proceeds by identifying themes and patterns from the data. As the analysis proceeds, more evidential data for the themes and patterns is searched for. Elliott et al. (2002), for example, used the grounded theory method for deriving the learnability factors from the observational data that was collected in user observations and focus group discussions. According to them, the grounded theory method is useful in that it can produce a theory that fits the available set of data. It can be successfully used for analyzing qualitative but also quantitative data.

Three sets of learnability factors were produced from the learnability phenomena we had collected: one set for learnability factors concerning the user interface, one for factors concerning the differences between the user and system models, and one for factors concerning training. One observation could contribute to one or more of the three sets of learnability factors. It was sometimes considered feasible to connect one observation to both user interface and training, for example, because a certain learnability problem may really be connected to both difficult user interface and insufficient training and these two can by no means be combined into one issue.

The following figure shows that we are now in the one but last activity in the research process.

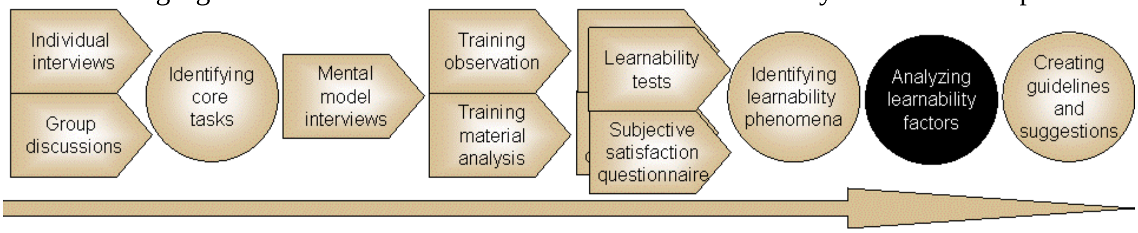



**Figure 22.** Progress of the research activities: learnability factors.

Finding a reasonable grouping was a time-consuming task with several iterations. As a result of the iterations, a set of 15 learnability factors was finally produced. Seven of them were related to the user interface, four of them to the system structure, and seven to training. Examples of phenomena related to each of the learnability factors are presented in appendix P, together with the suggestions for improving learnability. The following diagram gives an overview of the learnability factors.

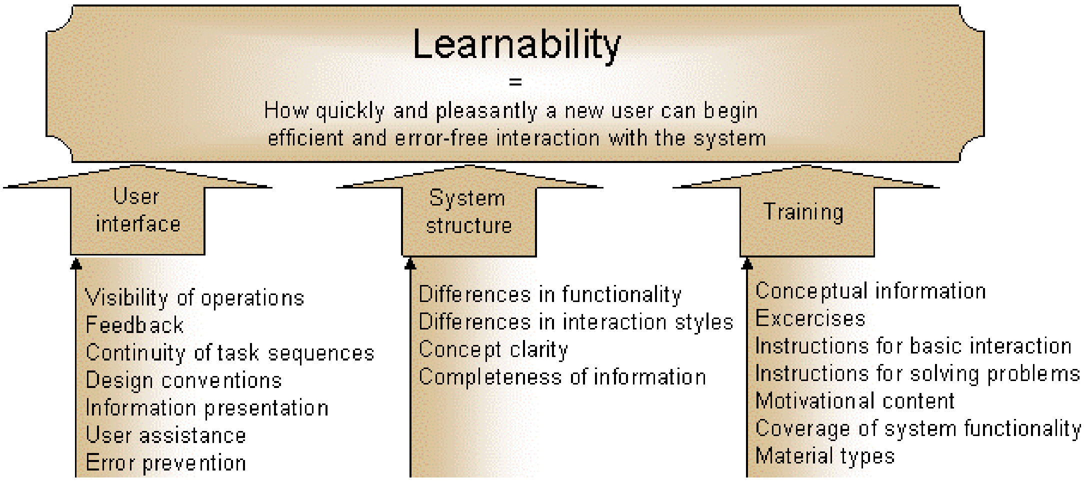



**Figure 23.** Learnability factors.

## *5.2 User Interface*

### Overview of Learnability Factors Related to the User Interface

User interface is crucial in determining how easy it is to learn to use the system. Numerous learnability problems and issues supporting learnability were noted during the research, and many of them are directly related to the design of certain user interface elements.

Some of the seven learnability factors that are related to the user interface are familiar from usability checklists (see e.g. Nielsen, 1993, p. 20). However, the factors differ in that they concentrate on the issues that are important for a new user. Visibility of operations, feedback, and continuity of task sequences are essential for enabling the user to perform operations that are new to him. Design conventions and information presentation have an effect on how the user will recognize usage principles and understand the functionality of the system. User assistance will aid the user in learning to use the system, and error prevention will reduce the number of problems the user will face.

The number of learnability phenomena that were related to each of the factors is presented below. The factors will be presented in more detail in the following sections, with some descriptive examples of user comments.

**Table 10.** Learnability factors and phenomena related to the user interface

| Factor influencing learnability | Number of related learnability phenomena |
|---|---|
| **Visibility of operations** | 68 |
| **Feedback** | 23 |
| **Continuity of task sequences** | 16 |
| **Design conventions** | 14 |
| **Information presentation** | 45 |
| **User assistance** | 10 |
| **Error prevention** | 6 |

## Visibility of Operations

An essential requirement for a learnable user interface is the visibility of possible operations and the type and syntax of the required input for performing the operation. Whereas an expert user can rely on his memorization, a novice user must deduce the possible operations and inputs from the hints given by the user interface. It was noted in the observations that users had problems finding commands that were not clearly visible near the object that they were interacting with. In addition, users did not necessarily remember the name of the command that was needed to reach a desired end result, even if they had used it in the training. If they remembered the name, they may not have remembered where in the menus or toolbars the command was located. To aid the user with this problem, the most central commands should be visible or easily found and the user should be directed towards performing the right operation. The user interface should provide all the necessary information so that the user can fill in the required fields on a dialog, select appropriate objects with mouse, or otherwise enter the required input.

Examples of comments that reflect the lack of visibility of operations and inputs are presented below. The comments were presented by users during the training observation and in the pre-training and post-training learnability tests. They are related to finding commands from menus, noticing methods to access some items, distinguishing the operations that can be performed in a certain state, and knowing the way to input information.

The research activity during which a comment was made and the user that voiced the comment is marked on the left-hand side of each comment. Abbreviations are used to shorten the presentation. The meanings of the abbreviations are as follows:

- MMI = mental model interview
- TO = training observation
- TMA = training material analysis
- LTa = scenario-based learnability test immediately after the training
- LTb = scenario-based learnability test two months later
- SSQ = subjective satisfaction questionnaire
- U1, U2, U3, U4, U5, U6 = users

| | |
|---|---|
| TO U1 | "Was it under View?" |
| LTa U4 | "So now I should add something there. How would it succeed… I don't have any idea how it would be done." |
| TO U6 | "How can I open that dialog?" |
| MMI U5 | "It is difficult to say how I would get... to edit that text." |
| TO U5 | "How can you go to the menu? … I don't have it there!" |
| LTa U1 | "Oh, it was that one… I had to go to the drawing state." |
| LTa U1 | "What should I enter there… Would it go like that?" |

## Feedback

Feedback is also important for experienced users but especially for novices. It was observed that novices were often unsure about whether they succeeded with a certain operation and therefore they would have appreciated a confirmative feedback message. Also, novice users would have sometimes needed feedback about the system state or former actions; for example, if a user had earlier selected certain settings and tried to perform an operation that contradicted with the settings that were selected earlier, he should have been reminded about the former operation and the contradiction between the operations.

Some examples of user comments in situations in which proper feedback was missing but would have been advantageous are presented below. Comments are related to understanding the object visibility and knowing if an operation had succeeded.

| | |
|---|---|
| TO U1 | "Why it selects only the columns?" |
| TO U1 | "Why I don't see some of the beams in these views?" |
| MMI U5 | "Did it do something?" |
| LTa U4 | "See… did it succeed?" |

## Continuity of Task Sequences

Discontinuities in the task sequence were noticed to be problematic for novice users because they often did not remember or recognize how they should proceed and failed to complete the task. The most desirable situation would be that when a user starts a command from a menu or by clicking on an icon, he would be directed until the end of the task sequence by providing a sequence of dialog boxes or instructions in the status bar. He should not be required to jump from one menu to another while performing one task.

Examples of user comments in situations where task sequences were not continuous are presented below. They are related to assigning numbers and defining drawing classifier settings.

| | |
|---|---|
| LTa U2 | "There is numbering in two places? This is confusing. Here it does it." |
| LTb U3 | "Ok. It was wrong numbering." |
| MMI U1 | "Now it does not create drawings because I have not done the numbering but..." |
| LTb U3 | "I'll try assembly drawing. Now it asks about the numbering. I think I need to renumber it now." |
| TO U3 | "If it does not do what you want it to do, it is difficult! And all these windows… It depends on so many things." |
| LTa U4 | "What was the name of the command? I had to filter something. That was rather difficult." |

## Design Conventions

Several parts of the user interface of Tekla Structures are designed differently than commonly in desktop computing. It was observed that users were wondering about the differences. If the design conventions set by user interface standards and the most common office, web, or CAD software were followed, users could easily grasp the meaning and usage of elements that they had seen in other applications as well.

Examples of comments that indicate that some user interface elements do not follow design conventions and were considered problematic by novice users are presented below. Especially the basic control buttons were unconventional and caused difficulties.

| | |
|---|---|
| MMI U2 | "Usually it is so that with OK, you accept the changes. This may feel a bit strange in the beginning." |
| LTa U1 | "Oh, I should have clicked Modify. This is a bit strange that it only does the changes with Modify. If you are not used to it…" |
| LTb U4 | "No, it didn't change. I had to click Modify. |
| MMI U1 | "I expect that a model is started with some kind of template? And there are many templates available, so that suitable templates can be used for different projects?" |
| TO U5 | " I already closed it with Cancel!" |

## Information Presentation

Novice users would have needed especially detailed descriptions for components on dialog boxes, fields that require input, or image details. Novice users did not always understand for example a graphical presentation that may be self-evident for an expert. Therefore, additional information needs to be available. The amount of information that seems excessive to an expert user may be necessary for a novice user. Special attention should be given to the clarity of text and images as well.

Examples of comments that indicate information was not presented clearly enough are presented below. Users had difficulties e.g. with interpreting the fields and images on dialogs and finding a suitable item from a list.

| | |
|---|---|
| LTa U5 | "Oh… I thought they would be straight. But it made 90 also there. But it doesn't matter. It is fine like that. The reinforcer will have more work." |
| LTb U1 | "I really don't know which one of these fields changes the distance of the bar from the column face. Or is it even possible to change it." |
| LTa U1 | "There is some X there. What does it mean then?" |
| LTa U6 | "Cast unit list... I don't know how to find that list there… Cast unit rebar list... I think the masters have to stay there wondering… I cannot find it!" |

## User Assistance

In many problematic situations that were observed in the training and during the learnability tests, properly designed user assistance might have helped the user to overcome the problem. Therefore, user assistance is also included in this list of learnability attributes. Current technologies also allow for the user assistance to become more a part of the user interface rather than a separate help system.

Examples of user comments in situations in which user assistance was missing or incomplete but would have been advantageous are presented below. Some of the comments reflect the fact that some of the instructions currently included in the user interface are unclear.

| | |
|---|---|
| LTa U3 | "Maybe they can not be seen as they are not there. I might know why. … I don't know why it is not shown in 3D. It's a bit strange. … It might be related to this: I set grid this… I don't know. It looks like being ok." |
| TO U6 | "How should I pick the points to create the reinforcing bar group?" |
| LTa U5 | "Select. There they are. 'In the model, select one or two positions'. Oh. One or two… What does it mean?" |
| | "Pick main part. Pick position. Pick main part. Pick position. Pick main part… Haha!" |
| SSQ | In the questionnaire filled in immediately after the training, one user marked that he had not used the help pages, and two months later, two users marked they had not used them. |
| | In the questionnaire filled in immediately after the training, two users marked that they had not used the context-sensitive help, and two months later, four users marked they had not used them. |
| | This indicates that a separate help system is not used very often. |
| LTb U2 | "Pick points… What are these?" |

### Error Prevention

Several errors that were observed in this study were faced by many or even all of the six users. It can be expected that a remarkable percentage of users will face these errors when learning to use the system. Several common causes of errors are also known by members of the development organization. Many of these errors could be prevented by doing a small change in the user interface.

Examples of comments that indicate errors that could have been prevented rather easily are presented below. The comments were presented in a situation in which the user tried to perform a basic operation such as create views, define a custom component, or run clash check, but did not succeed because of some simple mistake.

| | |
|---|---|
| TO U5 | "No selected grid found… I needed to select the grid first." |
| LTa U1 | "This was a bit… would it be… part? I cannot remember at all." |
| LTb U2 | "Connection… detail… what is this? Part?" |
| TO U5 | "It says that No collision detected… Do I need to select them?" |

## *5.3 System Structure*

### Overview of Learnability Factors Related to the System Structure

The learnability factors in this group are connected to the system design on a deeper level than the factors related to the user interface. Often in product development projects, the system structure, which contains the scope of the system, underlying concepts, and available functionality, is decided first and only after that, the user interface is designed. Learnability issues should be taken into account when planning the system structure, because that sets the foundation for an easily learned system. However, the learnability factors related to the system structure and the user interface are in many cases connected.

The evidential data for these learnability factors arose mostly from situations in which the user's mental model differed from the actual system structure. It was noted that users often compared the new system to the software applications they had used before. Users had formed their mental model on the basis of the system they were familiar with, and as the new system was different from the familiar one, users faced problems with using it. The connection between differences in users' mental models and system structure has been noted earlier (see e.g. Kellogg & Breen, 1988) and was confirmed in this study.

Two of the learnability factors are related to the differences between the system that users are familiar with and the new system. The first of these is differences in functionality and the second is differences in interaction styles. Both cause discrepancies between the mental model of the user and the system structure, as the mental model contains some items from the system that the user has used earlier but that is designed differently. The third factor, concept clarity, refers to the new concepts that are necessary to understand in order to learn to use the new system. If the concepts are named descriptively, it is relatively easy for the user to understand them and adopt them as a part of his mental model. Information presentation, which is the fourth factor, is crucial in determining how the user will interpret the new concepts, the user interface elements, and the underlying system structure. A correct interpretation will lead to a correct mental model, whereas an incorrect interpretation will lead to an incorrect model.

The number of learnability phenomena that are related to each of the learnability factors is presented in the table below. The factors will be discussed in more detail in the following sections.

**Table 11.** Learnability factors and phenomena related to the system structure

| Factor influencing learnability | Number of related learnability phenomena |
|---|---|
| **Differences in functionality** | 9 |
| **Differences in interaction styles** | 16 |
| **Concept clarity** | 30 |
| **Completeness of information** | 60 |

## Differences in Functionality

The functionality of different software applications naturally always varies. Even if two applications can be used for the same purpose, they will provide different tools for reaching the end result. Differences in functionality was observed to cause problems for learners. As was said in section 2.2 and observed in the mental model interview, users base their expectations for a new software application on their experiences with familiar applications.

In the case of Tekla Structures, most users have previously worked with two-dimensional drawing software and are in the process of moving to three-dimensional modeling. This caused many difficulties, as users had based their mental models on the drawing paradigm and now they should have switched to the modeling paradigm that changes many aspects of their work

Examples of comments reflecting the differences between mental models of users and system structure are presented below. Users had problems e.g. with understanding the difference between the concepts of drawing and modeling or drawing and view. The concept of numbering was also unclear to them.

| | |
|---|---|
| MMI U2 | "That is at least my understanding that the parts will be modeled. …And now the model and the drawing are the same thing." |
| MMI U3 | "I think this is such that you can make a whole object at a time and you can then modify it, whereas in AutoCAD you make one line at a time." |
| MMI U3 | "Is it (the drawing list) the same as this (the view list)?" |
| TO U1 | "Numbering is still unclear to me." |
| LTa U4 | "It remained unclear to me, what is the sense with numbering? We were not shown what is the advantage of numbering?" |
| | "Why it does not create it automatically? If I create an object that looks the same, it could recognize somehow which one it is. I mean that if I draw two of these, it would number them automatically, so that I would not need to do it. As it is done almost every time anyway." |
| | "In AutoCAD, we modify each number separately, for each element." |

## Differences in Interaction Styles

Interaction styles of different software applications naturally also vary. Some of this variation may be necessary because of the different nature of the applications, but some of it is unnecessary and should be avoided.

Just like mental models concerning system functionality, mental models concerning interaction styles seemed to be based on users' experiences with other software applications. As interaction styles are not domain-specific, users based some of their expectations on how the most common office software or operating systems work. They also expected that they could interact with a new software application similarly to the previously used, corresponding application.

Examples of user comments in situations in which users' expectations for interaction styles were different from the actual interaction style are listed below. Especially the methods for moving and resizing parts were different from the methods with which users were familiar.

| | |
|---|---|
| TO U3 | "I have been missing the possibility to grab an element and move it." |
| TO U5 | "How can I move the end of a part?" |
| LTb U1 | "I haven't found a simple stretch command here, at least yet." |
| TO U5 | "Does the point need to be yellow?" |
| | "How do I choose the point? What if there are two parts attached to the part? Is the one that is a square the starting point?" |

## Concept Clarity

When starting to use a new software application, the user usually needs to learn new concepts. This is also true for Tekla Structures: there are several concepts that have not been used in any other commonly known software because the functionality they reflect is new and domain-specific. Learning these concepts would be easiest if the concepts were self-explanatory, which means that the user interface communicated their meaning clearly and the terminology was familiar to users.

Certain concepts were observed to be difficult to grasp for users. Examples of comments that reflect the unclarity of certain concepts and the difficulty to understand terminology in English are presented below.

| | |
|---|---|
| SSQ | Users considered remembering the name and use of commands to be rather difficult (scores 2.8 and 3). |
| TO U1 | "What does this class refer to?" |
| TO U3 | "What is the difference of Save and freeze with the Save command?" |
| LTa U2 | "Part mark? Eh… So… Now I have a blackout again…" |
| LTa U5 | "Export? I was there but there was nothing feasible. I was there but there was just XML DWL…" |
| TO U5 | "Where do I find the hollow core slab profile?" |
| TO U6 | "How do I find the right profile?" |
| TO U4 | "What does the whole thing do?" |
| TO U6 | "Isn't there a Finnish language version of the software?" |

## Completeness of Information

It was noticed in the mental model interviews, training observation, and scenario-based learnability test, that users could not always form a correct mental model of the system because there was not enough information available. The change in the mental model could be facilitated by providing enough information about the user interface elements, concepts that are present in the system, and operations and their causes and effects.

Examples of user comments in situations in which incomplete information caused difficulties for learners are presented below. The first comment is related to the list of views and the other comments to dialogs containing images and fields for object properties.

| | |
|---|---|
| LTa U3 | "Let's take from here… 00 plan… Let's take it away as I don't know what it is." |
| TO U5 | "Can you explain the position settings (On plane, Rotation, At depth)?" |
| TO U1 | "In what direction do the From plane and In plane commands move the reinforcing bar?" |
| LTa U6 | "You must think about these a lot, these are not clear at all, even if they in principle are... What does that mean (the check box)?" |

## *5.4 Training*

### Overview of Learnability Factors Related to Training

In this section, aspects of training that were noticed to affect learnability are presented. The information was extracted from the training observation and comments that users made in the observation sessions after the training. Many of the issues were discussed in section 2.4 and research methods confirmed that the issues have an effect on learnability.

The first four factors, namely conceptual information, exercises, instructions for basic interaction, and instructions for solving problems, are components that are necessary for skill acquisition. The fifth factor, motivational content, was reasoned to be an important factor in determining how well learners will adopt the new information and how actively they will continue to apply the learned skills after the training. Good coverage of system functionality is essential in order to enable the users to use the system effectively after the training. The choice of instructional material is also crucial in determining the learning results.

The number of observed learnability phenomena that are related to each of the factors is presented in the following table. The factors will be presented in more detail in the following sections.

**Table 12.** Learnability factors and phenomena related to training.

| Factor influencing learnability | Number of related learnability phenomena |
|---|---|
| **Conceptual information** | 45 |
| **Exercises** | 44 |
| **Instructions for basic interaction** | 14 |
| **Instructions for solving problems** | 16 |
| **Motivational and orientational content** | 3 |
| **Coverage of essential system functions** | 9 |
| **Material types** | 13 |

### Conceptual Information

It was stated in section 2.4 that conceptual information should be included in training. Conceptual information will help the user to build a mental model of the subject. For skill learning, mere memorization of procedures is not enough but it is desirable to understand the procedure on conceptual level.

Several observations that were made during the training or in after-training meetings indicate that not enough conceptual information was delivered in training. Users did not know all the concepts that are related to the core tasks and therefore they had problems reaching the end result. Examples of comments indicating missing conceptual information are presented below. The comments are related to the drawing classifier and template editor that can be used for modifying drawings, and the freeze command that can be used for keeping the modified items in a drawing.

| | |
|---|---|
| LTa U5 | "Drawing classifier and template editor… what are they? I have no idea." |
| LTa U2 | "Well… This was also… dim to me…" |
| LTa U2 | "Yes… so… we did not go through it but… here was also… It is related to this, this freeze. It of course remained a bit unclear. Locking was clear so that you cannot access it but freeze was… it does something corresponding." |

## Exercises

In section 2.4, it was stated that it is necessary for skill learning to practice operations by doing exercises. A considerable amount of exercises is included in Tekla Structures training which is very good. However, it was noticed that users did not always remember how to perform an operation that they practiced in the training. Sometimes they did not even remember practicing them all. Providing very detailed step-by-step instructions may have resulted in a surface learning result that could not be applied in new situations. To overcome this problem, the nature of the exercises should be reconsidered.

Examples of comments that indicate that either a certain procedure is not practiced enough or the nature of the exercise is not the best possible are presented below.

| | |
|---|---|
| LTa U2 | "Here were these… Hm. How did I forget it that fast?"<br>"This was still… I have forgotten… How to create views there. How to get to the dialog box."<br>"Where was it? We went through it (in the training)." |
| LTa U4 | "We haven't done this before. Or we have… I just don't remember!" |
| LTa U5 | I think we should get to do more ourselves in the course. We should get to do more and then it would become familiar. That is at least how I learn, others may learn differently. |

## Instructions for Basic Interaction

It was observed that users were not familiar with all the basic interaction strategies even after the training. However, teaching those strategies thoroughly in the training would have raised productivity during the post-training learning period. This is because users would not have needed to spend time with simple interaction problems.

In Tekla Structures training material, instructions for basic interaction are included in the step-by-step instructions. This is one possible way to deliver the information but it is necessarily not the most efficient one.

Examples of basic interaction strategies that users were observed not to be familiar with are presented below.

| | |
|---|---|
| TO U2 | "How do I pick a point and enter a numeric location?" |
| TO U5 | "Can you show once more how to select the starting point?" |
| LTa U1 | "It was a bit unclear to me where it actually snaps."<br>"But it does not take it! It does not snap." |
| LTa U3 | "If I give it from the middle point, which I don't want, then I have to change the snap settings. I don't remember how to." |
| LTa U5 | "They are again these, I don't know about these at all, these snaps. What I should grab and when. Object, components, select object…" |

## Instructions for Solving Problems

As was written in section 2.4, users will unavoidably face problems when using a new software application. This was true also in the case of Tekla Structures. However, it was observed that users were not very well prepared for solving problems. In the training, they always asked help from the instructor when facing problems. During the scenario-based learnability test, they either asked help from the observer or often simply gave up. To change this, users should be equipped in training to solve problems. This would help them to use the application independently when no instructor or observer is available for help.

Example of a user comment during the learnability test is presented below. The user asked for help from the observer and did not try to solve the problem by e.g. looking at the help pages.

| | |
|---|---|
| LTa U3 | "If it still says that it is not good, I need some help." |

## Motivational Content

In section 2.4, motivational content was said to be important because it can affect the learning behavior of users both during the training and after the training. It could be heard from user comments that starting to use new software causes major changes for their work. Users' motivation after the training may have been too low, as only two of them had used the software after the training. Naturally, the reason for this may also be something else than the motivational factors, such as commands received from a manager. However, all the motivational elements that are available for the training should be used. Orientational elements in the beginning of the training, for example, would have helped the user to get motivated to learn the subjects and to connect them to the existing information.

A comment that indicates the need for orientational material is presented below. The comment is also related to presenting conceptual information.

| | |
|---|---|
| LTa U1 | "But, yes. I have such critique for the training that you could not really piece together what we were doing. Here are macros and custom components, but what is the classification for them. We looked at them with the binoculars, but I don't really know for example what the symbols mean. It is the basic information that we did not hear at all. There could be some explanation when we take some tool, that this triangle means this, and what are the differences between components and connections, and why there is a certain figure somewhere and another one elsewhere." |

## Coverage of System Functionality

The importance of analyzing user needs was stated in section 2.4. Only after this can the system functions that are essential for the participants be addressed in training. It was noticed in the observations that some central tasks had received only a little attention in the training and users had problems with performing them. Users said that especially exporting and importing data is important in their work and therefore it should have been covered in the training.

Observed issues that are related to the coverage of essential system functions are presented below.

| | |
|---|---|
| LTa U4 | "I should have asked that on the course. But he did not show it at all. Or did he? We have had it many times that a person is on holiday and he has drawn something with Xsteel, and then I had to get a drawing. We tried to make a DXF but we did not succeed. There was nobody near there that knew how to do it." |
| LTa U1 | "Making exceptional geometries is difficult."<br>"We could model some rather difficult thing and made some changes in it." |
| LTb U3 | "I don't remember seeing this (export) dialog at all. Did we use it in training?" |
| TMA | There are less than 10 pages of material available for creating grids, updating drawings, exporting / importing data, and specifying model properties. |
| TMA | There are more than 40 pages of material available for creating concrete or steel parts, creating connections, and creating drawings. |
| TO | Less than 20 minutes were spent with creating views, creating grids, modifying concrete or steel parts, creating numbering, updating drawings, creating reports, exporting / importing data, specifying model properties, and modifying catalogs. |
| TO | More than 2 hours were spent with creating concrete or steel parts, creating reinforcements, and modifying drawings. |
| LTa U3 | "This might have been taught in the training, but there was so much information, I might have gone past." |

## Material Types

The material type that is used in training and provided for additional support should be carefully considered. Several observations concerning the appropriateness of different material types were made in this study. It was noticed that different material types fit different situations; for example, users prefer using printed material in the training, but later, instructions on the computer screen would be advantageous. The quality of the material design also naturally affects users' perception of its appropriateness.

Examples of issues that were observed in this study and that indicate the particularity of user needs in each learning situation are presented in the following table.

| | |
|---|---|
| SSQ | Instructions on the computer screen got a very high score in the first subjective satisfaction questionnaire (score 4.7). |
| SSQ | Printed training material got a very high score in the first subjective satisfaction questionnaire (score 4.5). |
| SSQ | Training material CD got a low score in the subjective satisfaction questionnaires (2.8 and 1.2). |
| SSQ | Context-sensitive help got a low score in the subjective satisfaction questionnaires (3.0 and 1.0). |
| LTb U3 | "Of course, if I would have the manual here, it would be easier. Help is of course also available. I probably need to use it at some point." |
| LTa U5 | "Training material is good, but I think it should be in Finnish." |
| LTa U6 | "The correct number was here in the material somewhere." |
| TMA | In the training, users had both the printed tutorial and an electronic version of it available. As all the screen space was needed for the modeling software, all the users chose to use primarily the printed version of the tutorial. |
| TMA | The training material folder contains 452 pages, of which 363 pages form the concrete training package. |
| TO | 300 pages were covered in the observed training. |
| TMA | There are 130 references to electronic documentation in the training material. |
| TMA | Training material is divided into 17 lessons. Each lesson is divided into subsections that are numbered with one decimal, e.g. 3.2. The subsection may be divided into unnumbered sub-subsections. Sometimes there is also a fourth heading level. In addition, step-by-step instructions contain a subtitle in the left margin. |

# 6 Learnability Guidelines and Improvement Suggestions

## 6.1 Overview of Guidelines and Improvement Suggestions

The following figure shows that we are now in the last phase of our research activities.

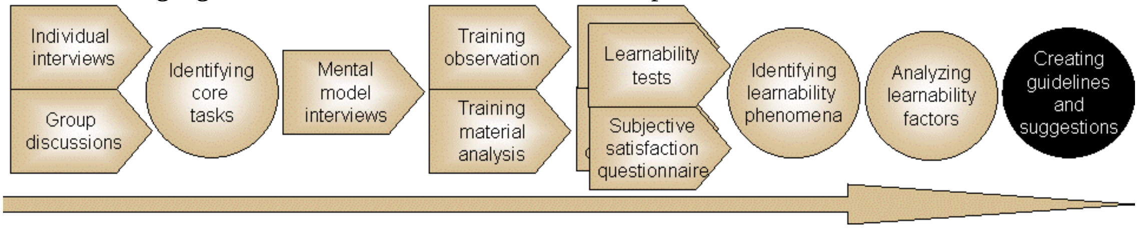



**Figure 24.** Progress of the research activities: suggestions for improving learnability.

The classification of factors affecting learnability and the list of learnability phenomena was used to create suggestions for improving the learnability of the Tekla Structures system. As there were over 200 observations concerning learnability and almost all of them would lead to a suggestion for improvement if processed further, it is not possible to present all the possible improvements here. Instead, some general guidelines were created and they were illustrated with examples. Many user comments related to the examples were presented in chapter 5.

For each factor affecting learnability, a set of guidelines was created. It was ensured that if all the guidelines related to certain learnability factor would be followed, all the observed learnability problems that were related to the same factor would be covered.

For some of the problems, detailed suggestions were created in order to demonstrate how the guidelines should be applied and in order to enable quick improvements in learnability.

When prioritizing the suggestions, the simple improvements that are easy to implement and are unambiguous should have a high priority. They will surely reduce the problems that novice users face but will not cause harm to expert users either. The cost-benefit ratio for these improvements is small, as they do not require a considerable amount of work for writing specifications and implementing.

However, it is also important to reserve time for larger redesign tasks related to task sequences or system features that users considered being difficult to learn. This requires careful needs analysis and optimization for both ease of learning and efficiency.

The prioritization of suggestions for improvement is not presented in this thesis, but a careful priority assessment for the suggestions will be made later together with members of the Tekla organization. The availability of implementation resources will affect the priority of different types of improvement suggestions.

## 6.2 User Interface

### Overview of Guidelines Related to the User Interface

27 guidelines altogether were formulated for improving the learnability of the user interface. The guidelines are presented in the table below and explained in more detail with illustrative examples in the following sections. The guidelines can be used as a checklist when designing new user interface elements. The parts of the user interface that were not covered in this study can also be compared against the guidelines and necessary adjustments can be made. Naturally,

applying the guidelines requires carefully considering the user interface element in question and possibly some expertise in human-computer interaction.

**Table 13.** Guidelines related to the user interface.

| Learnability factor | | Guidelines |
|---|---|---|
| **Visibility of operations** | 1.1 | Collect the related operations to the same location. |
| | 1.2 | Make all controls visible. |
| | 1.3 | Visually distinguish items that cannot be used in a certain situation. |
| | 1.4 | Support direct manipulation. |
| | 1.5 | Direct the user to giving the right input. |
| | 1.6 | Avoid states or if it is not possible, indicate the state clearly. |
| **Feedback** | 2.1 | When the user performs an action, the system must respond. |
| | 2.2 | If the user tries to perform an operation that is not possible in a certain situation, give directive feedback. |
| | 2.3 | Indicate the existence of hidden information. |
| **Continuity of task sequences** | 3.1 | Provide a direct link between successive actions. |
| | 3.2 | If doing a main task requires completing a subordinate task first, integrate the two task series. |
| **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 follow the desired design principles. |
| **Information presentation** | 5.1 | Organize menus so that they support user tasks. |
| | 5.2 | Design descriptive labels. |
| | 5.3 | Do not use system-oriented symbols or abbreviations. |
| | 5.4 | Do not present any unnecessary information. |
| **User assistance** | 6.1 | Provide information on existing objects. |
| | 6.2 | Inform users about errors. |
| | 6.3 | Give instructions for solving a problem. |
| | 6.4 | Design clear instructional texts. |
| | 6.5 | Provide advanced and beginner mode. |
| | 6.6 | Provide several forms of user assistance. |
| | 6.7 | Integrate user assistance with the system interface. |
| **Error prevention** | 7.1 | Automate operations that require unnecessary actions. |
| | 7.2 | Change errors to alternative paths of operation. |

Detailed suggestions for improving certain user interface elements were created in addition to the guidelines. A descriptive improvement suggestion is presented for each learnability factor in the following sections. More improvement suggestions are presented in appendix P.

For some learnability problems in the user interface, it was easy to create an improvement suggestion. If the user interface violated some generally known usability principle, the solution was to change the user interface to follow this principle. Some problem descriptions actually contained the solution in themselves. For example, as users looked for drawing commands only in the Drawing menu and the drawing wizard command was located in the File menu, the command could not be found. A self-evident solution was to move the command to the Drawing menu. For other problems, creating an improvement suggestion required more effort. This was especially true for problems that were related to a complex task sequence. To redesign these task sequences, it is necessary to know the user needs very well.

## Visibility of Operations

As was stated in section 5.2, a novice user cannot rely on his memorization in determining the available operations and the right input. Therefore, all operations should be made visible in the user interface. The guidelines suggest ways to do this in practice. Collecting related operations to the same location such as one dialog or menu (guideline 1.1) helps the user to find all the possible operations at the same time and not just a subset of them. Making all controls visible in the user interface (1.2) and hiding or disabling controls that cannot be used (1.3) guides the user to select the correct operations. Direct manipulation (1.4) is a natural interaction strategy and means that the user can manipulate objects that he sees directly, without having an intermediate layer such as a dialog in between. Directing the user to giving the right input (1.5) refers to showing an example or explicit instructions for inputting information. States should be avoided (1.6) because having different operations available at different times may confuse the user.

Example of applying the guideline 1.3 is deactivating the buttons that do not have any effect or cannot be used in a certain state.

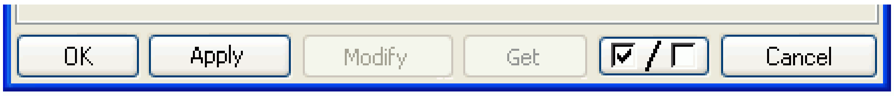



**Figure 25.** Buttons should be unavailable (gray) when they have no effect or cannot be used.

## Feedback

Feedback is crucial for novice users because otherwise they will be unsure of the results of operations as well as the system state. Therefore, the system must give feedback both when the user performs a successful operation (guideline 2.1) and when he tries to perform an operation but does not succeed (2.2). One kind of feedback is indication for the existence of hidden information such as objects that are not visible (2.3). There are many possibilities for the design of feedback messages. Messages could be shown in the status bar, where they do not bother experts but are useful for novices. In some situations, it may be feasible to open a message dialog to make sure that the feedback is read. In other situations, other than textual feedback might be the most appropriate.

Example of applying the guideline 2.2 is opening the drawing list automatically after a user has created drawings. Currently, the only indication about a successful drawing creation process is a text in the status bar. Many novice users were wondering if the system had done anything even though it actually had created drawings as the user requested.

## Continuity of Task Sequences

Task sequences should be planned so that when the user starts an operation, he is led until the end of it by the system. Successive actions should be linked so that the user never needs to access several menus or several separate dialogs to complete one action sequence (3.1). Instead, the dialogs may be linked with buttons or simply combined (3.2). Making task sequences continuous requires redesigning parts of the user interface. However, this should be done, because continuing task sequences will make the use of the system much easier for both novices and experts.

Example of applying the guideline 3.2 is integrating the numbering and drawing creation processes. Numbering is required before creating drawings but users often forget it and get a warning message. They should be provided the possibility to number parts directly from the message dialog.

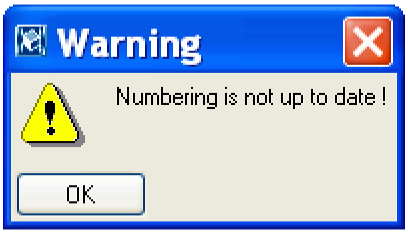



**Figure 26.** The old message that was shown when numbers had not been assigned.

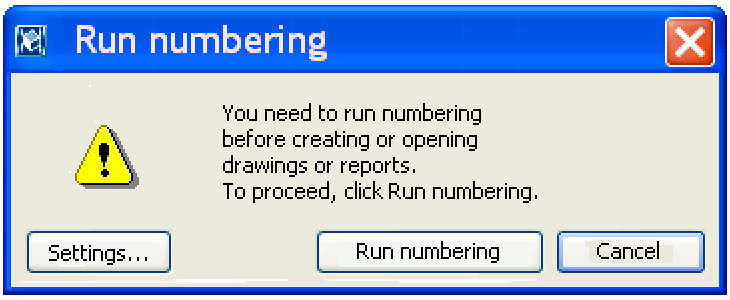



**Figure 27.** A redesigned message that allows for numbering directly from the message.

## Design Conventions

Following design conventions enables users to transfer their skills from applications they are familiar with. Most users are familiar with common office systems, web applications, and drawing software, for example, and therefore using similar controls (4.1) and task sequences (4.2) will enable users to transfer their skills to a new application. Templates should be provided for creating documents, to assist the user in creating output that follow the conventions of the particular system (4.3)

Example of applying the guideline 4.2 is replacing the Filter field on the Open file dialog with a Browse button. Currently, the user can enter a path into the Filter field to see files in a certain directory. The conventional method to see the files is to click a Browse button that opens a separate window where the user can graphically browse the files.

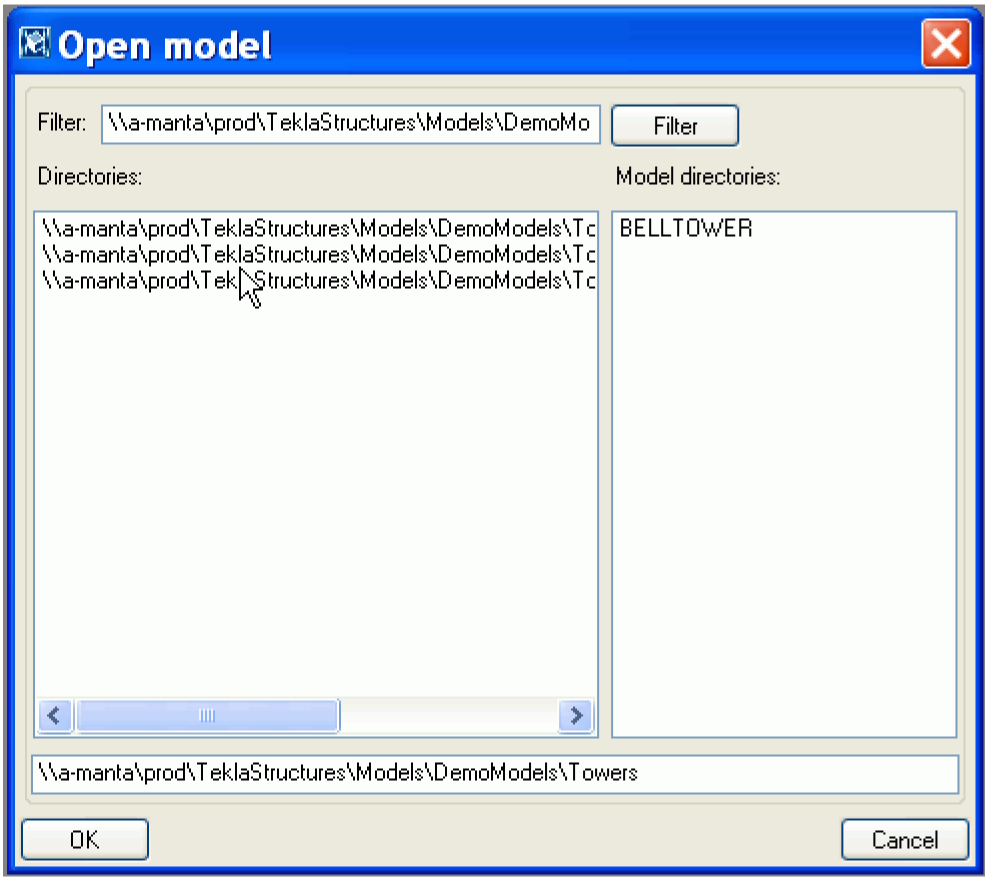



**Figure 28.** The old dialog for opening files.

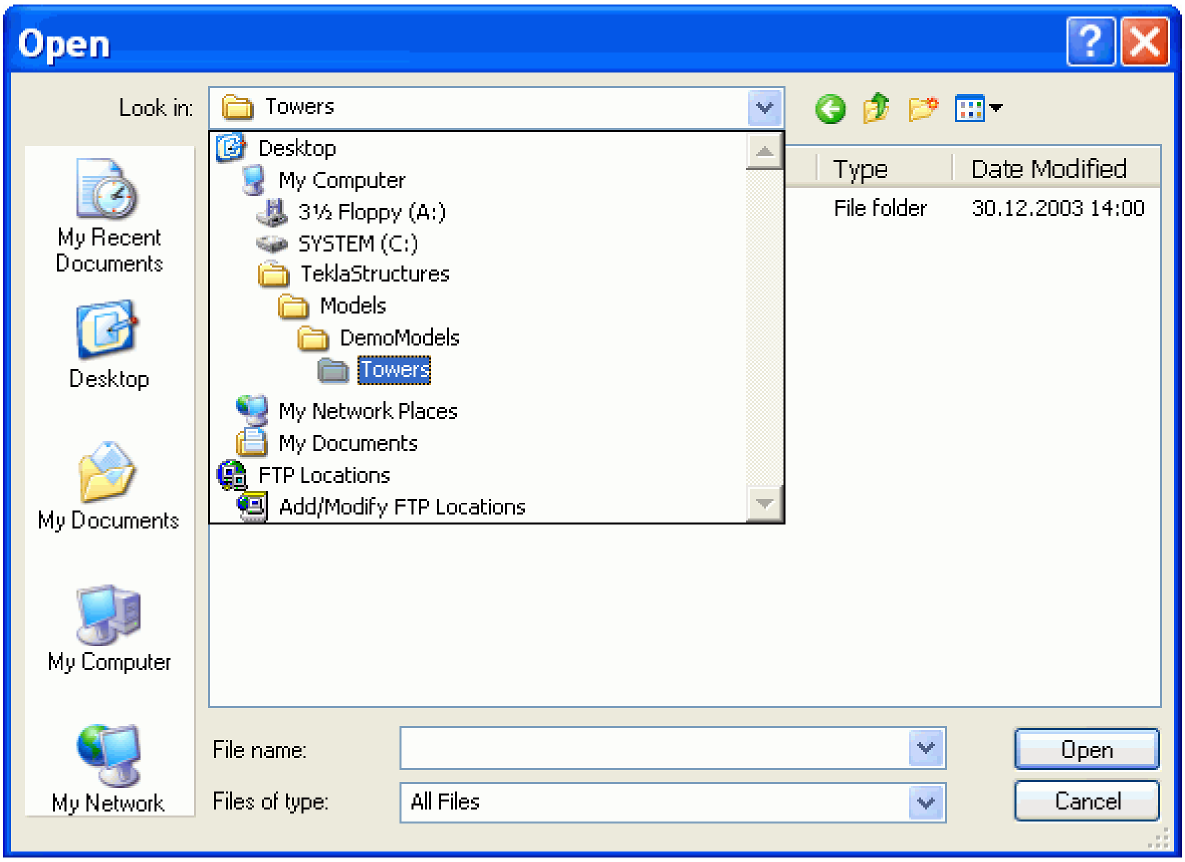



**Figure 29.** A redesigned dialog for opening files.

## Information Presentation

Clear information presentation is especially important for novice users. The guidelines present some practical suggestions for how information could be presented clearly. Firstly, menus should be organized according to users' tasks (5.1) because that will both make them more efficient to use and help the user to find commands in menus. Secondly, labels should be very descriptive (5.2). They should not be made so short that understandability suffers. No system-oriented abbreviations or symbols should be used or at least explanations for them should be easily available (5.3). All the information that is not needed, such as advanced options, should be hidden (5.4) because it will cause unnecessary cognitive load.

Example of applying the guideline 5.3 is replacing the symbols <--' and <-- with text on the dialog that is used for setting the part mark content. The symbols have been designed to resemble the keyboard labels for Enter and Backspace, but users did not understand their meaning.

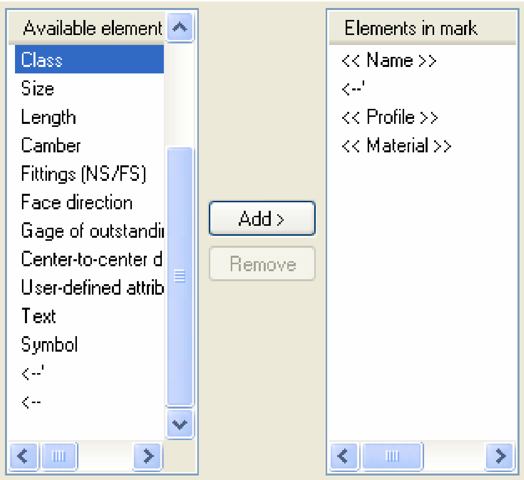



**Figure 30.** Old symbols.

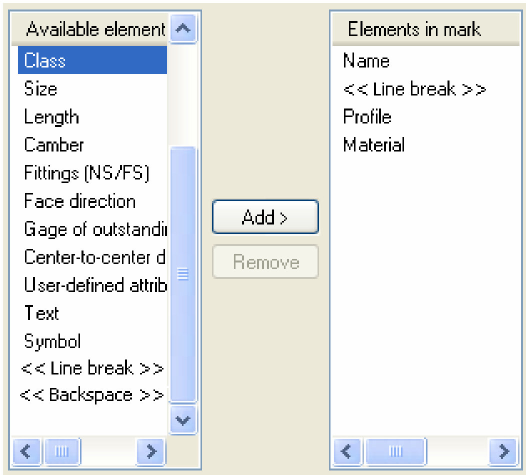



**Figure 31.** Symbols replaced with text.

## User Assistance

User assistance strategies should be designed and implemented in the user interface. Information on existing objects should be easily available (6.1). Users should be assisted when they encounter errors (6.2) or face problems (6.3). The system should recognize these situations and instruct the user to fix the issue. Instructional texts should be clearly worded and complete (6.4). It may be necessary to provide an advanced and beginner mode to avoid burdening an expert user with unneeded instructions but to provide enough help for a beginner (6.5). Several forms of user assistance may be needed to serve all information needs (6.6). In addition to a separate help file, there could be short help messages related to different dialogs or controls and the messages could be accessible by a single mouse click. The status bar should also be used for presenting messages. In critical situations, it may be useful to show additional pop-up messages. User assistance should be integrated very closely with the user interface elements (6.7) so that user could easily access the information that is related to the user interface element that he is currently operating on.

Example of applying the guideline 6.1 is providing explanations for the one-letter abbreviations in the drawing list. Example of applying the guideline 6.4 in turn is giving detailed instructions for picking points in the status bar. The current user interface shows the default prompt "Pick polygon position" in most situations. The prompt is not informative enough, and therefore it should be replaced with a more precise prompt.

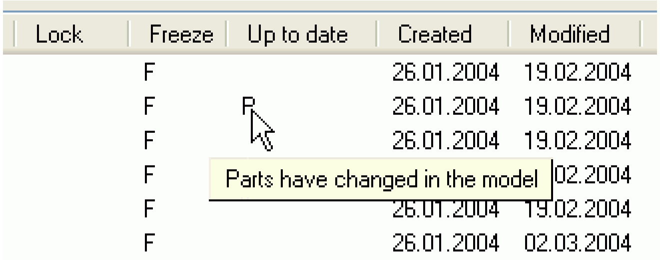



**Figure 32.** Abbreviations should be explained with a tooltip.

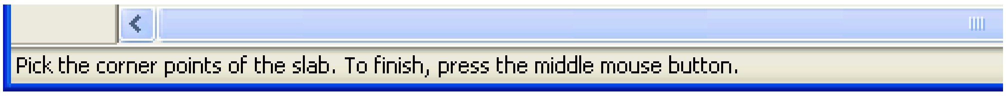



**Figure 33.** Instructions should be given in the status bar.

### Error Prevention

The errors that many novice users encounter should be prevented if possible. A set of most common errors were found in this learnability study and correcting them will reduce the number of errors considerably, even though there may be also other common errors that were not revealed in this study.

There are two types of errors that are especially easy to prevent. Automating the operations that do not require decision from the user (7.1) will reduce the amount of errors. In addition, the harmfulness of some errors can be reduced by providing an easy way for the user to recover from them (7.2). In some cases, an error could be changed into an alternative path of operation by allowing the user to perform an action that was not previously allowed.

The guideline 7.1 can be applied to for example the Create grid views dialog. When grid views are being created and there is only one grid in the model, the grid should be selected automatically. Currently, the user always has to select a grid in the model before clicking the Create button on the dialog, even if there is only one grid in the model. If he forgets to do it, he gets an error message.

## *6.3 System Structure*

### Overview of Guidelines Related to the System Structure

11 guidelines concerning the system structure were formulated. The guidelines are summarized in the following table and presented in more detail in the following sections. The guidelines can be referred to when designing new features to the system or when planning which new concepts are introduced to the system. The guidelines remind of issues that may either hinder or support correct mental model formation.

**Table 14.** Guidelines related to the system structure.

| Learnability factor | Guidelines | |
|---|---|---|
| **Differences in functionality** | 1.1 | Do not change the functionality but assist users with learning it. |
| **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. |
| | 4.3 | Provide user assistance. |

Descriptive examples of improvement suggestions are presented in the following sections after listing the guidelines for each of the learnability factors. More improvement suggestions can be found in appendix P.

For some learnability problems concerning differences in mental models and system structure, it was easy to create a suggestion for improvement; for others, it was not even possible. If problems are due to differences in the functionality of software applications, the only solution may be to aid the user in learning to understand the difference. Eliminating the difference may not be possible or even desirable. However, for problems that were due to ambiguity of concepts or unclear information presentation, it was possible to create a suggestion for improvement. Some more research should be done to check if the suggested terminology corresponds to the one that is used in the real world.

## Differences in Functionality

Usually it is not desirable to change the functionality of the system to correspond to the software application users are familiar with, even if that would facilitate learning. However, the most fundamental differences between the system in question and other common software applications should be taken into account when planning training or creating instructional documents and made explicit by designing a descriptive user interface. (Guideline 1.1)

This applies to the fact that drawings and views are separate concepts in the Tekla Structures system, whereas in most two-dimensional drawing applications they are the same. The difference cannot be eliminated but it should be taught in training and explained in the training material.

## Differences in Interaction Styles

Allowing users to interact with the model with the strategies they are familiar with would aid novice users but in most cases, it would also be advantageous for experienced users. As was already mentioned earlier, design conventions for controls and task sequences should be followed (2.1). In addition, if there are established practices for interacting with objects in software used in the same domain, users should be allowed to use the same interaction strategies when using the new system as well (2.2). The same holds for menu titles; established practices for menu names and organizations should be followed (2.3). All these issues will help the user to transfer their skills into new software.

Guideline 2.2 can be applied for example by providing the user the possibility to resize objects by selecting a line or a face and dragging it to the desired direction. Currently, parts can only be resized by entering dimensions into the part properties dialog or by manipulating points that define the part.

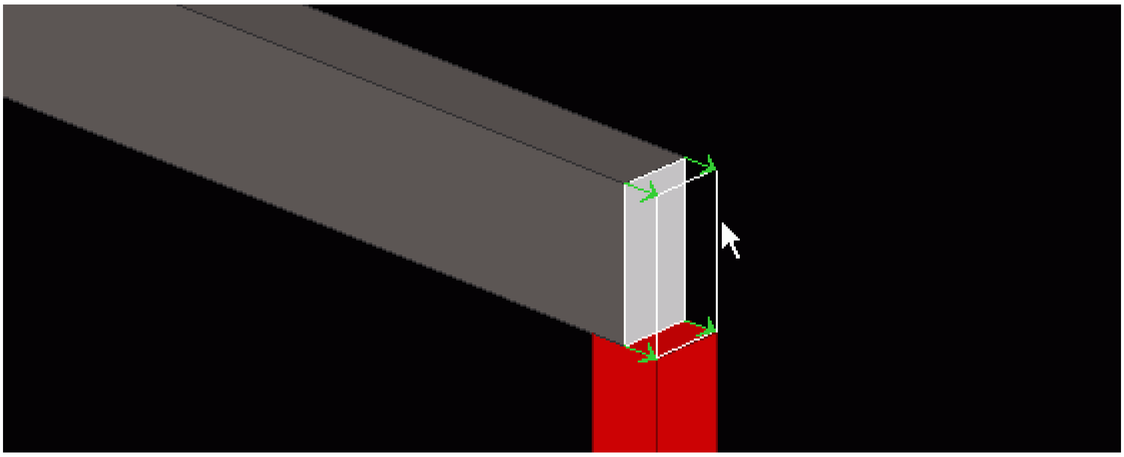


**Figure 34.** Users should be allowed to resize parts by dragging.

## Concept Clarity

Clarity of concepts is important because it will enable users to add the new concepts correctly into their mental model. Terminology choices are crucial in this. Terminology that is already familiar to users should be adopted (3.1), whereas terminology that is familiar from wrong contexts (3.2) or system-oriented (3.3) should be avoided. Concepts can be clarified with pictures or symbols on the corresponding dialogs or controls (3.4).

Guideline 3.1 can be applied to many individual commands that have nondescriptive names. For example, the label of a button Freeze should be changed to a more descriptive one. A suggestion for the label is Keep modifications as that is what the command essentially does.

## Completeness of Information

Complete information will help the user to understand the system structure and functionality, which in turn will help to form a correct mental model of the system. Explaining new concepts (4.1) will help users to assimilate new concepts into their mental model, and descriptive instructions for performing actions (4.2) will help to assimilate new task sequences. User assistance in general (4.3) will help users to understand the operating logics of the system. It is important to note that the need for user assistance stems partly from the need to understand the system and to form a correct mental model.

Example of guideline 4.2 is restructuring the AutoDefaults setup dialog. The dialog should communicate its purpose and usage to the user so that even a novice user could understand for what purpose the dialog is intended. The name of the dialog should be changed to a more descriptive one, Default properties for connections, as this name describes what the dialog is actually used for. A short instructional text should be added onto the dialog. The connections should be grouped into logical groups as elsewhere in the system, e.g. Splice connections, instead of the numbered groups, e.g. Components 2.

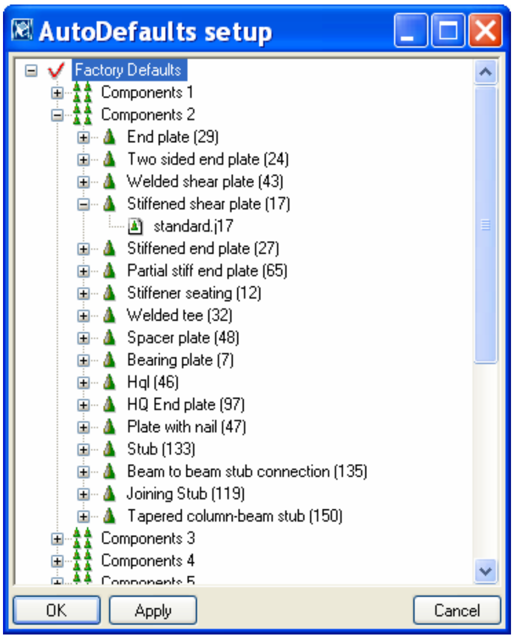



Figure 35. Old AutoDefaults setup dialog.

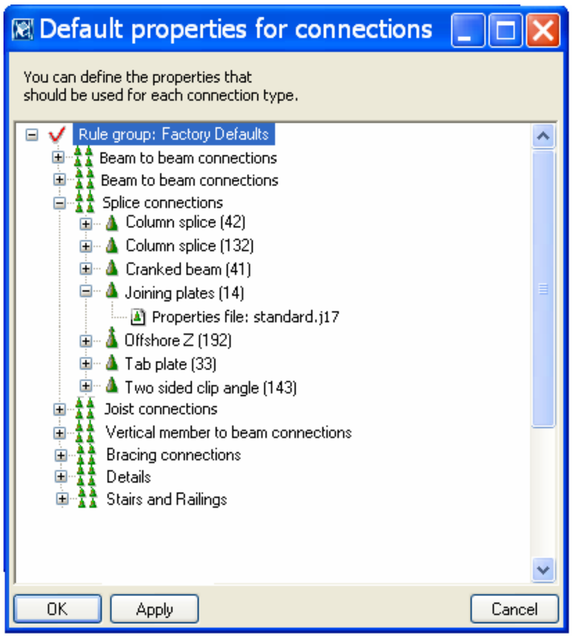



Figure 36. Dialog with descriptive information.

## 6.4 Training

### Overview of Guidelines Related to Training

26 learnability guidelines related to training were formulated. The guidelines are summarized in the following table and presented in more detail in the following sections. The guidelines are based on the observations made in this study and they are expected to summarize the issues that affect the learning result the most. The contents and organization of training can be compared against the guidelines to find the necessary adjustments.

**Table 15.** Guidelines related to training.

| Learnability factor | 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 should be used. |
| **Exercises** | 2.1 | Introduce the basic form of an operation and require learners to apply it to new situations. |
| | 2.2 | Encourage learners 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 | Tell users about the available documentation. |
| | 4.2 | Demonstrate how to use the documentation. |
| | 4.3 | Tell how to contact support personnel |
| | 4.4 | Address the most common causes of error. |
| **Motivational content** | 5.1 | Summarize the contents of the training in the beginning of it. |
| | 5.2 | Address the issues that each learner will need in his work. |
| | 5.3 | Follow-up with learners if possible. |
| **Coverage of essential system functions** | 6.1 | Get to know the participants 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. |
| **Material types** | 7.1 | Provide integrated help. |
| | 7.2 | Provide printed material or dual monitors in training. |
| | 7.3 | Limit the amount of material. |
| | 7.4 | Design a clear layout for material. |
| | 7.5 | Provide material in the native language if possible. |
| | 7.6 | Provide search possibilities. |

Detailed suggestions for improvement were also created and examples of them are presented in the following chapters. More suggestions can be found in appendix P.

Suggestions for training are not as exact as suggestions for user interface or system structure. This is due to the fact that the user interface and the system are 'static' and remain the same in all situations whereas training is always customized according to user needs. However, the suggestions given here can be used to modify the basic structure of the training, and each instructor can adjust the basic structure to fit the needs of each learner group and learning situation.

## Conceptual Information

It was written above that users did not understand all the concepts that are related to the task sequences they perform. It was also noted that some more explanations for concepts and interaction principles are needed. It would be especially important to explain the meaning of terms that are not used in real life such as AutoDefaults, AutoConnections or Drawing classifier (1.1). The relationship between concepts should also be explained (1.2), as well as the operating principles of the system (1.3). If these are not explained thoroughly, users may learn to perform a task sequence by looking at the example given by the demonstrator. However, they may soon forget it if they have not understood the idea behind it.

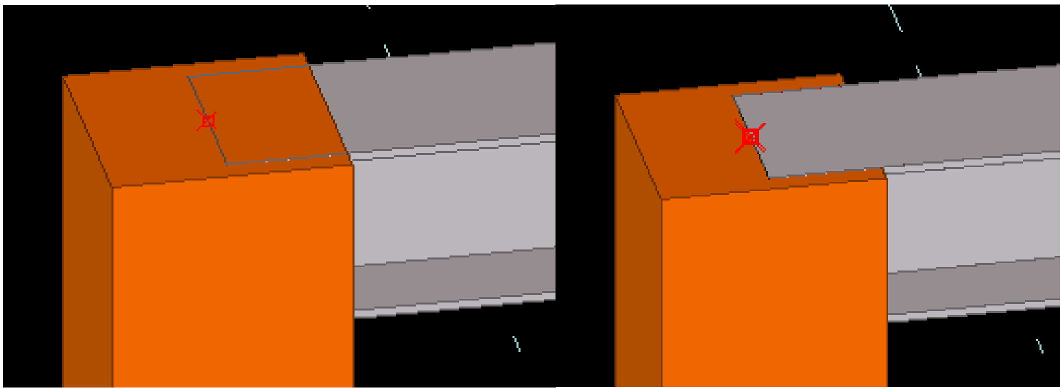
An example of applying the guideline 1.2 is explaining the relationship between different connection and component types, such as System components / connections, Custom components / connections, and AutoConnections. A visual connection map could be created to illustrate the types of components and connections as well as situations in which they can be used.

## Exercises

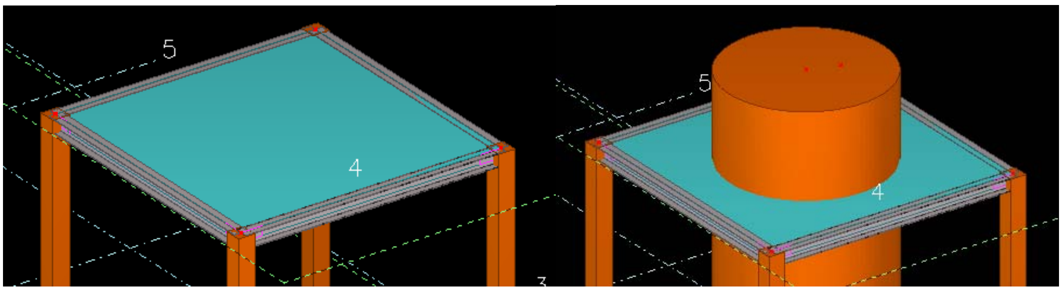
Introduce the basic form of an operation and require learners to apply it to new situations.

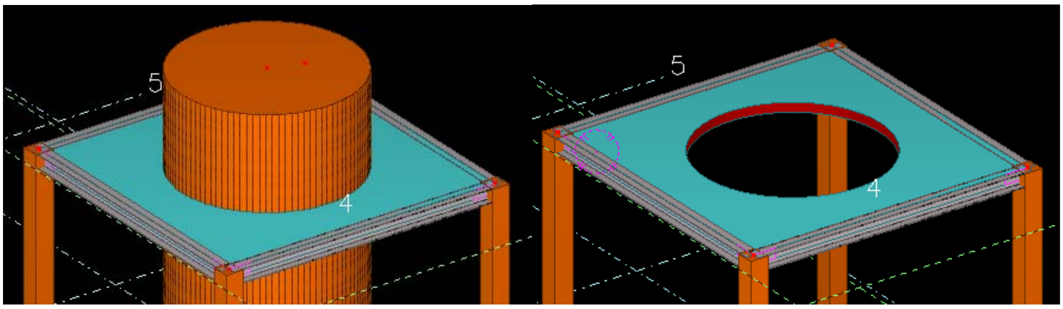
Exercises should be designed so that they produce long-term learning results on all the central operations. To facilitate deep learning, learners should apply each basic operation into a new situation (2.1). In practice, the instructor could demonstrate a basic operation briefly on the screen and after that, the learners could be given exercises that require applying the demonstrated operation. In addition, the instructor should encourage learners to process the information actively (2.2). This can be done for example by asking questions that require analyzing the user to analyze the phases of the exercises further. To motivate learners and to clarify the connection between the exercises and accomplished tasks, the goal of each exercise should be stated (2.3). In addition, the conditions in which each operation can or should be performed should be explained (2.4).

Examples of task sequences that could not be memorized as they were only practiced through following step-by-step instructions in the training are creating fittings, creating AutoConnections, defining Custom Components, and creating part cuts. These basic operations should first be demonstrated by the instructor in a very simple case such as cutting the part of the beam that overlaps with a column. The learner should then be asked to apply the operation to a new situation such as creating a rectangular hole using a part cut. (See guideline 2.1.)



**Figures 37 and 38.** Simple part cut.






**Figures 39 – 42.** Applied part cut.

## Instructions for Basic Interaction

Even though basic interaction strategies may be clear for an expert or an instructor, they will not become clear for a novice user during the training unless they are explicitly explained. Therefore, instructions should be given for interacting with objects (3.1), adjusting the basic settings (3.2), and using the basic controls (3.3).

For example, the following interaction techniques (see guideline 3.1) should be explained and demonstrated in the training: moving parts, using handles, snapping and picking, determining view properties, and changing part size. Users had problems with performing these basic actions even after the training.

## Instructions for Solving Problems

Solving problems is an essential activity in using software applications, and users should be prepared for it already in the training. To do this, the available documentation should be introduced (4.1) and use of it should be demonstrated (4.2). Users should also be given the contact information of support personnel (4.3). The most common causes of errors, some of which have been revealed in this study, should be addressed in the training (4.4).

Guidelines 4.1 and 4.2 should be applied by demonstrating the use of the help file in the training. The participants should be told how the help file can be installed (it requires separate installation) and how it can be accessed. An overview of the contents of the help file should be given in the training.

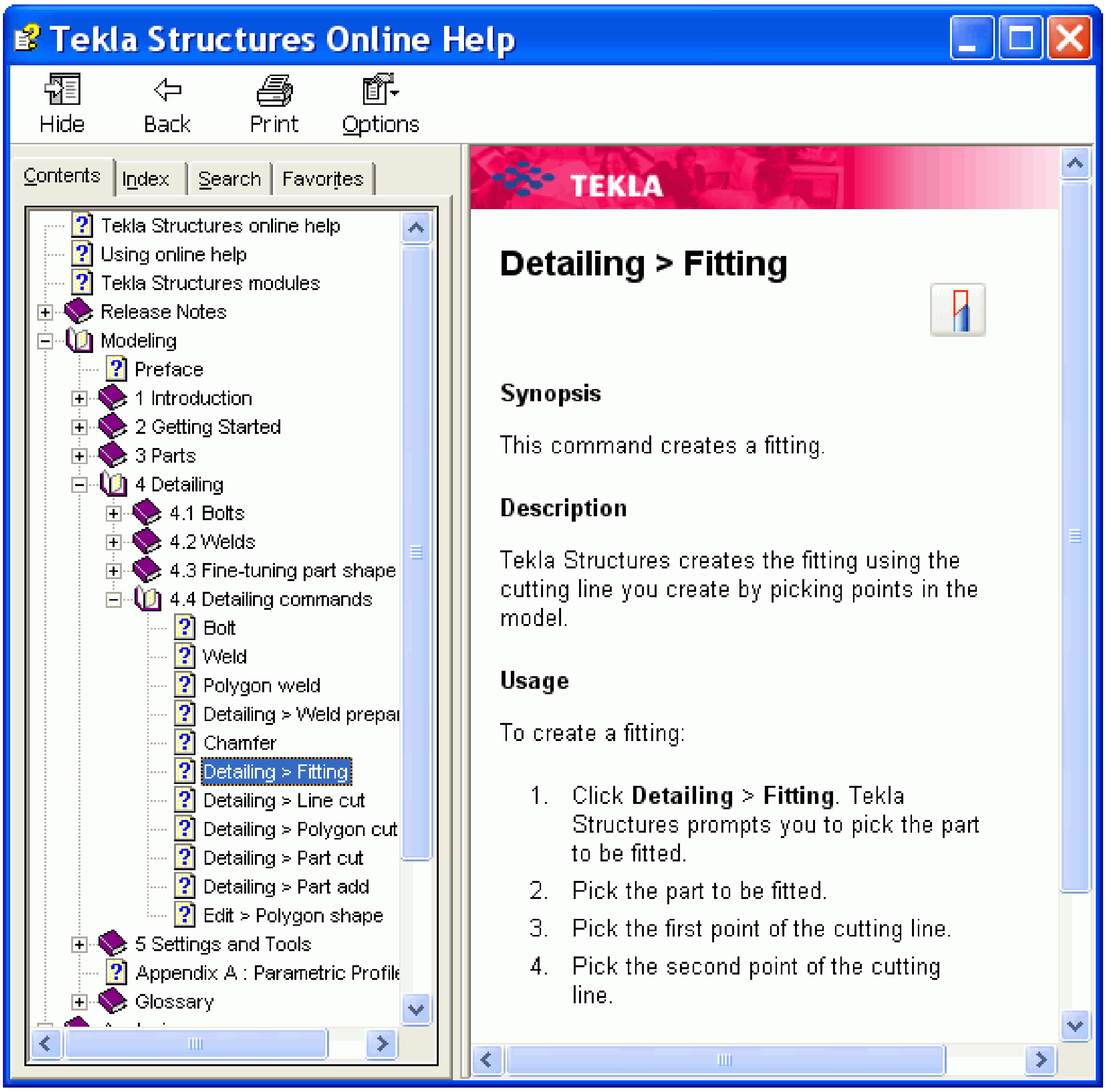



**Figure 43.** The help file should be introduced in the training.

## Motivational Content

Some orientational and motivational material should be added to the training material. Summarizing the contents of the training in the beginning of the training session (5.1) will motivate the learners and help them to start to connect the learned material to their existing knowledge. An important factor in determining the motivation level is whether the training addresses the issues that each user needs in their work (5.2). The contents of the training may even need to be customized a little for each participant. Keeping the motivation level of users high also after the training may require contacting them after the training (5.3) and asking about possible problems or wishes.

The guideline 5.1 could be applied by providing a written overview of training contents to the participants and going through the document in the beginning of the training. The table of contents of the training material should also be reviewed with the participants. Some or all of the orientational material could be sent to participants beforehand.

### Coverage of System Functionality

To ensure the usefulness of training, user needs should be carefully researched and analyzed. Getting to know participants and their needs (6.1) was mentioned above as a motivating factor and it is essential in determining which parts of the system should be covered in a certain training session. Information on the core tasks of the users should be used for adjusting the amount of material for each task (6.2) and time that is spent with each of them in training (6.3).

Several issues related to the guideline 6.1 were observed during the research activities. For example, most users said they need to export and import data to work with colleagues that use different software applications, but exporting and importing were not covered in training. The training contents should be adjusted so that these issues will be covered.

### Material Types

When material for a certain learning event such as training is being chosen, advantages and disadvantages of each material type should be compared to the specific requirements of the learning situation. Several observations in this study support the requirement for an integrated help (7.1) that means instructions integrated into the user interface instead of shown in a separate window. In addition, it was noticed that it is necessary to provide printed material or dual monitors in training (7.2). The user can be aided in going through and understanding the material by limiting the amount of material (7.3), designing a clear layout (7.4), providing material in native language (7.4), and providing search possibilities (7.5).

To apply guideline 7.4, the material structure could be clarified by formatting the heading styles. Only one heading level without outline numbering should be included. In the printed material, each subsection could be started on a new page.

# 7 Conclusions

## *7.1 Comparison of Methods for Evaluating Learnability*

Several methods were used in this study for evaluating learnability. The advantage of using many different methods is that they complemented each other and enabled us to assess different phases of the learning process. The disadvantage of using several methods is that the data obtained with different methods is not always commensurate. Extracting a set of learnability factors from this diverse material was challenging. However, a classification that seems to fit all the data could be created.

Of the individual methods, we consider the mental model interview, training observation, and scenario-based learnability tests to have been the most useful. They produced most material for the analysis of learnability factors and creation of suggestions for improvements. The training material analysis method and subjective satisfaction questionnaire were considered less useful but they also produced some information for the analyses. Next, we assess the usefulness of each method separately. This information can be used by other researchers when they plan methods for evaluating learnability.

In the mental model interviews, users were able to verbalize their expectations for the user interface that they saw in front of them rather well, even though it was speculated beforehand that it might be difficult for users. Letting them to try the simplest functions themselves revealed the difficulties that a user may face when he tries the system for the first time.

In the training observation, a lot of information was acquired on both the level of learnability of different task sequences and specific learnability problems in user interface elements. In addition, training methods could be assessed and factors that affect the effectiveness of training could be extracted. The Tekla Structures system differs from many other systems in that training is regularly organized for new users. This allowed us to get to observe training sessions easily. The problem related to the training observation method was that as there were many users doing their tasks at the same time, the observer could not see all their actions and may have missed some interesting point.

The training material analysis method supported the training observation method. However, the training material analysis method contributed to the formation of the learnability factors less than the other research methods, as its focus was very narrow and also because there was not much observational material on the training material usage available. On the other hand, training sessions follow the structure of the training material very closely, and therefore the training material must be addressed too if changes are to be done in training sessions. For self-learners, the quality of training material is even more essential.

The scenario-based learnability test followed the test setup of traditional usability tests. Usability tests have been considered as an effective method for finding usability problems in the user interface. The method proved to work well in this study too. A lot of information on usability problems and factors affecting learnability was acquired.

The results of the subjective satisfaction questionnaire could be used to compare the observational information to the subjective opinions of users. These two proved to be rather well in line even though there were some discrepancies. The results of the subjective satisfaction questionnaire also indicate which features users consider the most inconvenient to use. However, the problem of the questionnaire method is that different users may interpret the questions differently which may affect the results. Different users may also use rating scales differently.

During the research, we have been thinking especially over the relationship of training observation and learnability tests as methods for evaluating learnability. In many cases, training is not regularly organized for new users, but if it is, it should not be overlooked as a learnability evaluation method.

In training observation, we noted 111 learnability phenomena, and in learnability tests, we noted 137 phenomena. Thus, the difference in the number of phenomena is not large. However, the training observation enabled us to observe a larger selection of system functions than one-and-a-half hour learnability tests. Training observation also produced more information on the learnability of task sequences whereas learnability tests concentrated on the learnability of single user interface elements. In addition, the total time spent by the researcher, including that of material preparation and data analysis, was longer for learnability tests than for training observation. Training observation required less preparation because training contents were selected by the instructor to represent the task domain in the best possible way. Naturally, learnability tests have their advantages; for example, the observer is able to see every action that the user does, whereas in training observation, he needs to divide his attention among many participants. In addition, learnability tests organized in laboratories can provide the opportunity to stabilize environmental variables or use sophisticated tracking and recording equipment.

In this study, we had enough resources to use the combination of training observation and learnability tests, among other research methods. We think that using several methods produced more extensive and useful results than any single method could have produced.

## 7.2 Reliability and Validity of Data

In qualitative research, the outcome of the analysis is always a combination of the empirical data and the interpretation of the researcher. Even if the researcher tries to avoid biasing, he always brings his own preconceptions and ways to interpret data into the research. This reduces the reliability of the data even though not usually dramatically.

The observer bias can be partly avoided by planning the observational methods carefully and using predefined templates for recording observational data. Observation templates were used in this study too. However, even the use of templates requires some decisions from the observer: which events she records as learnability problems, how she writes down the user behavior, or if she interprets the actions of users correctly. The classification of learnability factors is especially prone to researcher bias. The researcher has classified the factors according to her best understanding, but several other classifications would certainly be possible too. Nevertheless, the classification developed in this study was compared with classifications made in other studies, and the results seemed to support each other rather well. Yet, this does not eliminate the fact that the researcher effect is certainly seen in the classification of the factors.

Some possible biases can be forecast by examining the background from which the researcher is looking at the observational data. The writer of this thesis has experience in the human-computer interaction branch but less experience on the educational branch. Even though this study includes also factors concerning the effect of training on learnability, the emphasis is on the effect of the user interface. A pedagogist would probably use different methods for evaluating training results. The methods used in this study come mainly from the research body of human-computer interaction.

Even though the learnability factors were extracted directly from the observational data, the three categories for learnability factors, namely user interface, differences between the mental models of users and the system structure, and training, were predefined. The question arises whether the categories were chosen correctly. However, the categories were chosen on the basis of literature research that was done in the beginning of the project, and the three issues have been recognized by many researchers, even though they have not usually been studied together.

The number of test subjects, six users, is rather small, which also affects the reliability of the results. The number was considered very small especially in the second observation, in which users were divided into two groups whose performance differed from each other: users that had used the system after the training, and ones that had not. The groups contained only two and three users, and therefore it is hard to draw conclusions of the performance of these two groups.

The subjects of the study were slightly untypical in that four of them had two years or less experience of building modeling. On average, a user that starts to use the Tekla Structures system has worked in the building modeling branch for several years during which time he has used two dimensional drawing software applications. A different combination of users as subjects of the study would probably have produced slightly different results. However, it is assumed that similar issues affecting learnability would be found even with somewhat different user groups. If the study would have concerned efficiency of expert users, the differences between user groups may have been larger, as each user would have developed his own strategies for interacting with the system and the strategies may have varied a lot.

An issue that adds to the validity of the results is that many of the learnability phenomena were observed with several research methods. In addition, the same phenomena were often encountered with several users. If many of the six users face a certain phenomenon, it can be expected that of a larger population, a remarkable percentage of users would face the same phenomenon.

All the research templates, questionnaires, and other material that were used in empirical methods are included in the appendices. Therefore, the experiments can be repeated by any researcher. The results would probably be slightly different because of different user groups and statistical issues, but it can be expected that observations supporting each of the learnability factors presented above could be made.

## *7.3 Implications of the Results*

18 factors affecting learnability were found in this study. These factors can be used as a general framework for understanding the learnability of Tekla Structures.

This study also includes suggestions for improving learnability, presented as general guidelines and illustrated with detailed examples. The detailed level suggestions can be implemented without a lot of additional functional specification and therefore they enable quick learnability improvements to the Tekla Structures system. The learnability guidelines in turn can be applied even to parts of the system or training that were not included in this study.

The need for researching how these suggestions for improvement will actually affect learning will be discussed in section 7.6. There is also a need to research how the improvement suggestions would affect the performance of expert users. When creating the suggestions for improvement, the requirement of efficiency was kept in mind, as it was recognized that improving learnability should not cause harm to expert users.

Throughout our study, three aspects of learnability are addressed. The learnability factors, guidelines, and detailed suggestions for improvement are divided into corresponding three groups. The first group contains issues related to user interface, the second group contains issues related to differences between the mental models of users and the system structure, and the third group contains issues related to training. These three aspects were chosen after a literature study because they repeatedly occurred in the literature considering learnability or the learning process of software users. The aspects could of course have been chosen differently, but we found the chosen three aspects to be a feasible framework for analyzing the data we collected.

The learnability factors, guidelines, and suggestions for improvement that are related to the user interface should be utilized when analyzing the existing parts of the user interface of Tekla Structures or planning new interface elements. They contain issues that make the user interface easy for novices to learn.

The learnability factors, guidelines, and suggestions for improvement that are related to mental models should be considered when designing new features to the Tekla Structures system on a conceptual level or when reconsidering the existing system structure. They should also be taken into account when introducing new features of the Tekla Structures system to expert users or when introducing the system to novices. Assimilating the new features as a part of the existing mental model should be supported. As the learnability factors in this group suggest, the concepts associated with the system and the way the information on the concepts is presented are crucial in determining how correct the mental model of users will be.

The learnability factors, guidelines, and suggestions for improvement that are related to training provide information on how the learning process of Tekla Structures system can be supported with training or training material. The guidelines related to training should be compared against the existing training setup or used as background information when planning new training sessions. Changes in the user interface cannot substitute the user interface development, but they can provide quick help with user interface elements that users consider difficult to learn.

We expect that our classification of learnability factors, the learnability guidelines, and the improvement suggestions are useful for not only the developers of the Tekla Structures but also the body of human-computer interaction researchers. Not too many classifications of factors affecting the learnability of complex systems exist and therefore we expect our classification to be valuable. As the information technology penetrates to the society, an increasing number of complex systems for domain experts are being used and therefore information on the factors affecting their learnability is useful. The classification of factors is based on a body of empirical data collected with several research methods. The classification was created with the grounded theory method, which is a generally accepted method for creating a theory that fits the available set of data.

The guidelines were created on the basis of the observed learnability problems. They are very thorough and that is one reason for which they should be interesting for other researchers as well. Several sets of usability guidelines have been presented in the literature, but sets of learnability guidelines are less common.

## *7.4 Applicability of the Results to Other Complex Systems*

The factors affecting learnability and suggestions for improving learnability have been created on the basis of the learnability study concerning the Tekla Structures building modeling system. However, it would be desirable to be able to apply the results to other complex systems as well.

It was mentioned in section 5.1 that the grounded theory method can be used for creating a theory that fits the available set of data. The method does not guarantee a theory that can be generalized. Naturally, if there is another system that is very similar to Tekla Structures, a large portion of the results can be generalized to it, but not necessarily all. The factors concerning the learnability of the user interface and mental models of users are presented on such a general level that they could be applied to other, dissimilar systems as well, but it is left as the responsibility of the person that studies the other system to estimate their applicability to that particular system. The factors concerning training may be more difficult to apply to other systems as training courses are not even available for many of them and if they are, they may vary in duration, scope, learning goals, and user population.

Applying the learnability factors to other systems, similar or dissimilar, may require some expertise on human-computer interaction. For example, most developers will probably argue that the visibility of operations is a desirable goal for a system design, but it is not easy to determine how the operations should be made visible.

The guidelines for improving learnability can also be applied to other complex systems, with a reserve of them being inapplicable to systems that differ from Tekla Structures a lot. The detailed suggestions for improving the learnability of Tekla Structures can in turn be used to clarify the meaning of each general guideline.

## 7.5 Comparing the Results to Previous Studies

It is interesting to compare the factors affecting learnability and learnability guidelines that were found in this study to classifications and guidelines presented by other researchers. Most of the previous learnability research covers the effect of the user interface alone. This is one of the reasons why we wanted to create a classification of our own and not use the existing classifications as a basis for improving learnability. Some classifications and guidelines concentrating on the effect of the user interface were presented in section 2.2. Another reason for creating our own classification is that the application domain and the context of use among other things affect the learnability requirements to a great extent. The preexisting classifications and guidelines would not necessarily conform to the system in question very well.

The classification of learnability factors created by Elliott et al. (2002) contains the following items: transparency of operation, transparency of purpose, accommodation, and accomplishment. The first item presented by Elliott et al., transparency of operation, is very close to the first user interface factor found in this study, visibility of operations. The second item presented by Elliott et al. (2002), transparency of purpose, has commonalities with several learnability factors found in this study but is not essentially the same as any of them. The third and fourth factors, accommodation and accomplishment, are very different from the factors we found. They are very general and orientate towards the experience of the user. Our factors address the characteristics of an easily learnable system on a rather detailed level.

The classification presented by Dix et al. (1993, p. 131-137) contains the items predictability, synthesizability, familiarity, generalizability, and consistency. Predictability and synthesizability are close to the factors visibility of operation and feedback. Familiarity is related to concept clarity and to differences in functionality and interaction styles of software applications. Generalizability and consistency are related to design conventions. As is the case with most of the learnability factors classifications, the factors found by Dix et al. are associated mainly with the user interface. Training factors are out of the scope of his classification. Mental model factors are loosely connected with some of the factors found by Dix et al. but they do not use the term mental model in his classification.

Rieman et al. (1994) in turn stressed the importance of consistency and presented the following guidelines: analogies should be used but only if they are inside the context of the program or its class; graphical cues should be provided that indicate the categories that have similar functionality; labels should be designed to indicate link the control to its effect; and clear and immediate feedback should be provided (see section 2.2). Our guidelines include similar items as Rieman's, but our guidelines are more detailed. This may be either a burden or a benefit: going through a long list of guidelines requires a lot of time, but if it is done thoroughly, the result is probably better than with only a few guidelines.

Our learnability factors and guidelines that are related to differences between mental models of users and system structure are in line with the theory on mental models (see section 2.3). According to Chandra and Blockley (1995), learning can be seen as a change in the mental model. The more the existing mental model on system functionality and interaction strategies differs from the structure of the new system, the more changes are required in the mental model and the more difficult the learning process will be. This supports the first and second learnability factors related to mental models. In addition, it was stated in section 2.3 that mental models are based on knowledge that is obtained from outside sources and on observations and experiences that a human acquires. The third and fourth learnability factors, clarity of concepts and completeness of information, are related to aiding the user to adjust his mental model to correspond to the actual system structure.

Connections can also be found between the factors related to training and theory on human learning process that was presented in the beginning of this thesis. In section 2.4, the methods for delivering conceptual information and doing exercises were discussed in detail and a wealth of studies have addressed these issues. Even if they are a central part of all training sessions, there is no general agreement on what is the best practice for delivering conceptual information or doing exercises. The issues of covering the essential system functions, including motivational and orientational content, and teaching basic interaction and problem solving strategies are more straightforward. The material type that suits each learning situation may differ, but the appropriateness of material certainly is an important factor in determining the learning results.

## 7.6 Suggestions for Further Research

In the future, it would be interesting to study in more detail especially the factors related to the differences between mental models of users and system structure. The mental model interview method that was used in pre-training meetings proved to provide a lot of information on users' expectations and their understanding of the system. Similar interviews could be arranged after the training with more in-depth questions. This would provide information on how the view of the system has changed and what kind of misconceptions users have after using the system for a certain period.

It could also be researched how the factors related to the user interface affect the use experience of experienced users. Some factors such as continuity of task sequences will most probably have as positive an effect to expert users as novices, but others such as visibility of operations may have a smaller impact to experts than novices.

Different training methods could be compared by assigning novice users to groups that receive different type of training. The learning results could be measured by arranging a performance test before and after the training. The method producing best learning results could be used in forthcoming training sessions. In this study, only one type of training could be observed and information was obtained on factors that affect learnability, but exact information on what the best possible training method would be like could not be obtained.

It would also be interesting to study the effect of doing the learnability improvements suggested in this study. The suggestions for improvement should improve the performance of novices but should not slow down an expert user either. The effect of addressing one of the problems was already observed in this study. In the observation right after the training, users had many problems with finding a connection or a reinforcement to add into the model. The reinforcements and connections were presented in a list that contained their name and sequence number. Users often did not recognize the names as they were used to Finnish terminology for reinforcements and connections. However, during the two months after the training, a new alpha version of the system was launched. The list of reinforcements and connections had been improved by adding an image describing each list item. This practically eliminated the problem of finding suitable reinforcements and connections. Users commented that the new user interface was much better than the old one. It can be expected that some of the suggested improvements will cause a similar reaction if implemented.

## 7.7 Summary of Results on Learnability

In this study, we researched the learnability of a building modeling system with several methods. By analyzing the data that was collected with the empirical methods, we aimed at distinguishing factors that affect learnability of the system and producing suggestions for improving the learnability.

The difference of this study compared to most other learnability studies is that we addressed the effect of the user interface, system structure, and training, whereas most other studies concentrate on only one of these, usually the user interface. We consider that it is necessary to consider all these issues to gain an understanding of the learnability of a system.

Seven learnability factors related to the user interface were found in the study: visibility of operations, feedback, continuity of task sequences, design conventions, information presentation, user assistance, and error prevention. In addition, four factors related to the system structure were found: differences in functionality, differences in interaction styles, concept clarity, and completeness of information. Furthermore, seven factors related to training were found: conceptual information, exercises, instructions for basic interaction, instructions for solving problems, motivational content, coverage of system functionality, and material types.

27 guidelines related to the user interface, 11 guidelines related to the system structure, and 26 guidelines related to training were constructed. They are designed to fix the learnability problems that were noticed in the user observations.

In addition to creating a classification of learnability factors and guidelines for improving learnability, we created detailed suggestions for improving the learnability of the Tekla Structures system. They can be used to produce immediate improvements in the learnability of the system.

The learnability factors and guidelines related to the user interface that we found have elements in common with usability guidelines presented in the literature. They also have some commonalities with classifications of learnability factors presented in the literature, but the issues have been presented differently in different studies. The factors and guidelines concerning the system structure are in line with the theory of mental models, and the factors and guidelines concerning training correspond with learning theories presented in the literature.

The classification of factors affecting learnability and the guidelines for improving learnability are expected to be the most important academic contributions of this study. The factors and guidelines were developed to cover the issues affecting the learnability of a building modeling system, but they can be used as a reference when studying the learnability of other systems as well.

# References


Anderson, J. R. (1980). *Cognitive psychology and its implications.* 519 p. New York, NY, USA: W. H. Freeman and Company. ISBN: 0-7167-2385-9.

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

Borgman, C. L. (1999). The user's mental model of an information retrieval system: and experiment on a prototype online catalog. *International Journal of Human-Computer Studies, 51,* 435-452.

Butler, K. A. (1985). Connecting theory and practice: A case study of achieving usability goals. In CHI'85, *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 85-88). New York, NY, USA: ACM Press. ISBN: 0-89791-149-0.

Capobianco, A. (2003). Experimental evaluation of the effectiveness of expert online help strategies. In J. Jacko & C. Stephanidis (Eds.), *Human-Computer Interaction: Theory and Practice.* Proceedings of the 10th International Conference on Human-Computer Interaction, vol. 1 (pp. 626-631). Mahwah, NJ, USA: Lawrence Erlabaum Associates, Publishers. ISBN: 0-8058-4930-0.

Carroll, J. M. (1997). Reconstructing minimalism. In SIGDOC'97, *Proceedings of the 15th annual international conference on Computer documentation* (pp. 27-34). New York, NY, USA: ACM Press. ISBN: 0-89791-861-4.

Carroll, J. M., & Carrithers, C. (1984). Training wheels in a user interface. *Communications of the ACM,* 27, 800-806. ISSN: 0001-0782.

Chambers, M., & Connor, S. (2001). Tehcnology as an aid to coping with caring: a usability evaluation of a telematics intervention. In V. Pater et al.: *Medinfo 2001.* Amsterdam: IOS Press.

Chandra, S., & Blockley, D. I. (1995). Cognitive and computer models of physical systems. *International Journal of Human-Computer Studies*, *43*, 539-559.

Chin, J. P.; Diehl, V. A., & Norman, L. K. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In CHI'88, *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 213-218). New York, NY, USA: ACM Press. ISBN: 0-201-14237-6.

Dix, A., Finlay. J., Abowd, G., & Beale, R. (1993). *Human-computer interaction.* 570 p. Hertfordshire, UK: Prentice Hall International (UK) Limited. ISBN: 0-13-437211-5.

Dykstra-Erickson, E.; Curbow, D. (1997). The role of user studies in the design of OpenDoc. In DIS'97, *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques* (pp. 111-120). New York, NY, USA: ACM Press. ISBN: 0-89791-863-0.

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

Everingham, M., & Brown, B. (1986). Developing and delivering hands-on training. In SIGUCCS'86, *Proceedings of the 14th annual ACM SIGUCCS conference on User services: setting the direction* (pp. 209-223). New York, NY, USA: ACM Press. ISBN: 0-89791-208-X.

Fein, R. M., Olson, G. M., & Olson, J. S. (1993). A mental model can help with learning to operate a complex device. In CHI'93, *INTERACT '93 and CHI '93 conference companion on Human factors in computer systems* (pp. 157-158). New York, NY, USA: ACM Press. ISBN: 0-89791-574-7.

Frese, M., Brodbeck, F., Heinbokel, T., Mooser, C., Shcleiffenbaum, E., & Thiemann, P. (1991). Errors in training computer skills: on the positive function of errors. *Human-Computer Interaction, 6,* 77-93.

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.

Halasz, F. G., & Moran, T. P. (1983). Mental models and problem solving in using a calculator. In CHI'85, *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 212-216). New York, NY, USA: ACM Press. ISBN: 0-89791-121-0.

Haramundis, K. (2001). Learnability in information design. In SIGDOC'01, *Proceedings of the 19th annual international conference on Computer documentation* (pp. 7-11). New York, NY, USA: ACM Press. ISBN: 1-58113-295-6.

Karn, K. S., Perry, T. J., & Krolczyk, M. J. (1997). Testing for Power Usability: A CHI 97 Workshop. *SIGCHI Bulletin, 29.*

Kellogg, W. A.., & Breen, T. J. (1986). Evaluating user and system models: Applying scaling techniques to problems in human-computer interaction. In CHI'86, *Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface* (pp. 303-308). New York, NY, USA: ACM Press. ISSN: 0736-6906.

Kieran-Greenbush, S. (1991). Reaching the adult learner: Adult learning and computer training. In SIGUCCS'91, *Proceedings of the 19th annual ACM SIGUCCS conference on User services* (pp. 177-182). New York, NY, USA: ACM Press. ISBN: 0-89791-454-6.

Kirakowski, J. (1996). The software usability measurement inventory: background and usage. In P. Jordan, B. Thomas, B. Weerdmeester, & I. L. McClelland, *Usability Evaluation in Industry*. UK: Taylor & Frances. ISBN: 0-748-40460-0.

Koivulahti-Ojala, M. (2001). *Atk-opettajan opas.* 208 p. Jyväskylä, Finland: Gummerus Kirjapaino. ISBN: 952-14-0335-7.

Koli, H., & Silander, P. (2002). *Verkko-oppiminen: Oppimisprosessin suunnittelu ja ohjaus.* 94 p. Hämeenlinna, Finland: Hämeen ammattikorkeakoulu. ISBN: 951-784-093-4.

Kujala, S., Kauppinen, M., Nakari, P., & Rekola, S. (2003). Field studies in practice: Making it happen. In Rauterberg et al. (eds): *Human-Computer Interaction – INTERACT'03* (pp. 359-364). IOS Press. ISBN: 1-58603-363-8.

Lazar, J., & Norcio, A. (2003). Training novice users in developing strategies for responding to errors when browsing the web. *International Journal of Human-Computer Interaction, 15,* 361-377.

Lehtinen, M. (2003). User Need Questionnaire. Internal document. Espoo, Finland: Usability Engineering Team, Tekla Oyj.

Lewis, J. R. (1995). IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use. *International Journal of Human-Computer Interaction, 7,* 57-78. ISSN: 1044-7318.

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.

Lonka, K.., & Lonka, I. (1991). *Aktivoiva opetus. Käsikirja aikuisten ja nuorten opettajille.* 127 p. Helsinki, Finland: Kirjayhtymä. ISBN: 951-26-3529-1.

Marton, F., & Booth, S. (1997). Le*arning and Awareness.* 224 p. Mahwah, NJ, USA: Lawrence Erlbaum Associates, Inc. ISBN: 0-8058-2454-5.

McKita, M. (1988). Online documentation and hypermedia: Designing learnability into the product. In IPCC'88 *Conference record, Professional Communication Conference* (pp. 301-305). IEEE Professional Communication Society.

Nelson, R. R., Whitener, E. M., & Philcox, H. H. (1993). The assessment of end-user training needs. *Communications of the ACM, 38*, 27-39.

Nielsen, J. (1993). *Usability Engineering.* 358 p. San Francisco, CA, USA: Morgan Kaufmann. ISBN 0-12-518405-0.

Olfman, L., & Shayo, C. (1997). The role of training in preparing end users to learn new but similar software packages. In SIGCPR'97, *Proceedings of the conference of the Special interest group on computer personnel research (pp. 210-233).* New York, NY, USA: ACM Press. ISBN: 0-89791-904-1.

Paymans, T. F., Lindenberg, J., & Neerincx, M. (2004). Usability trade-offs for adaptive user interfaces: ease of use and learnability. In IUI'04, *Proceedings of the 9$^{th}$ international conference on Intelligent user interface* (pp. 301-303). New York, NY, USA: ACM Press. ISBN: 1-58113-815-6.

Preece, J. (1994). *Human-Computer Interaction.* 773 p. Wokingham, England: Addison-Wesley Publishing Company. ISBN: 0-201-62769-8.

Rieman, J. (1996). A field study of exploratory learning strategies. *ACM Transactions on Computer-Human Interaction, 3,* 189-218.

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 CHI'94, *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence* (pp. 438-444). New York, NY, USA: ACM Press. ISBN: 0-89791-650-6.

Roberts, T. L., & Moran, T. P. (1983). The evaluation of text editors: Methodology and empirical results. *Communications of the ACM, 26,* 265-283. ISSN: 0001-0782.

RT 10-10577. (1995). *Scope of work in structural design*. Reference document. Finland: Rakennustieto Oy.

Santos, P. J., & Badre, A. N. (1995). Discount learnability evaluation. Graphics, Visualizationa & Usability Center, Georgia Institute of Technology. Retrieved 2.12.2004 from ftp://ftp.gvu.gatech.edu/pub/gvu/tr/1995/95-30.pdf

Sein, M. K., & Bostrom, R. P. (1989). Individual differences and conceptual models in training novice users. *Human-Computer Interaction, 4*, 197-229.

Shayo, C., & Olfman, L. (1993). Is the effectiveness of formal end-user software training a mirage? In SIGCPR'93, *Proceedings of the 1993 conference on Computer personnel research.* (pp. 88-99). New York, NY, USA: ACM Press. ISBN: 0-89791-572-0.

Shayo, C., & Olfman, L. (1998). The role of conceptual models in formal software training. In SIGCPR'98, *Proceedings of the 1998 conference on Computer personnel research* (pp. 242-253). New York, NY, USA: ACM Press. ISBN: 0-89791-959-9.

SFS-EN ISO 9241. (1998). Ergonomic requirements for office work with visual display terminals (VDTs). Part 10: Dialogue principles, and Part 11: Guidance on usability. Helsinki, Finland: Suomen Standardoimisliitto. 43 p.

Shneiderman, B. (1998). *Designing the User Interface.* USA: Addison-Wesley Publishing Company.

Sinkkonen, I. (2000). *Things that facilitate learning in products.* Licenciate's thesis, Helsinki University of Technology, Department of Computer Science, Espoo, Finland. 138 p.

Sinkkonen, I., Kuoppala, H., Parkkinen, J., & Vastamäki, R. (2002). *Käytettävyyden psykologia.* 343 p. Helsinki, Finland: Edita Publishing Oy. ISBN: 951-826-574-7.

Tynjälä, P. (1999). *Oppiminen tiedon rakentamisena.* 214 p. Helsinki, Finland: Kustannusosakeyhtiö Tammi. ISBN: 951-26-4419-3.

Van der Veer, G. C., & Bamossy, G. (1990). Operating systems in education – mental models in relation to user interface. In A. Finkelstein, M. J. Tauber, & R. Traunmuller (Eds.), *Human factors in analysis and design of information systems* (pp. 223-241). Amsterdam, The Netherlands: Elsevier Science Publishers B.V. ISBN: 0-444-88285-5.

Van Welie, M., van der Veer, G. C., & Eliens, A. (1999). Breaking down usability. In *Proceedings of the Interact 1999 conference (pp. 613-620).*

Vanderlinden, G., Cocklin, T., & McKita, M. (1988). Designing tutorials that help users learn through exploration. In IPCC'88 *Conference record, Professional Communication Conference* (pp. 295-299). IEEE Professional Communication Society.

Vosniadou, S. (1996). Towards a revised cognitive psychology for new advances in learning and instruction. *Learning and Instruction, 6,* 95-109.

Whiteside, J., Jones, S., Levy, P., & Wixon, D. (1985). User performance with command, menu, and iconic interfaces. In CHI'85, *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 185-191). New York, NY, USA: ACM Press. ISBN: 0-89791-149-0.

Wright, P. (1988). Issues of content and presentation in document design. In M. Helander (Ed.), *Handbook of human-computer interaction* (pp. 629-652). Amsterdam, The Netherlands: Elsevier Science Publishers B. V. ISBN 0-444-88673-7.

# Appendices

## Appendix A
## Consent for research

Name of the company: ________________________________________

Name of the participant: ________________________________________

**Research conditions:**

The purpose of the research is to collect information for product development needs. The aim is to make the software application to appear familiar, logical, and easy to use even when it is used for the first time. The results of the research will be presented in a diploma thesis. The subject of the thesis is the learnability of the Tekla Structures system.

Interviews and other research activities will be audio recorded, but the results will be presented in literal form. The results will be presented as a summary of several participants and personal data concerning participants is not included. Information will be treated confidentially and they will not be given to the employee of the participants or other outsiders.

I want to participate in the research ☐

**Use of the audio recordings and photographs:**

The audio recordings or photographs that describe the course of the research may be useful for others as well, if you give your consent for it.

I give my permission for presenting the audio recordings anonymously in situations that are related to product development. ☐

I give my permission for presenting the photographs in situations that are related to product development. ☐

I give my permission for using the photographs in the diploma thesis report. ☐

__________________________, ___/___2004

Signature:_______________________________________________

# Appendix B
## Individual interview: questions

| |
|---|
| **Background information** |
| What is your profession / job description [in company X]? |
| What kind of job did you do before that? |
| What is your education? |
| How long have you used the Tekla Structures system? |
| **Current job description** |
| What activities belong to your job? |
| How has the process that determines your tasks been defined? |
| What initial information you have when you begin the modeling process? |
| What information / drawings / reports you give forward? |
| What phases does the structural modeling process with Tekla Structures include? |
| What kinds of exceptions occur in the phases you described? |
| What kind of changes need to be done in the model during the design process? |
| Are there some phases that need to be finished before starting anything else? |
| What is the most demanding phase of the work? |
| Which issues are important in order the make the modeling process successful? |
| What kinds of problems have occurred in the modeling process? |
| **Advantages and disadvantages of Tekla Structures** |
| How does the terminology used in the system and the real world differ? |
| What are difficult and time-consuming features in the system? |
| What are the good sides of the system? |
| How do you think the system could be improved? |
| What other tools do you use, in addition to Tekla Structures? |
| Have you faced problems with the interoperability of the tools? What kinds of problems? |

# Appendix C
# Mental model interview: questions

The most important questions have been marked with a star and they were emphasized in the mental model interviews.

**General about the interface**
* Which icons seem familiar to you? What do you think the other icons represent?
* What do you expect the software to be able to do, in addition to modeling columns, beams and connections?
* What do you expect to be the biggest differences compared to the software you used before?
If you used 2D modeling software before, how does 3D modeling change the way you work?

**1. Creating views**
* What do you expect the items that you see to be?
Please explain in your own words, what you expect a 'view' to be in the context of this software?
How do you expect the coordinate system to function?

**2. Creating grids**
How would you change the grid dimensions?

**3. Creating concrete or steel parts**
* How would you start creating columns and beams?
- The user is shown how to create a column, or he can do it himself.

* How do you think you can copy and mirror elements?

**4. Modifying concrete or steel parts**
* How would you change the properties (e.g. profile) of a column?
- The user is shown how to change the properties, or he can do it himself.

What do you think about the properties dialog box?

**5. Creating reinforcements**
* How do you think you can create reinforcements?
- If time permits, the user is shown how to create reinforcement for the pad footing, or he can do it himself.

What modifications would be needed in a real use situation to the predefined reinforcements?

**6. Creating connections**
* How do you think you can create connections?
- If time permits, the user is shown how to create another column, a beam and a connection, or he can do it himself.

**7. Saving components**
* If you create something, for example a reinforcement, that you would like to use in other models, how would you make it available in them?

**8. Creating numbering**
What do you think are the preconditions for being able to create drawings?
How do you think you can number the parts?

**9. Creating drawings**
* How do you think you can create 2D drawings from the model?

**10. Updating drawings**
* How do you expect changes in the model to affect the drawings?
* What do you think you need to do to drawings after changing the model, in order for them to stay up-to-date?

**11. Modifying drawings**
How do you think you can modify a drawing, e.g. add part marks?

**12. Creating reports**
How do you think you can create reports?

**13. Exporting / importing data**
What kind of collaboration with other software do you expect this software to support?

**14. Specifying model properties**
Where do you expect to specify project properties, e.g. your company's and your name?

**15. Modifying material catalog**
Where do you expect to save material properties, part profiles and other project specific information?

# Appendix D
# Mental model interview: extract from results

Issues about which users had mostly correct assumptions

| Core task | Issue | User comments |
|---|---|---|
| **General about the interface** | Users recognized or could guess the meaning of most icons, such as file manipulation operations (New, Open), move and copy operations, and icons for creating objects. (4 users) | "That one will create some listings." (U1)<br>"These must be connections there." (U1)<br>"Here are some that are familiar from Office programs: Open and Save and such." (U2)<br>"At least Snaps, Grids and dimensions are familiar." (U4)<br>"The pictures tell quite a lot." (U5) |
| | Most users understood that this system is intended for modeling parts whereas the software application they had used before was intended for drawing two-dimensional objects. I.e. they understand the basic difference between the two groups of software applications that will make their functionality different in nature. (4 users) | "None of the software I have used before can be called modeling software." (U1)<br>"That is at least my understanding that the parts will be modeled. …And now the model and the drawing are the same thing." (U2)<br>"I think this is such that you can make a whole object at a time and you can then modify it, whereas in AutoCAD you make one line at a time." (U3)<br>"You don't need to draw parts yourself but it makes them as they are dimensioned there." (U5) |
| **3. Creating concrete or steel parts** | Users could easily place parts in the model without any training. Placing parts is one of the most basic operations of the system. (5 users) | "Now it is there!" (U1)<br>"If I would go like in AutoCAD, I would need to find the column and it will ask a point. Pick point, ok…" (U2)<br>"Now there in Parts, there is the Concrete column. … Now it seems that I need to select a point where I want to put it. … Ok, there. Now there is a steel column." (U3)<br>(U4 and U5 made it so fast that they did not even comment anything while they were doing it.) |
| **10. Updating drawings** | The task sequence for updating drawings was intuitive. (3 users) | "I expect that after making changes there is somewhere a button Update drawings." (U1)<br>"Probably with a separate command that updates the drawing. I would expect that. I think that is an ok way to do it. I would expect that there will be some message that model has been updated." (U2)<br>"I need to go here and press update or something corresponding, and it will update the drawings." (U3) |
| **13. Exporting / importing data** | Users' expectations for export and import features were consistent with the existing features. The features will probably fulfill their data exchange needs. (3 users) | "I need to for example import AutoCAD models. For example use architect images. That is rather important for us. We can use DWG format." (U2)<br>"But if it is possible to take a plan view in this system and put it one upon the other with a plan view in AutoCAD. So that you can compare if lines match, if they are really on top of each other." (U3)<br>"And I think you can bring DWG images here." (U4) |

Issues about which users had a lot of incorrect assumptions

| Core task | Issue | User comments |
|---|---|---|
| **General about the interface** | The meaning of buttons OK, Apply, Modify, and Cancel was unclear to users. (4 users) | "Usually it is so that with OK, you accept the changes. This may feel a bit strange in the beginning." (U2)<br>"At least Microsoft has used it so that when you press OK, changes take effect right away." (U4)<br>"I think it is good, that always when you press Modify, when you press Cancel, it goes backwards. So that you can preview what is being made. And if it is not suitable, just cancel, and it will be returned to what it was. I expect it works like that." (U4)<br>"I always press all of them, Modify, Apply, and OK." (U5)<br>"If I press this (OK), does it create it or do I need to save first?" (U3, Create grid views dialog) |
| | Users expected that the system would provide more instructions and feedback than it actually does. They also had problems interpreting the existing instructional texts. (3 users) | "What it is asking here?" (U2)<br>"Doesn't it tell in what state the command is?" (U4)<br>"What did it do?" (U4)<br>"Did it do something?" (U5) |
| **7. Saving components** | Users expected that storing connections for later use was simpler than it actually is. They used the term 'Save' whereas the current term is 'Define custom component'. (3 users) | "Then, I would imagine I can just simply save the connection." (U1)<br>"There is probably some copy command. Or save." (U2)<br>"Is there a save command?" (U4) |
| **8. Creating numbering** | Users did not understand the reason why all parts need to be numbered before creating drawings. The concept of numbering was not familiar to them. (4 users) | "Now it does not create drawings because I have not done the numbering but..." (U1)<br>"In AutoCAD, there is no numbering." (U3)<br>"What does numbering actually mean?" (U4)<br>"I have never heard about numbering in AutoCAD. So you need to name each part?" (U4)<br>"Numbering! What did I have to do? I had to do something! In AutoCAD you don't need..." (U5) |
| **12. Creating reports** | Users were confused with the number of report types available. The titles of the report types did not clarify the scope of the report. Users would need only a few report types. (4 users) | "I need at least element lists. Not many others." (U2)<br>"I don't know if these are some existing ones." (U3)<br>"Is it this one (report type) then?" (U4)<br>"There are so many of these!" (U5) |

# Appendix E
## Training observation form

| Duration | | | | Topic | Teaching method | Related concepts that were explained during training | Related concepts that were not explained during training |
|---|---|---|---|---|---|---|---|
| 1st day | 2nd day | 3rd day | Total | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

| Chapter in the electronic training material that is handled in the training | References to help material | Questions from participants | Behavior of participants |
|---|---|---|---|
| | | | |
| | | | |
| | | | |

## Appendix F
## Training observation: extract from user comments

| Core task | User | Question or comment |
|---|---|---|
| **1. Creating views** | U6 | "I could create the grid views but they disappeared!" |
| | U1 | "Why I don't see some of the beams in these views?" (Had small view depth in plane views.) |
| | U4 | "Everything else disappeared as I tried to make the hollow core slabs invisible." (Did not choose the Not check box on the view filter dialog box.) |
| | U1 | "It would be handy if in the view properties dialog box, you could take the view filter into user or from use with one selection. For example, you could choose a standard filter or a user-defined filter." |
| | U2 | "Why can't I see the hollow core slabs?" (The view filter was on.) |
| **2. Creating grids** | U3 | "It did not change anything!" (Had done changes in grid dimensions but did not have the grid chosen and therefore the Modify button did not change anything.) |
| | U5 | "How was it? I'm still in the grid thing." |
| **3. Creating concrete or steel parts** | U5 | "How can I pick points?" |
| | U6 | "How do I find the right profile?" |
| | U1 | "What does this class refer to?" |
| | U5 | "Do I always need to select the starting point for a beam by pressing Ctrl and selecting the point?" |
| | U5 | "Can you explain the position settings (On plane, Rotation, At depth)?" |
| | U5 | "On which plane are these slabs? Some of them are very low, others are in the sky. I have a lot of them! Oh, I copied ten pieces of them! I wondered why it loaded them for so long!" |
| | U2 | "Why my slab is triangular? One corner is missing." |
| **4. Modifying concrete or steel parts** | U6 | "How can you grab the chamfers?" |
| | U5 | "Where is the part cut command?" |
| | U1 | "The measurement is probably not accurate if you change the polygon shape like that?" |
| | U5 | "How can I move the end of a part?" |
| | U5 | "Does the point need to be yellow (to move the starting / ending point)?" |
| **5. Creating reinforcements** | U3 | "How can I see what the macro is like?" |
| | U3 | "Now the reinforcements are overlapping. How can I modify them to avoid overlapping?" |
| | U5 | "I reinforced the silos… and the bars are outside of it!!" |
| | U5 | "What points should I pick in the model? Which direction?" |
| | U4 | "What is the difference between From plane and On plane?" |
| | U5 | "What did it do? Pick object?" |
| **6. Creating connections** | U1 | "If I use the system for the first time, how do I know which connection I should use?" |
| | U3 | "What did I need to enter there (in the connection properties dialog box)?" |
| | U6 | "Why are all the connections not created?" (Was trying to create AutoConnections.) |
| | U1 | "What are the blue rectangles around the connection?" |
| | U3 | "Does number 90 something mean that it is on the ninth page in the connection toolbar?" |

| Core task | User | Question or comment |
|---|---|---|
| **7. Saving components** | U5 | "I cannot do the whole detail! How can I grab the hole?" |
| | U2 | "Can I move the hole somehow?" |
| | U5 | "Where is this distance measured from? From the edge or the middle line?" |
| | U1 | "But if you modify an existing custom component, it will change all the existing parts!" |
| **8. Creating numbering** | U5 | "I don't have the part field in my dialog box!" (Was looking at the concrete column dialog, did not hear that concrete and steel parts have different settings.) |
| | U3 | "How can I modify the numbering settings?" |
| | U1 | "Numbering is still unclear to me." |
| **9. Creating drawings** | U4 | "Where did you get that?" (Go to Drawing menu and select general arrangement drawing.) |
| | U5 | "How can you go to the menu?" (Was told to open the properties menu.) I don't have it there! (Was told to close the drawing first.) |
| **10. Updating drawings** | U4 | "What did n in the drawing list mean?" |
| | U3 | "What is the difference of Save and freeze with the Save command?" |
| **11. Modifying drawings** | U1 | "I cannot modify the part!" (Was told that the drawing editor needs to be closed first.) |
| | U5 | "What is this part mark?" |
| | U5 | "I cannot modify the caption fields?" |
| | U5 | "What is the difference between buttons Update and Apply?" |
| | U5 | "I already closed it with Cancel!" |
| | U3 | "If it does not do what you want it to do, it is difficult! And all these windows… It depends on so many things." |
| | U1 | "It would be easy just to click different things in the image and define the properties for them." |
| | U4 | "How can I change the size of the paper?" |
| **12. Creating reports** | U3 | "Can you only choose a part from the ID list?" |
| **13. Exporting / importing data to other applications** | | Was not covered in training. |
| **14. Specifying model properties** | | Was not covered in training. |
| **15. Modifying catalogs** | U2 | "How do you open it?" (Was told to click Edit.) |
| | U6 | "How do you find edit then?" (Was told to right-click on an item.) |

## Appendix G
## Learnability test: relationship between core tasks and test tasks

1. Creating views: task A
2. Creating grids: task L
3. Creating concrete or steel parts (columns, beams, slabs etc.): tasks B, M, O
4. Modifying concrete or steel parts: tasks G, N, S
5. Creating reinforcements: task C
6. Creating connections: task P
7. Saving components (reinforcements, connections etc.): task K
8. Creating numbering: task D
9. Creating drawings: task E
10. Updating drawings: task H
11. Modifying drawings: task I
12. Creating reports: task F
13. Exporting/importing data to other applications: task J
14. Specifying model properties: task Q
15. Modifying catalogs: task R

# Appendix H
# Learnability tests: scenarios

The tasks that users were expected to do are numbered. The scenario that was presented to the user is written in italics.

**A. Create views &**
**B. Create concrete parts for the foundation**
*A customer wants to build a garage with a concrete foundation. The customer has created preliminary drawings for the foundation with an architect and your company's project manager. They bring their sketches to you because your task is to create the model, drawings and reports to order the material and guide the construction workers.*

*First, you need to model the pad footings and concrete columns so that the construction workers can cast the footings. Start creating the model just as you would normally do. Create one instance of a pad footing and column.*

**C. Create a reinforcement for concrete parts**
*Create reinforcements for the concrete parts. You can choose the type of reinforcement yourself.*

**D. Create the numbering &**
**E. Create drawings**
*The construction manager called that he needs some images right away to start planning the formwork operations. Create for him formwork drawings that show the pad footings.*

**F. Create reports**
*Create also a material list that shows both concrete parts so that the material need can be calculated.*

**G. Modify the pad footings**
*You get the results from the soil analysis and decide that the pad footings need to be larger. Enlarge them in the model.*

**H. Update drawings**
*Update the formwork drawings you created earlier so that they show the new pad footing dimensions.*

**I. Modify a drawing**
*Add information on the material of the pad footings to a formwork drawing.*

**J. Export / import data to other applications**
*The customer wants to see the drawings for the footings. Unfortunately, he does not have Tekla Structures but only AutoCAD software available. Save a drawing in a form that can be opened in AutoCAD.*

**K. Save a reinforcement for later use**
*In the future, you will probably have similar projects with similar reinforcements. Save the reinforcements that you have created so that you can use them later in other models.*

**L. Create appropriate grids (if not already created) &**
**M. Create the remaining pad footings and columns and the concrete slabs**
*Now it is time to finish the drawings for the foundation. Model the pad footings, columns and concrete slabs that are shown in the drawings.*

**N. Modify the model**
*The architect calls you and tells that the width of the garage must be changed from 5m to 6m to have enough space for two cars. Change the model accordingly.*

**O. Create concrete parts for the upper part of the model**

*The architect and project manager have now created preliminary drawings for the concrete parts of the garage walls and roof. Add the concrete parts to the model.*

**P. Create a connection**
*Create connections that connect the upper beams to the columns.*

**Q. Specify properties for the model**
*Update the project and designer information so that it corresponds to the conventions in your company. Add e.g. your name.*

*Modify the numbering style so that it corresponds to the one used in your company.*

*Modify concrete column and beam default properties (e.g. material) so that they correspond to the conventions in your company. Save the new properties so that you can use them later when you create parts.*

**R. Modify material catalog**
*There is a new concrete type available and you want to use it in your model. Save the material information so that the material can be used in the model.*

**S. Modify concrete slab properties**
*Change the concrete slab material type to the new material.*

# Appendix I
## Learnability tests: observation form

| Seq. number | Core task | Test task | Duration | Time on tape | Maximum time (min) | User actions |
|---|---|---|---|---|---|---|
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |
| | | | | | | |

| Errors that affect the end result / issues that hinder task completion | Difficulties / non-optimal actions | Instructions given | User comments | Ideas |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

## Appendix J
## Learnability tests: extract from qualitative results

| Core task | Examples of problems | Users experiencing the problem | |
|---|---|---|---|
| | | First observation | Second observation |
| **1. Creating views** | Users often pressed the Create button in the Create grid views dialog before selecting the grid and therefore they got a warning message. | U5 | U4, U5 |
| **2. Creating grids** | Users were not able to enter grid dimensions to the fields on the Grid properties dialog correctly. When users wanted to create three grid lines with the spacing of 5000, they entered "0 3*5000" to the grid properties dialog, which actually produced four grid lines. | U2, U3, U5, U6 | U3, U4, U5 |
| **3. Creating concrete or steel parts** | Users did not know how to define the snap settings that they needed. They needed to snap to all points or only grid lines but did not know how to do it. | U1, U2, U3, U5 | U4 |
| **4. Modifying concrete or steel parts** | Users did not know how to move parts. They did not realize they can be moved by grabbing the points that are attached to the part. Some users asked if there is a stretch command available. | U1, U2, U4, U5 | U1, U5 |
| | Users sometimes selected several parts, double-clicked one of them, and thought that changes they made in the dialog would affect all the parts that they had selected in the beginning. However, if several parts are selected and after that one of them is double-clicked, the selection is applied only to the part that was double-clicked. | U1, U2 | U3, U4, U5 |
| **5. Creating reinforcements** | Users had problems finding suitable reinforcements and connections. Users were not familiar with the names of the reinforcements and connections and therefore it was difficult to select a reinforcement from the list of names. Users sometimes entered the search term reinforcement, but as not all reinforcements contain the word in their name, a suitable reinforcement was not found. | U1, U2, U3, U4, U5, U6 | U2, U4 |
| **6. Creating connections** | Was not covered in the learnability test. | | |
| **7. Saving components** | Users could not choose a correct type for custom components (part, detail, connection, or seam). | U1, U5, U6 | U1, U2, U4 |
| | Users had difficulties with entering points for a custom component. Users seemed to give the points randomly. | U1, U5 | U1, U2, U4 |
| **8. Creating numbering** | Users often forgot to run numbering before creating drawings. The warning message was shown. | U2, U3, U4, U5 | U1, U2, U3 |
| | Users were confused about Tools and Setup menus both having a numbering item. They often went to Setup menu when they wanted to do numbering. Some users even thought that they had done numbering even though actually they had only opened and closed the numbering setup dialog. | U2, U3 | U3, U5 |
| **9. Creating drawings** | Users were wondering if anything had happened when they had created drawings and succeeded in it. There is not enough feedback about drawings being created. | U6 | U1, U3, U5 |
| **10. Updating drawings** | The meaning of Freeze is not intuitive but users asked what its meaning is. | U2, U3, U5, U6 | |
| **11. Modifying drawings** | Users did not understand the difference between the drawing state and the modeling state. They were wondering why they could not see all the views, use the menu commands, or delete | U1, U4, U6 | U3 |

| | | | |
|---|---|---|---|
| | drawings in the drawing list. The reason was that they were in the drawing list. They also had difficulties realizing how to exit the drawing state. | | |
| **12. Creating reports** | After selecting the report type, users pressed the Show button. However, pressing the Show button opens an existing report if it is available, but if the user wants to create a new report, he would need to press Create from all or Create from selected. | U4, U6 | U3 |
| **13. Exporting / importing data to other applications** | When users wanted to export the model to AutoCAD, they often chose the option Export > CAD drawing as the name suggests that it will create drawings that are suitable for that. However, they would probably need to create DXF drawings in most situations, which is a separate menu item. | U2, U4, U5 | U1, U3, U4 |
| **14. Specifying model properties** | Users did not remember where to enter the project properties. | U3 | U4 |
| **15. Modifying catalogs** | Users did not remember how to add or edit materials. They did not remember that material information is stored in catalogs. | U3, U4 | U2, U3, U4 |

## Appendix K
## Learnability tests: Quantitative results

| |
|---|
| **Light gray:** percentage of users 70 – 100% or average time 0 – 3 min |
| **Dark grey:** percentage of users 30 – 70% or average time 3 – 7 min |
| **Black:** percentage of users 0 – 30% or average time 7 – 10 min |

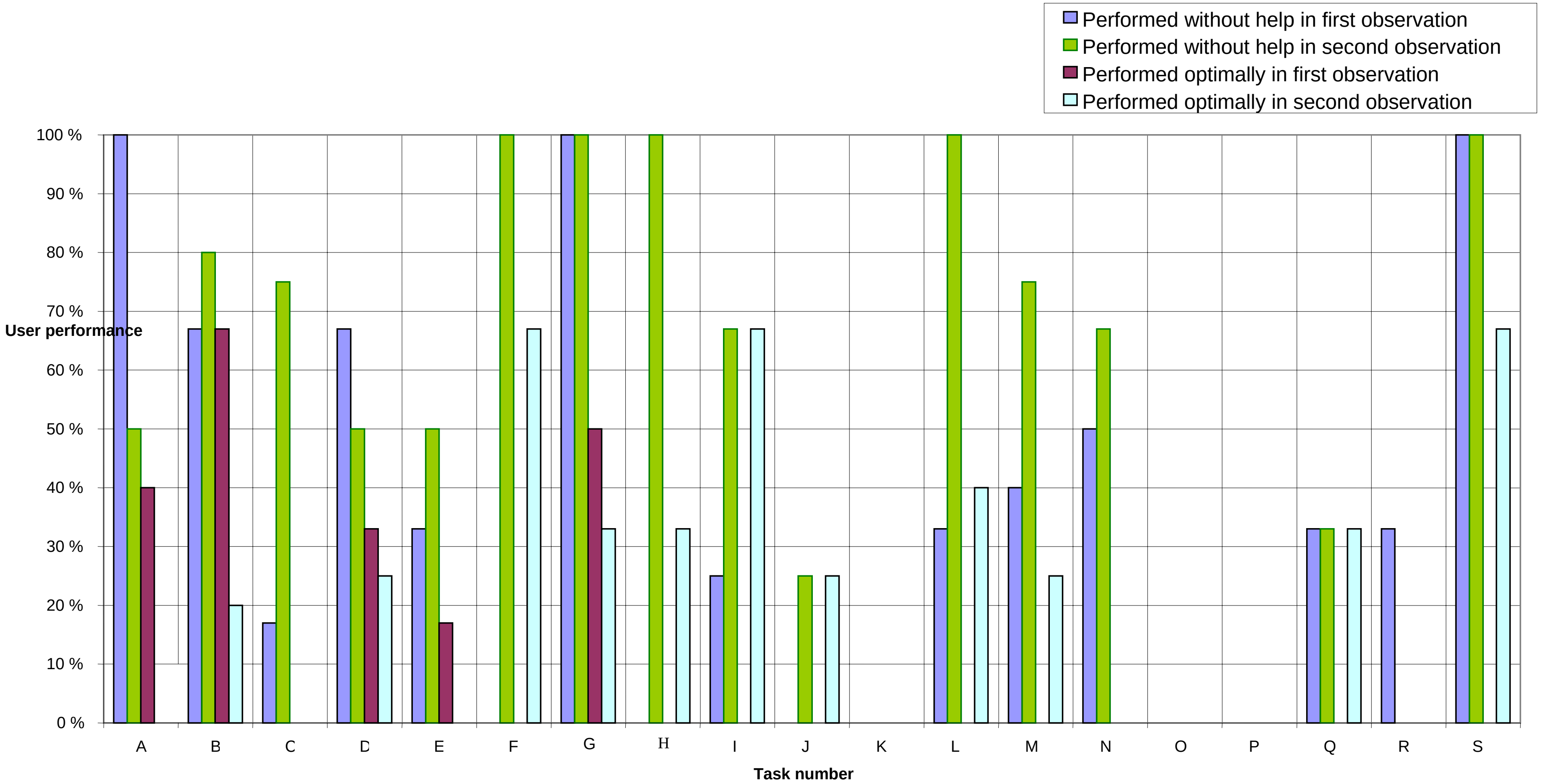
Results of the learnability test arranged immediately after the training.

| | **Task** | **Number of users that did the task** | **Percentage of users that could perform the task without instructions** | **Percentage of users that could perform the task optimally** | **Average time for performing the task** |
|---|---|---|---|---|---|
| **A** | **Create views** | 5 | 100% | 40% | 1:04 |
| **B** | **Create concrete parts for the foundation** | 6 | 67% | 67% | 5:52 |
| **C** | **Create a reinforcement** | 6 | 17% | 0% | 7:44 |
| **D** | **Create the numbering** | 6 | 67% | 33% | 1:02 |
| **E** | **Create drawings** | 6 | 33% | 17% | 5:59 |
| **F** | **Create reports** | 4 | 0% | 0% | 3:31 |
| **G** | **Modify the pad footings** | 4 | 100% | 50% | 2:11 |
| **H** | **Update drawings** | 3 | 0% | 0% | 7:01 |
| **I** | **Modify a drawing** | 4 | 25% | 0% | 6:41 |
| **J** | **Export / import data** | 3 | 0% | 0% | 4:52 |
| **K** | **Save reinforcement for later use** | 3 | 0% | 0% | 9:15 |
| **L** | **Create appropriate grids** | 6 | 33% | 0% | 6:03 |
| **M** | **Create the remaining pad footings and columns and the concrete slabs** | 5 | 40% | 0% | 6:09 |
| **N** | **Modify the model** | 2 | 50% | 0% | 6:24 |
| **O** | **Create concrete parts for the upper part of the model** | 0 | | | |
| **P** | **Create a connection** | 0 | | | |
| **Q** | **Specify properties for the model** | 3 | 33% | 0% | 3:43 |
| **R** | **Modify material catalog** | 3 | 33% | 0% | 3:17 |
| **S** | **Modify concrete slab properties** | 1 | 100% | 0% | 5:16 |
| | | | | | |
| | **Average over the tasks** | | 43% | 12% | 5:03 |

Results of the learnability test arranged two months later.

| | **Task** | **Number of users that did the task** | **Percentage of users that could perform the task without instructions** | **Percentage of users that could perform the task optimally** | **Average time for performing the task** |
|---|---|---|---|---|---|
| **A** | **Create views** | 4 | 50% | 0% | 2:46 |
| **B** | **Create concrete parts for the foundation** | 5 | 80% | 20% | 4:56 |
| **C** | **Create a reinforcement** | 4 | 75% | 0% | 5:37 |
| **D** | **Create the numbering** | 4 | 50% | 25% | 1:10 |
| **E** | **Create drawings** | 4 | 50% | 0% | 6:37 |
| **F** | **Create reports** | 3 | 100% | 67% | 2:09 |
| **G** | **Modify the pad footings** | 3 | 100% | 33% | 4:52 |
| **H** | **Update drawings** | 3 | 100% | 33% | 3:52 |
| **I** | **Modify a drawing** | 3 | 67% | 67% | 6:49 |
| **J** | **Export / import data** | 4 | 25% | 25% | 5:44 |
| **K** | **Save reinforcement for later use** | 3 | 0% | 0% | 8:49 |
| **L** | **Create appropriate grids** | 5 | 100% | 40% | 3:37 |
| **M** | **Create the remaining pad footings and columns and the concrete slabs** | 4 | 75% | 25% | 7:28 |
| **N** | **Modify the model** | 3 | 67% | 0% | 7:02 |
| **O** | **Create concrete parts for the upper part of the model** | 0 | | | |
| **P** | **Create a connection** | 0 | | | |
| **Q** | **Specify properties for the model** | 3 | 33% | 33% | 6:50 |
| **R** | **Modify material catalog** | 3 | 0% | 0% | 4:44 |
| **S** | **Modify concrete slab properties** | 3 | 100% | 67% | 1:24 |
| | | | | | |
| | **Average over the tasks** | | 63% | 26% | 4:58 |

# Appendix L
## Learnability tests: graph of quantitative results




The percentages of users doing each task optimally or without instructions.

# Appendix M
# Subjective satisfaction questionnaire: questionnaire form

EVALUATING THE TEKLA STRUCTURES SYSTEM

**General**

| | | | |
|---|---|---|---|
| Learning to operate the system is | difficult | ❑❑❑❑❑ | easy |
| Exploring new features by trial and error is | difficult | ❑❑❑❑❑ | easy |
| The software corresponds to my expectations | not at all | ❑❑❑❑❑ | very much |
| Using the software is | frustrating | ❑❑❑❑❑ | pleasant |

**Learnability of the user interface**

| | | | |
|---|---|---|---|
| Remembering names and use of commands is | difficult | ❑❑❑❑❑ | easy |
| Tasks can be performed in a straight-forward manner | never | ❑❑❑❑❑ | always |
| Different functions have similar operating logic | never | ❑❑❑❑❑ | always |
| The amount of guidance that the software offers is | insufficient | ❑❑❑❑❑ | sufficient |
| Understanding the structure of the program is | difficult | ❑❑❑❑❑ | easy |

**Material and training**

| | | | | |
|---|---|---|---|---|
| *Example row:* | *useless* | *❑❑❑❑❑* | *useful* | *❑ I have not used* |
| Help pages are | | ❑❑❑❑❑ | | ❑ |
| Printed training material is | | ❑❑❑❑❑ | | ❑ |
| Training material CD is | | ❑❑❑❑❑ | | ❑ |
| Context-sensitive help (opens with F1 button) is | | ❑❑❑❑❑ | | ❑ |
| Instructions on the computer screen are | | ❑❑❑❑❑ | | ❑ |
| Training session is | | ❑❑❑❑❑ | | ❑ |

**Function specific questions**

| | | | | |
|---|---|---|---|---|
| *Example row:* | *difficult* | ❑❑❑❑❑ | *easy* | ❑ *I cannot do it* |
| Creating views is | | ❑❑❑❑❑ | | ❑ |
| Creating grids is | | ❑❑❑❑❑ | | ❑ |
| Creating concrete or steel parts is | | ❑❑❑❑❑ | | ❑ |
| Modifying concrete or steel parts is | | ❑❑❑❑❑ | | ❑ |
| Creating reinforcements is | | ❑❑❑❑❑ | | ❑ |
| Creating connections is | | ❑❑❑❑❑ | | ❑ |
| Saving reinforcements and connections is | | ❑❑❑❑❑ | | ❑ |
| Creating numbering is | | ❑❑❑❑❑ | | ❑ |
| Creating drawings is | | ❑❑❑❑❑ | | ❑ |
| Updating drawings | | ❑❑❑❑❑ | | ❑ |
| Modifying drawings is | | ❑❑❑❑❑ | | ❑ |
| Creating reports is | | ❑❑❑❑❑ | | ❑ |
| Exporting or importing data to other applications is | | ❑❑❑❑❑ | | ❑ |
| Specifying model properties (e.g. designer information) is | | ❑❑❑❑❑ | | ❑ |
| Modifying material and profile catalogs is | | ❑❑❑❑❑ | | ❑ |

# Appendix N
# Subjective satisfaction questionnaire: results

Answers are marked with color coding. The color codes are:

| |
|---|
| **Light grey:** Excellent (4.1-5.0) or none of the users gave a zero score |
| **Dark grey:** Above the average but not excellent (3.1-4.0) or one of the users gave a zero score |
| **Black:** Same as or below the average (3.0) or two or more users gave a zero score |

**Table 16.** Results of the subjective satisfaction questionnaire.

| Question | Average grade | | Number of answers "I cannot do it" or "I have not used" | |
|---|---|---|---|---|
| | After training | 2 months later | After training | 2 months later |
| **General** | | | | |
| Learning to operate the system is | 3.2 | 3.6 | - | - |
| Exploring new features by trial and error is | 3.5 | 3.4 | - | - |
| The system corresponds to my expectations | 4.2 | 4.2 | - | - |
| Using the system is | 4 | 3.6 | - | - |
| **Learnability of the user interface** | | | | |
| Remembering names and use of commands is | 2.8 | 3 | - | - |
| Tasks can be performed in a straight-forward manner | 3.5 | 2.8 | - | - |
| Different functions have similar operating logic | 3.7 | 3 | - | - |
| The amount of guidance that the system offers is | 4 | 3.4 | - | - |
| Understanding the structure of the program is | 3.3 | 3.4 | - | - |
| **Material and training** | | | | |
| Help pages are | 3.8 | 2.6 | 1 | 2 |
| Printed training material is | 4.5 | 3.4 | - | 1 |
| Training material CD is | 2.8 | 1.2 | 2 | 3 |
| Context-sensitive help (opens with F1 button) is | 3 | 1 | 2 | 4 |
| Instructions on the computer screen are | 4.7 | 3.2 | - | 1 |
| Training session is | 4.7 | 4.6 | - | - |
| **Function specific questions** | | | | |
| Creating views is | 3.8 | 4.4 | 1 | - |
| Creating grids is | 4.7 | 4.4 | - | - |
| Creating concrete or steel parts is | 4.7 | 4.4 | - | - |
| Modifying concrete or steel parts is | 3.8 | 3.6 | - | - |
| Creating reinforcements is | 3.7 | 3.2 | - | - |
| Creating connections is | 3.3 | 3.8 | - | - |
| Saving reinforcements and connections is | 3.3 | 2.8 | - | 1 |
| Creating numbering is | 4 | 4 | - | - |
| Creating drawings is | 3.5 | 3 | - | - |
| Updating drawings | 2.8 | 3 | - | - |
| Modifying drawings is | 2.7 | 2.4 | - | - |
| Creating reports is | 3.2 | 2.4 | - | 2 |
| Exporting or importing data to other applications is | 1.3 | 2 | 4 | 2 |
| Specifying model properties (e.g. designer information) is | 2.5 | 2.2 | 1 | 2 |
| Modifying material and profile catalogs is | 2.5 | 3 | 2 | 1 |

## Appendix O
## Extract from the table of learnability phenomena

| Issue / Core task | Problem | Noted in mental model interview | Noted in training | Noted in training material assessment | Noted in usability test | Noted in usability test in October | Noted in the questionnaire | Nb. of users experiencing the problems | UI learnability factor | System functionality factor | Training factor |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **8. Creating numbering** | Users said that the idea of numbering and how to change the settings were unclear to them. Most training participants could not follow the instructions for defining the numbering settings. | | TO task 8 U1,3 own model U6 | | LTa U1,3,4, 5 | | | 5 | Continuity of task sequences | | Exercises |
| **8. Creating numbering** | Users often forgot to run numbering before creating drawings. The warning message was shown. | MMI CT3,3,5 | TO | | LTa U2,3,4, 5 | LTb U1,2,3 | | 5 | Error prevention | Differences in functionality | Conceptual information |
| **8. Creating numbering** | Users did not know where to check the numbering series settings. There are too many places where numbering settings can be changed. | | TO task 11 U5 | | LTa U1,4,5 | | | 3 | Continuity of task sequences | | Conceptual information |
| **8. Creating numbering** | Users were confused about Tools and Setup menus both having a numbering item. They often went to Setup menu when they wanted to do numbering. Some users even thought that they had done numbering even though actually they had only opened and closed the numbering setup dialog. | | | | LTa U2,3 | LTb U3,5 | | 3 | Continuity of task sequences | | Conceptual information |
| **8. Creating numbering** | One user said that controlling numbering is a bit difficult. When the user restarts the computer, the settings are always reset. If he doesn't check the settings every time, there is a danger of the numbering changing. He usually changes the settings for modified numbering so that it keeps the old numbers if possible. He also checks the second and third check-box on the numbering setup dialog. | | | | | LTb U2 | | 1 | Suggestion | | |

109

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **8. Creating numbering** | It was confusing for some users that steel and concrete parts had different ways to set numbering prefixes and starting numbers. | | TO task 8 U5 | | | | 1 | Information presentation | | |
| **9. Creating drawings** | None of the users used the wizard command for drawing generation as the command name is not descriptive, it is located in the wrong menu (File) and it was not taught in the training. A better name would be Drawing wizard or the command should be moved to the Drawing menu. | MMI 1, gt 1 | TO task 10 U1,2, 3,4,5, 6 | LTa U1,2, 3,4,5, 6 | | | 6 | Visibility of operations | | |
| **9. Creating drawings** | Many users were wondering if anything had happened when they had created drawings and succeeded in it. There is not enough feedback about the drawing creation. To overcome this, when a drawing is created, the drawing list could be opened automatically. | MMI 3 | | LTa U6 | LTb U1,3, 5 | | 4 | Feedback | Completeness of information | |
| **9. Creating drawings** | Users did not know after the drawing how to create a simple drawing, as there were so many confusing things about modifying drawings presented in the training. | | | LTa U1,2 | LTb U1,5 | | 3 | Continuity of task sequences | | Conceptual information |
| **9. Creating drawings** | Users were often not happy with the drawing that was created. For example, when an assembly drawing was created of a pad footing, it did not show the pad footing from above whereas a cast unit drawing showed it. Or the pad footing and column were not included in the same image as the user wished. Or a table of bending shapes for reinforcements was included even though it would not have been needed. | | | | LTb U1,3 | | 2 | Design conventions | | |
| **9. Creating drawings** | There should be drawing templates available and it should be easy to modify them. | MMI 1,3 | | | | | 2 | Design conventions | | |
| **9. Creating drawings** | It is not clear which drawing is related to which part. There could be a preview option and some more information on the drawing. | | | LTa U1,3 | LTb U1 | | 2 | Feedback | Completeness of information | |
| **9. Creating drawings** | Users sometimes had not selected any parts and wondered why the cast unit drawing option has been grayed out in the menu. | | | | LTb U5 | | 1 | Visibility of operations | | Instructions for solving problems |

110

# Appendix P
# Extract from the list of suggestions for improving learnability

The tables contain learnability problems, improvement suggestions, references to guidelines that address each problem, and the number of users that experienced each problem (if applicable). There is one table for each learnability factor.

**User interface**

Visibility of operations

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| After hiding some objects in the model view, users did not know how to show them again as there are several dialogs and commands that can be used to determine which objects are shown. | Controls for object visibility should be collected on one dialog, | 1 | 3 |
| There is a drawing wizard available that eases the drawing creation process. However, the wizard was not used because it is located in File menu and not in the Drawing menu like other drawing commands. | The drawing wizard command should be moved from the File menu to the Drawing menu. | 1 | 6 |
| For some hierarchical lists, elements can be added or edited by right-clicking on an existing list item on either the same or the upper level and selecting Add or Edit from the pop-up menu that appears. Many users failed to notice this possibility as there is no visual indication of it. | An alternative for right-clicking on list items should be provided. | 2 | 5 |
| The buttons that have no effect or cannot be used in a certain state are not greyed out. Greying out buttons would prevent the user from doing operations that have no effect and reduce the number of available options which makes the interaction sequence easier. | Buttons (e.g. the basic buttons OK, Apply, Modify, and Get) should be grayed out when their use has no effect or they cannot be used. | 3 | 1 |
| Users tried to directly modify the text of part marks. The correct method would have been to double-click the part mark to open a dialog on which the modifications can be made. | It should be possible to modify part mark text directly, not through a dialog box. | 4 | 4 |
| Almost all users made mistakes when they were entering the grid dimensions. They would have had to enter the number of gaps between the grid lines but they entered the number of grid lines. | The fields for entering grid dimensions should be replaced with a graphical representation and an instructional text. | 5 | 6 |
| Users did not notice the difference between drawing and modeling states but tried to use commands that are only available in the other state. | The change from the modeling state to the drawing state should be indicated more clearly by e.g. changing the color of the cursor. | 6 | 5 |

Feedback

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| There are select filters available that make a selection to address only the predefined object types. Users often tried to select an object that was not selectable because of | If the user has taken a Select filter into use but repeatedly tries to perform operations that are disabled because of the Select filter, he should | 1 | 3 |

| | | | |
|---|---|---|---|
| the select filter settings. | be reminded about the filter. | | |
| The user may have hidden some parts earlier but does not remember it any more. There is not any indication of parts being hidden from view. | An icon for showing all hidden model objects should be provided in the toolbars. Also, the view depth that defines which items are visible should be graphically indicated in the model. | 2 | 4 |
| The user can create drawings by selecting a drawing type in the menu. After doing this, a text "X drawings created" appears in the status bar but no other feedback is given about a successful drawing creation process. Users often thought that nothing had happened as they did not see any result on the screen | The list showing all the drawings should be opened automatically after the user has started the command for creating drawings. | 3 | 4 |

Continuity of task sequences

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| Before running the numbering command, users often want to see the numbering settings because incorrect settings may cause extra problems and override work done by other designers. The numbering settings and the command for running numbering are located in different menus. Users had problems remembering which menu item was for adjusting settings and which one for running numbering. | The numbering setup dialog and the numbering tool should be combined. | 1 | 3 |
| If the user wants to define settings for certain object groups in a drawing, he must complete a complex task sequence to define drawing classifier settings. There were so many phases in the process that users could not follow instructions in the training and were not able to complete the sequence later. | The task sequence for using the drawing classifier should be redesigned. | 1 | 5 |
| Users often forgot to run numbering before creating drawings. The message that requested the user to perform numbering first interrupted the drawing creation process. | If the user has forgotten to run numbering before creating drawings, it should be possible to assign numbers directly from the message dialog that reminds about numbering. | 2 | 5 |

Design conventions

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| The meaning of the buttons Modify, Apply, OK, and Cancel in dialog boxes was surprising to some users. Many users expected that the OK button would complete the action they were aiming to do in the dialog box, for example modify the object that was selected before opening the dialog box. Here, the OK button only closes the dialog and retains the settings for the next time that the dialog is opened. Also, the Cancel button usually cancels all the changes that were made when the dialog box was open but here it does not cancel things that were done before pressing Apply | The basic buttons (OK, Apply, Modify, Get, and Cancel) used in almost all dialogs should be redesigned. | 1 | 5 |

| | | | |
|---|---|---|---|
| of Modify. | | | |
| When defining drawing classifier settings, the user needs to close certain dialog boxes with Cancel in order to not cause unwanted additional changes in the model. This in inconsistent with what usually needs to be done. Users made mistakes with this in training | The task sequence of defining drawing classifier settings should be redesigned so that a dialog never needs to be closed with the Cancel button. | 1 | 1 |
| The dialog box for opening and closing files is different from what it usually is in other software applications. Instead of the common Browse… button, there is a text field where the user can write the destination folder. After writing the folder, he must press the Filter button which is not a commonly used button type. The observed users did not use the Filter button at all but found alternative strategies to access the desired file location | The Filter field in which a user can type the path of a folder should be replaced with a Browse button that enables the user to graphically see the folder structure. | 2 | 2 |
| Some users expected that a model or a drawing could be created using templates. | There should be templates available for creating a new model and a new drawing. | 3 | 3 |

Information presentation

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| It was very difficult for users to find items from the menus. | Menus should be redesigned to reflect users' task sequences and the results obtained in the grouping test. | 1 | 6 |
| On the component properties dialog box, there are numerous different controls such as fields, check-boxes and drop-down menus next to the enlarged picture of the component. Most of the controls do not have a title or other textual description. Users had problems determining which control is related to which property of the component. | Descriptive labels should be added to component dialogs to indicate the scope of the image and the relationship of images and text fields. | 2 | 6 |
| On the part mark content dialog, line breaks and backspaces are presented with symbols that resemble keyboard buttons but are not clear enough: <--' and <--. | The symbols should be replaced with text. | 3 | 5 |
| There are 67 templates available for reports, which is such a large number that users had problems finding a suitable template to use. Users will need only a few templates in their work. | The number of report templates available should be reduced. Currently, there are over 60 templates. Only the most commonly used ones should be shown on the basic dialog. | 4 | 4 |

User assistance

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| The state of drawings is indicated with one-letter abbreviations. For example, if a drawing is outdated, there is a letter n visible in the drawing list. This information is not sufficient for beginners that will not remember the meaning of the abbreviation. | One-letter abbreviations in the drawing list should be explained with a tooltip text. | 1 | 3 |
| If an action requested by the user cannot be performed for some reason, the user is not | The limitation for the number of views that can be created should be | 2 | 1 |

| | | | |
|---|---|---|---|
| informed about the problem. For example, if the user already has nine views of the model open and he tries to create more, nothing happens as nine is the maximum number of views. | removed. If the limitation remains, the user should be informed if views cannot be created because of the limitation | | |
| In some situations, warning messages appear that contain only a very general description of the problem and no instructions for recovering from the problem. An example of this kind of a message is "Illegal profile!" The message was shown in user observations when users tried to assign an object a profile that had not been defined yet. These ambiguous warning messages are not helpful to the user because they do not assist him in completing the action correctly. | The message text "Illegal profile" should be changed to one that describes the cause of the error and instructions for selecting a profile that is not "illegal". | 3 | 1 |
| Instructional texts are sometimes presented in the status bar on the bottom of the user interface but they are not descriptive enough. Users frequently asked the meaning of messages such as "Pick polygon position" or "Pick point". | Detailed instructions for picking points should be shown in the status bar. The current user interface shows the default prompt "Pick polygon position" in most situations. The prompt is not informative enough, and therefore it should be replaced with a more precise prompt. | 4 | 3 |
| Users did not know how to use certain simple dialogs, for example how to enter the direction into the copy field. | There should be instructions for using certain dialogs such as the move or copy dialog. | 4 | 2 |
| The amount of help that novice users need would probably be obtrusive for expert users. | There should be separate advanced and beginner modes that result in different amount of assistance being given. | 5 | |
| In some situations, the instructions in the status bar are not visible enough indication for something happening. In other situations, a message box would interrupt users' work. Needs for assistance types vary. | There should be several types of user assistance available, such as messages in the status bar, message boxes, and help dialogs accessible through buttons on dialogs. | 6 | |
| Users utilized the help file rather seldom for solving their problems. It seemed to be difficult to find issues from the help file. | The help topics should be integrated more closely with the system interface. | 7 | |

Error prevention

| **Learnability problem** | **Improvement suggestion** | **Guideline** | **Nb. users** |
|---|---|---|---|
| When creating grid views, users should first select the grid in the model, even if there was only one grid, and then press the Create button. Users frequently forgot to select the grid. | When grid views are being created and there is only one grid in the model, the grid should be selected automatically. | 1 | 3 |
| When creating a custom component, the user needs to select the part type to be either connection, detail, seam, or part. Users did not understand the meaning of these types and almost randomly selected one of the types. | The type of the custom component should be selected automatically. The available options - part, detail, connection, or seam - are unambiguous and selecting a wrong type easily causes errors. | 1 | 5 |
| One of the most frequent errors that novice users made was forgetting to number the model before creating drawings. This was described above. | If the user has forgotten to number objects before creating drawings, the option to assign numbers directly from the message dialog should be | 2 | 5 |

| | | | |
|---|---|---|---|
| | provided. | | |

**System structure**

Differences in functionality

| Learnability problem | Improvement suggestion | Guid eline | Nb. users |
|---|---|---|---|
| The fundamental differences between two dimensional drawing systems and three dimensional modeling systems caused some difficulties for users. They understood the issue in principle but still their thinking often reflected the two dimensional world that they had worked with. | The functionality of three dimensional modeling software applications differs fundamentally from the functionality of two dimensional drawing applications. The difference should be explained in the training and instructional material. | 1 | 2 |
| Users confused the concept of drawing and view. When they were asked to create drawings, they frequently opened the view creation dialog. In Tekla Structures, views show the model from different angles but they are not the same thing as drawings that show a static image of the model, with predefined settings. In two-dimensional drawing, the concepts of view and drawing are essentially the same. | This should also be explained in training and training material. | 1 | 1 |
| Users considered the concept of automatic numbering difficult to understand as there was no such operation in the software application they had used before. This is due to the fact that in two-dimensional drawing, the principal components are lines, not objects, and therefore objects cannot be numbered automatically. | This difference could be eliminated by making assigning numbers optional. | 1 | 5 |

Differences in interaction styles

| Learnability problem | Improvement suggestion | Guid eline | Nb. users |
|---|---|---|---|
| Users expected the button 'OK' to accept all changes made in a dialog box and modify the model accordingly. This is the case in the most office software applications and other common applications. However, in Tekla Structures, the model is only modified if the user presses the 'Modify' button. Users frequently forgot to press Modify and had to wonder why changes did not take effect. The functionality of the buttons was already discussed above. | The basic buttons should be redesigned. This was already discussed above. | 1 | 5 |
| The dialog for opening files, and especially the method for browsing directories, is different than usual. This was also discussed above. | The filter field should be replaced with a Browse button. | 1 | 2 |
| Users expected that an object could be resized by dragging one end of it to the desired location. They usually tried this first when they needed to resize parts and only after that considered alternative strategies for resizing. Dragging is a very intuitive interaction strategy and also used in many other applications, and therefore, it is natural that users expected it to work also in | The user should be allowed to resize objects by selecting a line or a face and dragging it to the desired direction. | 2 | 5 |

| | | | |
|---|---|---|---|
| Tekla Structures. Currently, resizing by dragging is not possible in the basic state of the interface; however, there is an optional 'drag-and-drop' state which allows for resizing by dragging. | | | |
| The use of object handles that are shown as points with different colors was unclear to users. | The use of handles should be reconsidered. The role of each handle should be indicated clearly. Information on each handle could possibly be provided as a tooltip. | 2 | 5 |
| Users had difficulties finding items from the menus. | Menus should be redesigned. | 3 | 6 |

Concept clarity

| **Learnability problem** | **Improvement suggestion** | **Guideline** | **Nb. users** |
|---|---|---|---|
| The terminology used in the system seemed not to be familiar to users. An example of an unfamiliar term is Freeze (refers to keeping modifications in a drawing). | The name of the Freeze command should be changed to e.g. Keep modifications. | 1 | 6 |
| Users called certain items with different names than what was used in the system. An example of such an item is reports that users called lists. | The name of the Report… command should be changed to List… | 1 | |
| One of the reasons why terminology is unclear is that there is no Finnish version of the system available but the users need to use it in English. Users are not familiar with all the domain-specific terminology in English. | If possible, a Finnish language version of the system should be provided. | 1 | |
| Some of the terms seemed to be misleading. For example, there is an option Export AutoCAD drawings among other export commands. However, there are also other export commands, such as Export DXF, that are commonly used for exporting information to be used with AutoCAD. Users almost always selected the Export AutoCAD drawings option and expected that all file formats that can be exported to AutoCAD are presented under it. | Either all file formats should be listed under the File > Export command or all file formats should be included on one dialog that can be opened by selecting the menu item File > Export > CAD. | 2 | 5 |
| Some system-oriented terminology has been used in Tekla Structures. For example, the meaning of custom component, that is a combination of objects tied together by users, was not understood or remembered in after-training observations. The concepts of AutoDefaults and AutoConnections were not remembered either. | System-related terms for component types (Custom connection / component, System connection / component, AutoConnection) should be changed. It makes no difference for the user whether the component was included in the system configuration (System component) or created by a user (Custom component). The command Define custom component could be renamed to Save a component. The term AutoConnection could also be omitted and the user could simply be provided the option to create connections automatically. | 3 | |
| The concepts of library profiles and parametric profiles were not familiar to users and that may be one of the reasons | The distinction of profiles to Library profiles and Parametric profiles should be removed. | 3 | 5 |

| | | | |
|---|---|---|---|
| why they considered it difficult to determine where to look for profiles. | | | |
| The meaning of the fields On plane, Rotation and At depth on connection dialogs was unclear to users. | The concepts On plane, Rotation, and At depth should be clarified with an image. | 4 | 1 |

Completeness of information

| **Learnability problem** | **Improvement suggestion** | **Guid eline** | **Nb. users** |
|---|---|---|---|
| Additional information on interface items is not easily available. There are no explanations or instructions visible on dialog boxes. The connection symbols, for example, are impossible for a novice user to understand without any reference information. | Tooltips that contain explanations for connection symbols should be provided. | 1 | |
| On object property dialogs, field labels are very short. Users were observed not to understand the meaning of certain labels. This was especially true for reinforcement and connection dialogs that contain a lot of fields that are connected to an image. Users often found the correct field with trial-and-error strategy and did not indicate they had understood the issue. Thus, a correct mental model was probably not formed. | Fields on component dialogs should be informatively labeled and written instructions should be provided for some fields. | 2 | 6 |
| The instructions in the status bar are incomplete. Therefore, they did not help users to understand and remember a procedure. Instead, users often voiced their amazement about the meaning of the instructions and proceeded with the task with trial-and-error strategy. This issue was already discussed above. | The types of user assistance were discussed above. User assistance should help the user to understand the interface items and the principles for using the commands. The assistance may also contain step-by-step instructions for performing certain actions, but it must still provide enough information so that the user can understand what is being done and not only passively follow the steps. | 3 | 6 |

**Training**

Conceptual information

| **Learnability problem** | **Improvement suggestion** | **Guid eline** | **Nb. users** |
|---|---|---|---|
| The meaning of unintuitive terms remained unclear even after training. The terms seemed to not have been explained thoroughly enough. Examples of terms that were not recognized or whose meaning was not known are Freeze, Part mark, and Class. The terminology issue was already discussed above. | The terms that were not understood by users and should be explained. | 1 | 6 |
| The hierarchy of connections remained unclear to users. This was already discussed above. | The relationship between different connection and component types, such as System components / connections, Custom components / connections, and AutoConnections, should be explained. A visual connection map could be created to illustrate the types of components and connections and situations in | 2 | 2 |

| | | | |
|---|---|---|---|
| | which they can be used | | |
| Users were confused about which changes to drawings should be done in drawing classifier, which ones in drawing property dialogs, and which ones using the template editor. They did not seem to understand the scope of each of these methods of modifying drawings. | The relationship between different tools for modifying drawings (drawing classifier, template editor, drawing properties dialog) should be explained. | 2 | 5 |
| The logic of updating drawings remained unclear to users and they had problems with it in after-training observations. | The logic of updating drawings should be explained in training. | 3 | 3 |
| The differences between three dimensional modeling and two dimensional drawing were not fully understood. | The fundamental differences between drawing and modeling systems should be explained. | 3 | 1 |
| The difference between views and drawings was also not understood after the training. This was already discussed above. | It should also be explained that in this system, views and drawings are two totally different things. | 3 | 1 |
| Users did not understand the conception of numbering even after the training. They did not see the connection between the numbering setup dialog, numbering prefix fields on object property dialogs, and the numbering command. They often forgot to run numbering before creating drawings and as they got the error message, they completed the numbering procedure mechanically without even checking the numbering options. Some problems with numbering were already discussed above. | The idea of numbering and the reason for why it is needed should be explained. | 3 | 5 |

Exercises

| **Learnability problem** | **Improvement suggestion** | **Guideline** | **Nb. users** |
|---|---|---|---|
| Several task sequences that had been practiced in training were not remembered in observations after the training. Examples of task sequences that could not be memorized as they were practiced only once in the training are: creating fittings, creating AutoConnections, defining Custom Components, and creating part cuts. | After the instructor had demonstrated an operation, learners should be given exercises that require applying the operation to new situations. | 1 | 6 |
| Task sequences that users had performed with extensive help from the instructor could not be repeated by users after the training. | Example of this kind of a task sequence is defining the drawing classifier settings. Participants should always work through the exercises themselves and strive to understanding what is being done. | 2 | 5 |
| Users often commented that they remembered something was done in training but could not memorize how it was done. Users may have done exercises passively as they did not remember how things were done. | Questions could be asked that encourage learners to think about the how an operation can be generalized to other situations or what is the relationship between concepts. | 2 | 6 |
| Users did not remember what Drawing classifier or AutoConnections were used for. | It should be explained for what purpose the Drawing classifier or AutoConnections are practiced. | 3 | 5 |
| Users did not remember that certain operations can only be done in the drawing state and certain ones in the modeling state. | It should be explained in which conditions the practiced operation can be performed. | 4 | 5 |

Instructions for basic interaction

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| Users did not know how parts can be moved or resized, even though this is a basic operation in the modeling process. They tried to stretch parts by grabbing them even though this is not possible in the basic state of the interface. They did not remember that part dimensions are entered to the fields on the part properties dialog. Also using handles, snapping and picking, and determining view properties were observed to be difficult even after receiving training. Other issues have already been discussed above but snapping has not. | Basic operations such as moving parts, using handles, snapping and picking, determining view properties, and changing part size should be explained and demonstrated. | 1 | 5 |
| Also the snap settings were not explained clearly enough and users did not know how to adjust the settings. | The available snap settings and how to adjust them should be explained and demonstrated. | 2 | 5 |
| Only one user was observed to use the Get button in training or in the post-training observations. The Get button fills the dialog box with the properties of the selected object. This is useful when setting the properties of several objects to be similar. The use of the button was not demonstrated in training and that resulted in most users not using it. | The functionality of the Get button should be explained and demonstrated in training. | 3 | 5 |

Instructions for solving problems

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| When users faced problems in the post-training learnability test, they often gave up without trying to solve the problems. They often stated that they cannot proceed and asked help from the person that was observing the learnability test. Users needed help for 88% of the tasks on average in the observation session that was held right after the training and for 74% in the session that was held two months later. Both percentages are rather high. Users seldom used the help file when they faced problems. | The help file should be introduced to training participants. They should be told how the help file can be installed (it requires separate installation) and how it can be accessed. An overview of the contents of the help file should be given | 1 | |
| Users did not find information from the help file very easily. | Different methods to search for information should be demonstrated. The table of contents, index, and search function should be introduced. | 2 | |
| Users had not contacted Tekla Structures support after the training even though they had some questions that they asked from the researcher in the learnability test. | The email address and phone number of the support personnel should be told to training participants. | 3 | 6 |
| There were many errors that were faced by a large portion of users in learnability tests. | The errors that novice users often make should be addressed in the training. A variety of common errors has been presented in this study. Instructors also know the most common causes of errors. | 4 | |

Motivational content

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| As described above, the relationship of some basic concepts, such as views and drawings, was not understood by all users. | Presenting some orientational material in the beginning of the training could help users to understand the basic idea of each core component of the system. Advanced issues would be easier to learn if the principles would have been taught in the beginning. This is also in line with the theory of mental models: when the orientational information is available, the user can correctly connect the new information with the existing mental model and the danger of misconceptions is reducedA written overview of training contents should be provided. The table of contents of the training material should be reviewed before the training. Orientational material could be sent to participants beforehand. | 1 | 6 |
| Only two of the six users had used the Tekla Structures system during two months' time period after the training. Four others said that they had been busy with other projects so that they could not find time for all the initial set-up that is needed for taking the new system into use. | Needs of the training participants should be carefully researched. They should be asked which features they think they will need in their work. Also other wishes from participants should be taken into account. This will motivate them further. | 2 | 4 |
| Same as above | Users should be contacted by the instructor or other support personnel after the training. Contacting users afterwards may motivate them to continue to learn to use the system. They may also have questions that they will ask if somebody is available for answering. | 3 | 4 |

Coverage of system functionality

| Learnability problem | Improvement suggestion | Guideline | Nb. users |
|---|---|---|---|
| Several users said they need to export and import data to work with colleagues that use different software applications, but exporting and importing were not covered in training. | Exporting and importing data should be covered in training. | 1 | 3 |
| The amount of material for each core task was not very well balanced. | More material should be available for creating grids, updating drawings, exporting or importing data and specifying model properties. Less material could be needed for creating concrete or steel parts, creating connections, and creating drawings. | 2 | |
| The amount of time spent for learning each core task was not very well balanced | More time should be spent with teaching how to modify concrete or steel parts as users had a lot of problems with it. More time should be spent with teaching how to create | 3 | |

| | | | |
|---|---|---|---|
| | views, create views, create numbering, update drawings, create reports, or modify catalogs as well. Exporting and importing data and specifying model properties should be covered in the training. Less time could be spent with creating concrete or steel parts, creating reinforcements, and modifying drawings. | | |

Material types

| **Learnability problem** | **Improvement suggestion** | **Guideline** | **Nb. users** |
|---|---|---|---|
| Users estimated an integrated help to be useful. An integrated help means that the user need not open a separate help window but instructions are attached to user interface items. | As users considered integrated help to be useful, it should be provided. Instructions could be given in the status bar, integrated into buttons on dialogs, or displayed in some other way. This was already discussed when the learnability factor 'user assistance' was covered. | 1 | |
| Users were observed to use mainly the printed training material in the training. | Providing printed material to all users may not be feasible because of e.g. the need to update the manual, but it should be considered as an alternative. Opening the electronic training material on the screen would have made the screen rather cluttered. Therefore, either printed material should be available in the future as well, or there should be dual monitors in the training classroom. | 2 | 6 |
| There was not enough time to go through all the material. The amount of material is very large. | The amount of material should be cut down by writing shorter instructions or leaving out some issues that are not central for users. It was mentioned above that less material could be available on creating concrete or steel parts, creating connections, and creating drawings. | 3 | |
| There are many heading and text styles in the training material but the relationship of different styles is not always clear. | The material structure could be clarified by e.g. numbering the third heading level and including only one heading level without outline numbering. Each subsection should be started on a new page. | 4 | |
| Users asked for material in Finnish. Users would benefit a lot from material that would be in their native language. | A Finnish training material package should be written. | 5 | 2 |
| There is table of contents or index in the training material, which makes finding the relevant material difficult. | A table of contents and index should be added to the printed training material. | 6 | |