Transfer of High-Level CAD/CAM Technology to Developing Countries

Dissertation zur Erlangung des Grades

Doktor-Ingenieur

Fachbereich D, Abteilung Bauingenieurwesen

Bergische Universität Wuppertal

Lehr- und Forschungsgebiet Bauinformatik

Univ.- Prof. Dr.- Ing. Georg Pegels

vorgelegt von

Hamid Reza Azadnia, MSc.

Wuppertal, Dezember 2007

Dissertation

Dissertationsschrift eingereicht:

Dezember, 2007

Mündliche Prüfung und Disputation:

Mai, 2008

Prüfungskomission:

Univ.-Prof. Dr.-Ing. Pegels (1. Gutachter)

Univ.-Prof. Dr.-Ing. Harte (2. Gutachter)

Prof. Dr. Hamid Eslimy-Isfahany

Univ.-Prof. Dr.-Ing. Schlenkhoff (Vorsitzender)

Diese Dissertation kann wie folgt zitiert werden:

urn:nbn:de:hbz:468-20090602
[http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20090602]

Acknowledgments

First of all, I wish to thank my supervisor Univ.-Prof. Dr.-Ing. Pegels for his kind support, helpful comments, and editing of this thesis.

I am very grateful for the scholarship I received from DAAD (Deutscher Akademischer Austausch Dienst/German Academic Exchange Service) during this study.

I wish to thank the members of the examining committee Univ.-Prof. Dr.-Ing. Harte, Univ.-Prof. Dr.-Ing. Schlenkhoff, and Prof. Dr. Hamid Eslimy-Isfahany for their beneficial comments.

I also would like to thank Prof. Dr. rer.nat. Beisel for his very useful comments on the survey conducted in this research.

Special thanks to Mr. Dipl.-Ing. Weckmann, and Mr. Dipl.-Ing. Niederprüm (Chief Managers of ICW GmbH) without whose cooperation and assistance this research would not have been successful.

During one year of my experimental investigations at ICW GmbH, I had the opportunity to work together with Mr. Dipl.-Ing. Barwich. I truly thank him for all the things I could learn from him. I am sure these are going to help me in my future professional carrier.

I wish to thank my friend and my PhD fellow student Mr. M. Samsamshariat, MSc for his useful comments and generous assistance throughout this research.

I also would like to convey my thanks to all friendly co-workers at ICW who patiently took part in the survey. Obviously, without their cooperation, the survey would have been impossible.

I would like to express my gratitude to Mr. McKelvy for reading and editing some parts of the text.

Last but not least, I take this opportunity to express my deepest gratitude to my beloved wife Mrs. Shabnam Kabiri, MSc and my parents for their wholehearted support and patience during this study. I dedicate this thesis to them.

Abstract

Many developing countries around the world are in dire need of upgrading their building industry to meet the basic demands of their society, such as housing. In this regard, the successful transfer of high-level CAD/CAM technologies is indeed the best solution. There are indeed many problems to overcome, but amendments to the User Interfaces (UI) of CAD systems, and their adaptation to specific users in developing countries, is clearly a key step forward.

The present study reveals that the market-leading CAD/CAM software systems of today are heavily language-dependent and relatively inconsiderate to the usability concerns of even native-speaking users. Clearly, language barriers and complex user interfaces are very likely to trip over CAD/CAM users in developing countries. This calls for the realization of more intuitiveness and directness as well as minimising the use of textual characters in the interfaces of these programs. In this aspect, a user profile of CAD/CAM engineers in developing countries has been developed which reflects their specific mental models, preferences and limitations. Moreover, a survey was conducted to determine the most frequent commands in high-level CAD systems for steel detailing. The results are useful for prioritising the effort necessary to develop text-free GUIs for international use. Finally, by attending and working on real projects, a thorough task analysis has been carried out that is based on day-to-day use of these systems. These results are key informative tools laying the groundwork for the innovation of CAD/CAM software systems. Subsequently, several graphical prototypes of basic GUI components, as well as specific command dialogs, are presented.

In the end, important aspects of international cooperation in worldwide networks are discussed, as well as some of the bottlenecks hindering such collaborations in the specific case of German-Iranian companies. This discussion aims to raise awareness on both sides for the sake of future cooperation. It should furthermore emphasise that technology is not the only prerequisite to successful collaboration and that, indeed, there are many challenges to overcome.

Zusammenfassung

Weltweit müssen viele Entwicklungsländer dringend ihre Bauindustrie verbessern, um die Grundbedürfnisse ihrer Bevölkerung wie z.B. den Wohnungsbau befriedigen zu können. In diesem Zusammenhang ist ein erfolgreicher Wissenstransfer von fortgeschrittener computergestützter Konstruktion und Fertigung (CAD/CAM) die beste Lösung. Um dieses Ziel erreichen zu können, müssen viele Probleme bewältigt werden. In diesem Sinne spielen die Verbesserung der Benutzungsoberflächen (GUI) von CAD-Systemen und die Anpassung an Benutzer in Entwicklungsländern offensichtlich eine entscheidende Rolle.

Die vorliegende Arbeit macht die große Sprachabhängigkeit und nur bedingt taugliche Gebrauchstauglichkeit, sogar für muttersprachige Benutzer, von heutigen marktführenden CAD/CAM-Programmen deutlich. Sprachprobleme und komplexe Benutzungsoberflächen stellen Benutzer in Entwicklungsländern große Schwierigkeiten vor. In diesem Sinne, wurde in dieser Arbeit ein Benutzerprofil eines typischen CAD/CAM-Ingenieurs in einem Entwicklungsland ausformuliert, das seine Mentalität, besonderen Vorzüge und Beschränkungen reflektiert. Darüber hinaus wurde eine Umfrage durchgeführt, die die häufigsten Befehle in leistungsstarken CAD/CAM Stahlkonstruktionsprogrammen erforscht. Die Ergebnisse helfen bei der Entwicklung der für internationale Märkte geeigneten textfreien Benutzungsoberflächen. Ferner wurde eine auf der täglichen Anwendung basierende Benutzertätigkeitsanalyse ausgeführt. Anschließend werden mehrere grafische Prototypen auf Basis dieser GUI-Komponenten sowie spezifische Dialogboxen entwickelt.

Anschließend werden wichtige Aspekte internationale Kooperation in weltweiten Netzwerken diskutiert und dabei typische Probleme, die in deutsch-iranischen Kooperationen auftauchen mögen, werden formuliert. Es zeigt sich, dass CAD/CAM Hochleistungssysteme allein auch bei textarmer Benutzungsoberfläche keine wirtschaftlich tragfähige, in Europa wettbewerbsfähige deutsch-iranischer Kooperation von Ingenieurbüros und Baufirmen sicher stellen. Das beharren auf alt hergebrachten Konstruktionsweisen verhindert praktikable arbeitsteilige Kooperation. Schulungen mit „Training on the job“ könnten langfristig zum Erfolg führen.

Contents

1. Motives and Target

Enormous population growth in developing countries clearly demands housing and new jobs on a scale of many millions worldwide. On the other hand, many developing countries are often exposed to devastating earthquakes. In the home country of the author, Iran, millions of young professionals suffer from a high unemployment rate as well as a shortage of housing. Following the baby boom in 1980's, around 800 000 families are formed every year, while the ultimate annual capacity for building houses stands at roughly 450 000 [1]. Many of these houses are not adequately resistant to earthquakes in a country that is frequently visited by them. Fig. 1-1 shows an example of such a building which despite being constructed in recent years, collapsed during the Bam earthquake (in December 2003). Therefore, affordable, earthquake-proof housing would be one of the most significant targets of humankind since it also has the potential to generate many jobs in the housing industry.

Fig. 1-1: Recently-constructed building collapsed from earthquake tremors in Bam, Iran (Courtesy of BHRC)

Scientific research worldwide has shown how earthquake-proof housing must be practised; nevertheless, we have experienced a deplorable situation that, in spite of effective knowledge and proven technologies, most people in developing countries do not reside in earthquake-proof houses and even new houses, for the most part, fail to fulfil all the building code requirements. Given this situation, any strong earthquake will result in a sad human disaster.

A closer look at the home country of the author reveals the fact that, although respective measures are taken in the design phase of structures, they are not fully practiced in the construction phase. This is mainly due to unskilled workmanship as well as a lack of competent control and supervision. To a large degree, structural designers have no control over the construction phase to preclude the faults of the on-site workers. In other words, the design concept devised by the structural designers is by far hardly realised on site [2].

There must be several basic reasons why housing in developing countries fails to meet standards. To discover the reasons behind this and propose reliable technical solutions to solve each of these is a fascinating, significant and innovative target for courageous civil engineers.

To successfully tackle this problem and address the scientific aspects, an international team was set up at Bergische Universität Wuppertal. The scale of the issue was ascertained and innovative basic solutions were proposed by the contribution of this team to the competition for the MONDIALOGO Award of UNESCO, in which the team placed among the forty finalists [3]. The team modified the architectural style of the famous German "Fachwerkhaus" to attain cultural acceptance in developing countries. The "Fachwerkhaus" steel structure, composed of channel or L-profiles, remains visible from outside not only as an architectural element but also as explicit as visible proof of its lifelong earthquake-safety, see Fig. 1-2.

The topics investigated by the MONDIALOGO team included management and financial aspects, earthquake-proofing by means of base isolation, and adaptation of typical structural components for the most efficient fabrication/construction process. The research of the author within this team addressed high-performance CAD/CAM systems, which are the key tools to innovate the design and detailing of buildings in developing countries.

This is clearly a decisive step towards a powerful and efficient building industry amongst global competition. Driven by a CAD/CAM system, all subsequent processes for construction and fabrication can make use of the efficient and precise technology of NC machinery [4]. This way, all structural components can be fabricated automatically so as to pre-empt unnecessary human intervention. Afterwards, the steel members will be connected exclusively by rivets to avoid any poor welding by unskilled labourers.

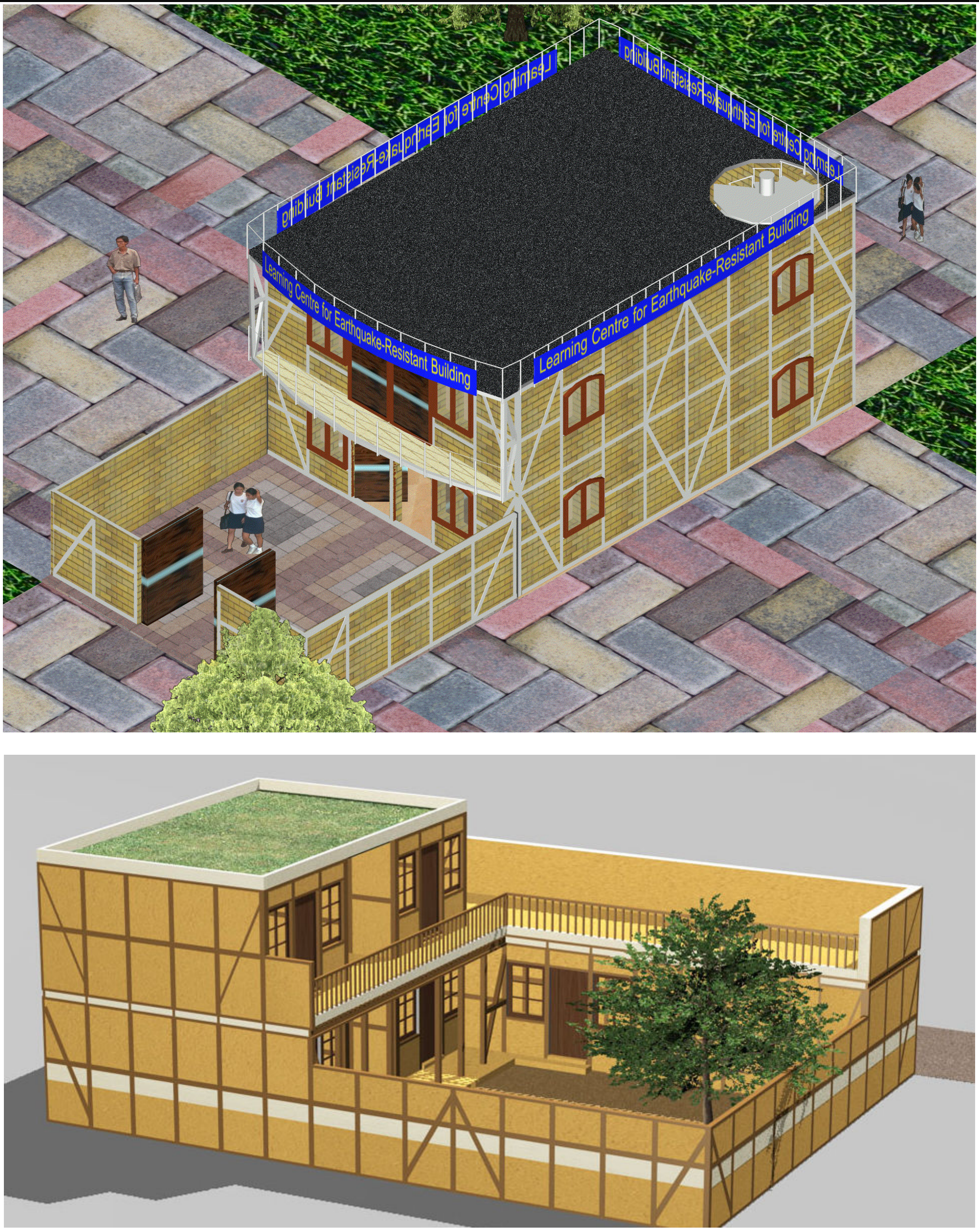


Fig. 1-2: “Fachwerkhaus”, earthquake-proof principle for developing countries

To build individual houses in such a quality-proof manner in developing countries, a reliable planning phase must be established. In this context, “Transfer of High-Level CAD/CAM Technology to Developing Countries” allows for a reliable planning phase.

The scope of this research work involves high-performance CAD software for the building industry. The objective is to seek and propose solutions on how the user interfaces of such programs should be modified for users in developing countries. The target software to investigate includes Bocad-3D version 20 and TeklaStructures version 12.0. Both programs are the current market leaders in the world. Even with the assumption that these software products have been designed to match the requirements of their native target users—a statement which is disputed by this work—they are not designed at the moment to be operated by specific users in developing countries. This identifies a gap to be dealt with in the transfer of the technology.

Exactly why this observation matters and why different users have different requirements are the questions to be answered in Chapter 2 of this work, which introduces usability as a means to measure the quality of interaction with a system. Subsequently, major characteristics of human cognitive systems, which help designers better understand the underlying guidelines of user interface design, will be briefly discussed. Following, important techniques of “direct manipulation” and “metaphors” are discussed, which can highly increase the directness and intuitiveness in CAD programs for civil engineers. One of the main objectives that metaphors could serve is to lower the amount of text in software by using the global legends found in construction drawings, which are understandable world wide by civil engineers. This would be a decisive step toward the internationalisation and transferability of a CAD/CAM program. The chapter concludes with basic guidelines of computer interface design and how some of these guidelines are violated in existing software.

In Chapter 3, a user profile of CAD/CAM engineers in Iran is presented, which may as well serve for developing countries alike. Such user profiles are needed for getting to know the target user. Moreover, results of the survey conducted during this research are presented which identify the most frequent commands found in a CAD program. These results will help product developers to prioritize their efforts in enhancing the usability of their systems.

Chapter 4 addresses the basic Graphical User Interface (GUI) components such as pointing devices, keyboards and GUI controls. This chapter aims to emphasise that even “apparently simple” user interface components contribute to the overall usability of software. This lends testimony to the usability slogan that “details matter”. Chapter 5 pursues the objectives of this research in a broader sense by providing typical solutions about major commands in high-performance CAD/CAM systems.

Although the approach proposed in Chapter 3 together with the standard solutions presented in Chapters 4 and 5 enhance usability from the perspective of users in developing countries, and thereby contribute to the transfer of technology to these countries, successful application is not as ensured as it would be given their usage within more modernised industry. This issue is addressed in Chapter 6, which deals with some of the practical challenges faced by the international collaboration of German-Iranian civil engineering teams. In addition, “the lessons learned” from different attempts of such collaborations made during the course of this research are presented.

The final chapter summarizes the results and presents motives for future research.

2. Basic Principles and Guidelines of Graphical User Interfaces (GUIs)

In Bauinformatik (Applied Informatics in Civil Engineering), it is necessary to know and apply the knowledge in computer and communication science in order to develop and innovate IT-tools in civil engineering. This chapter intends to achieve the following objectives by referring to the relevant knowledge in computer science and cognitive psychology:

- why different classes of users need different adaptations;
- why application of construction drawings in civil engineering software can be useful for civil engineers worldwide;
- scientific background for the solutions proposed in chapter 4 and 5.

2.1 Usability

According to ISO9241-11 [5], usability is defined as “The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.” In other words, usability of a product measures how easily and quickly the users can accomplish their task by using that product. Karat knows usability as a specific context-dependent attribute of interaction with a product not merely as an attribute of a product alone [6]. In the literature of Human-Computer Interaction (HCI), the term “usability” replaced “user friendly”, which in early 1980’s “had acquired a host of undesirably vague and subjective connotations” [7].

It must be noted that usability alone is not the decisive factor for market acceptability of a product. In fact, market acceptability of a product is determined based on its usefulness which in turn can be broken down to utility and usability [8]. Utility addresses the functionality of a system; whereas usability is “the question of how well users can use that functionality” [9]. According to Microsoft [10], “Both qualities are necessary for market acceptance, and both are part of the overall concept of usefulness. Obviously, if a program is highly usable but does not do anything of value, nobody will have much reason to use it. And users who are presented with a powerful program that is difficult to use will likely resist it or seek out alternatives.”

Usability is one of the corner stones of all HCI design guidelines and is seen equally important as other aspects of the system. This philosophy, also called user-centred design, highly values users’ requirements in the design process and dictates that designers must incorporate user’s need from the early stages of design [10]. Johnson knows this as the “mother of all principles” of HCI design and prescribes in a similar way: “Focus on the user and their task, not the technology”

[11].

In order to quantify usability, different researchers have attempted to define different attributes for it; for example, ISO 9241 considers three attributes of Effectiveness, Efficiency, and Satisfaction for usability. Folmer et. al present a full account of different usability attributes among researchers [12]. In general, there is almost a consensus on five attributes of learnability, efficiency, memorabiltiy, errors and satisfaction to represent usability. These five attributes normally function as measurable components enabling systematic approach, evaluation and improvement of usability. Nielson (1993) defines each of these terms as follows [9]:

- Learnability: is the system easy to learn?
- Efficiency: once learned, how productive it is to work with the system;
- Memorability: how easy is it for an infrequent user to remember the learned skills within the system?
- Errors: is the error-rate in the system low? Can users easily recover from their errors within the system?
- Satisfaction: how pleasant is the system use?

Nielson further gives a full account on all these terms and their evaluations in his book [9]. On the other hand, according to Miller [13], usability attributes are not equally important among different classes of users; for example, while for some users learnability may have the most importance, for some others efficiency counts. Therefore, knowing the user is important to know what to optimize for.

It is noteworthy to know that design guidelines alone are not sufficient to guarantee usability of software. Guidelines are either too general or too specific to be applied in all applications [14]. They are too general because they are inherently designed to apply to a variety of cases—refer to Schulten's (2007) account of this issue on ISO EN 9241 [15]. On the other hand, they may be too specific when they are taken from studies of particular applications since "unique characteristics of these applications make general conclusions impossible and, again, may invalidate the recommendations" [16]. However, guidelines can be used to ensure a consistent look and feel within the application or between applications, although "consistency by itself is not the ultimate goal" and a consistent interface is not necessarily a usable one [14]. Thus, the only way to achieve the desired level of usability is to put the system at scrutiny of the intended users.

Usability matters on a number of grounds. Aside from the fact that usable software sell better, enhance productivity, and users enjoy working with [17], it is not hard to recognise how usability can save enterprises many uncontrolled overhead costs; the reasons are clear: User productivity increases, user's error as well as training costs decrease [18]. Some rule-of-thumb calculations confirm this very easily as shown by Theofanos and Laskowski [19]. A similar calculation could show this significance particularly for CAD programs in steel detailing:

According to the survey conducted in this research (see section 3.2), a user carries out about 62 commands on average every five minutes or 12x62=744 commands an hour. Now, assuming that through better usability, an improvement of 0.1 sec per command is achieved, the conceivable saving per user in a year will be:

1 CAD Engineer

x 9x0.6 hours a day working with the program (Assuming that they work 9 hours a day and they spend 60% of their time working with the program)

x 220 days per year (365 excluding weekends, holidays, and vacations)

x processing time reduced by 744x0.1 seconds an hour ÷ 3600

x hourly rate of 42,5 € (the estimated hourly rate of 40-45)

≈ 1044 € savings per year per engineer

Given that programs like Bocad-3D and TeklaStructures have certainly sold more than 10 000 licenses, one could imagine what an enormous amount of money could be saved by a slight improvement in usability. There are also more official estimates, which address the costs incurred by the industry due to poor software usability; for example, according to Landauer (1995) "Inadequate use of usability engineering methods in software development projects have been estimated to cost the US economy about $30 billion per year in lost productivity" [20]. More statistics concerning the financial aspects of usability are presented in [18]. Nielson provides more "well-documented examples" of cost savings from the use of usability engineering methods [9].

Besides financial justifications, usability matters since usability issues often remain hidden and unreported. The reason is that users tend to blame themselves for usability problems and this is how/why they become frustrated with low-usable system. Users know whom to blame when certain functionalities are missing within a system or when a system is unstable [13], but when they encounter usability problems within a system, they may firstly do not recognize them as a problem and secondly, they may blame themselves and question their own capability to use the system!

The discussed significant aspects of usability obviously highlight the important role of the User Interface (UI) of a software system; through the UI, the user interacts with the system and as Raskin interprets, for the end-user the interface is the system. Therefore, the quality of the interaction is determined by the quality of the interface. Raskin knows this quality to be so crucial that it can highly affect all other aspects of the system [21]: "The quality of any interface is ultimately determined by the quality of the interaction between one human and one system. If a system one-on-one interaction with its human user is not pleasant and facile, the resulting deficiency will poison the performance of the entire system, however fine that system might be in other aspects.“ Thus, there has been a growing emphasis among researchers on how to incorporate usability in interface design from early stages of software development, since later amendments are normally harder and costly [22, 23].

Gould et al. provide a short summary of how to design usable computer systems identifying four tenets of usability design process [24]: Early and continual focus on users, empirical measurements, iterative design and integrated design, wherein all aspects of usability evolve together. Based on similar principles, many brand-name techniques for user-centred design have also been developed (e.g., GUIDE, STUDIO, OVID, LUCID) [13]. Other researchers consider usability of products to be dependent on a series of conditions [25]:

- the physical and logical characteristics of the products;
- the characteristics of the users;
- the characteristics of the tasks;
- the impact on the physical, social and organisational surroundings (environments).

What is clear is that usability engineering is an iterative process of designing, implementing and evaluating by user testing as Miller has summarised [13]:

- iterative design using rapid prototyping;
- early focus on users and tasks;
- evaluation throughout the iterative design process.

Early and continual focus on the user includes two key aspects: first, learning about human abilities and limitations, and second, designing for a particular group of people [26].

Designing a usable system requires an understanding of human abilities and limitations. This leads to the paradigm that the “basic machine” can fit in better with what human finds natural and easy to do—a trend, which first appeared in UI industry by a 1984 marketing brochure of Apple Computer, Inc. [27]. Herein, cognitive psychology has lent much support to HCI discipline in understanding the nature of human’s cognition. By applying findings of cognitive psychology, designers and trendsetters in HCI have been able to develop guidelines and principles, which better meet the characteristics and cognitive limitations of human users. According to Raskin, before focusing on specific differences of the human users, it makes more sense to learn about their commonalities, which root in their “[universally] common mental attributes” [21]. To better understand the role of cognetics (cognitive engineering), one may compare it with ergonomics [21]: Ergonomics addresses the physical characteristics of human beings in design, whereas cognetics incorporates their psychological characteristics. Norman describes cognitive engineering as a type of applied cognitive science that aims to apply findings of science in the design and construction of machines [28].

The key characteristics of human cognitive system are memory, attention, learning, decision making, error, and mental models. Within the scope of this research, all these characteristics are the same among all human beings except mental models, which may significantly differ among users from different nationalities. Therefore, significance of mental models is discussed more thoroughly, while a short review follows next on other aspects of human's cognition mostly to identify their relevance to UI design guidelines.

2.2 Key Characteristics of Human Cognitive System

Memory

Human memory consists of Working Memory (WM) and Long-Term Memory (LTM). Working memory is a temporary store which functions also as a "slow processor" of incoming information from sensory buffer; it examines, evaluates and compares different mental representations [27]. Moreover, it is used to carry out mental calculations and predictions. Finally, working memory acts as a gate for information to pass and enter the long-term memory [27, 29]. The rehearsal capacity of WM depends on different factors but is extremely low (7±2 units/chunks) [30].

In contrast to working memory, human long-term memory is "unlimited in capacity and permanent in storage"; however, storage and retrieval processes are slow, unreliable and difficult. Memory retrieval processes is primarily based on recognition and recall. In general, in equal conditions, human beings can better recognise a learned item than to recall it without prompting cues [31]. Moreover, frequency of access strongly influences the retrieval ability [27].

Hitch and Gardiner provide an extensive summary of memory characteristics applicable to HCI design [32, 33]. Some of these may be paraphrased as follows:

Working memory:

- In general, the load on working memory has a direct influence on the speed and accuracy with which people can process the information to carry out different tasks;
- Multi-tasking and mental parallel processing increases the load on working memory;
- The more information to be stored and remembered temporarily "or held in mind", the more load is imposed on working memory;

- Task complexity increases the load on working memory even for "familiar well-learned" tasks.

Long-term Memory:

- Pictures and visualizable graphics aid recall and are in this sense more effective than textual information;
- Recall is enhanced if retrieval operations are compatible with storing operations;
- Recall is easier "if the item was stored in response to a cue or rule, than if the item was simply read or seen at the time of storage";
- "Context" affects recall; i.e., recall is enhanced if the retrieval takes places in a similar context to storing.

Mayhew proposes various implications of these rules in HCI design [27]. In short, these rules advocate consistency, simplicity, use of icons and visual cues, as well as facilitation of structured learning and training experience for user in manuals and tutorials.

Attention

Owing to the limitations of working memory, human's information processing system selects the aspects of the environment to be processed. This is called selective attention, as it is intentional and voluntarily. Selective attention is useful because it lowers the load on working memory by limiting the number of options to be processed. Problem, however, arises when the mind selects the cues that "stand out rather than being useful" [29]. In other words, we err in our attention because we do not always select the correct cue from the environment. Moreover, we sometimes fail to concentrate on a specific clue despite our effort to do so ("focused attention") [29]; that is, we may easily get distracted by interfering clues from the environment. This results in "divided attention", where some of our attention is involuntarily expended for processing the undesirable clues [29].

Learning

From the perspective of cognitive psychology, learning is indeed a complex phenomenon, which is well beyond the scope of present discussion. However, interaction designers need to know the major principles that facilitate human's learning. By consciously applying these rules, designers can lower the learning period which in turn results in increased productivity.

Factors that facilitate learning are analogy, structure and organisation, as well as consideration of a levelled learning pace [27].

Analogy allows users to build upon their prior knowledge, which makes the storing and retrieval of the new information easier. However, analogy is not always helpful, because human beings are very likely to draw inappropriate analogies of information in the environment which results in "erroneous comparisons and conclusions preventing them from recognizing possibilities for new function" [34]. (See section "Mental Models" for more information about this phenomenon).

Structure and organisation allows new information to be viewed as associated pieces. This can overcome the inherent low capacity of working memory and thereby enhances one's learning [27]. Learning is greatly facilitated if the user can learn small pieces of information and then continuously build upon and add to that core knowledge [35].

The fact that learning is facilitated by analogy and structure requires that designers must strive for designs which "… draw upon the user's current knowledge and skills as much as possible and present information in simple, well-organized way. [To this end,] arbitrary or jargon-like function names and unnecessary new concepts [must be avoided]" [27]. This gives way to useful application of metaphors like the desktop metaphor, which minimises the necessity for learning jargons and many unfamiliar new concepts (see section 2.3).

Finally, Carroll and Rosson (1987) make two observations about users learning computer systems (known as paradox of the active user among experts) [34]:

1. Users focus on accomplishing their tasks with a system rather on learning to use a system more effectively and efficiently (Production paradox). Therefore, the system must promote learning by making learning easy and rewarding. In addition, users' desire for their productivity should be exploited to drive learning.

2. When using a system, users do not always draw the correct analogies (Assimilation paradox). Therefore, potential misleading analogies must be discouraged by performance feedback; for example, metaphors must be "true and complete", but must be departed where this can be easily accommodated by users. In other words, their tendency to draw similar analogies must be exploited to offer more power and flexibility in using metaphors.

Nielson describes the paradox of active user as such:"…is a *paradox* because users would save time in the long term by taking some initial time to optimize the system and learn more about it." He prescribes that "we cannot allow engineers to build products for an idealized rational user when real humans are irrational: We must design for the way users actually behave" [36].

Problem Solving and Decision Making

Within the scope of human-computer interaction, characteristics of human's decision making are of especial importance. This is because users are constantly making decisions when carrying out their tasks through interaction with the system. Therefore, human's decision-making characteristics should be well-known to system designers.

Several components of human information processing system are involved in decision making [29]: perceptual sensory, selective attention, working memory, and long term memory. The process includes filtering the clues, forming hypothesis, diagnosing the current state, and determining a choice of action. Three statements that are of relevance to interactive system designers about human problem-solving behaviour can be made:

1) “People are more often heuristic than algorithmic in decision-making and problem-solving strategies” [27]. Heuristics are procedures which are valuable for getting a solution, but the solution is not guaranteed to be the optimal one [37]. In contrast, algorithms are procedures which specify actions guaranteed to produce the desired outcome. People use heuristics in decision-making either in the diagnosing phase or in the choice phase. There are two reasons to use heuristics: first, it is normally faster and hence, fits better within the time constraint of decision-making; second, it consumes less cognitive resources. There are two types of heuristics in diagnosing phase: “representative heuristics” and “availability heuristics” [29]. “Representative heuristics” address the fact that people try to match the clues they receive from environment with the patterns of incidents previously restored in their long-term memory. “Availability heuristics” suggest that more frequent events are recalled more easily to form our hypothesis. However, other factors which may precede “frequency” in our availability heuristics are “recency”, and elaboration in the memory—how elaborated the previous pattern is stored in the memory [29].

2) “Humans do not always meet the best/optimal choice even when these strategies are available to them” [27]. Normally, choosing the best strategy costs many cognitive efforts. Therefore, human beings normally “satisfice”, which means they choose a strategy without any guarantee that it is the “absolute best” strategy [38-40]. This does not necessarily mean that they choose a wrong strategy, but rather a sub-optimal one in order to save cognitive resources.

3) “There is a natural tendency to learn better strategies with practice” [27].

Error

It is a fact of life that humans make error. Therefore, it is one of the most important characteristics of usable systems to minimise error-prone situations and to allow fast and safe error recoveries. Hence, in the realm of human-computer interface interaction, researchers and practitioners in the area of usability and human error are advocating the motto: “To err is human, to forgive is the role of the computer interface”—putting all the pressure on the designer [41]. For this purpose, it is obviously useful to understand the nature of human error, so that designers can provide solutions that are more specific. Here again, research from cognitive science has provided good background knowledge. Prabhu and Prabhu provide a summary of human error classifications [41]. Mayhew classifies human errors in terms of human information processing subsystems [27]. A summary of his classification as well as corresponding user interface guidelines are presented in Table 2.1.

Error Type	Cause	Interaction Design Guideline
Perceptual	Insufficient perceptual cues (for instance, display objects that are visually similar) Mode errors [42]; invisible modes or states of the system Failure to capture user's attention (e.g. visually indistinct important messages, poorly formatted instructions) Lack of perceivable feedback	Elimination of invisible modes Providing salient, visible cues for modes Using coding techniques such as Blinking, Bold, Size, etc.
Cognitive	Taxing the memory and problem-solving capabilities of the human mind Lack of/poor mnemonic aids Inconsistency Lack of context and status information Requiring users to perform mental calculation and translations	Maximising recognition versus recall tasks [42] Providing mnemonic aids Building consistency, rules, and patterns Providing context and status information to orient users Minimising mental calculation and transformations
Motor	Taxing the eye-hand coordination and level of motor skill Highly similar motor sequences Pressure for speed Requiring a high-degree of eye-hand coordination Requiring other types of motor skills such as a high typing skill	Minimizing the need for high skill levels Designing non-similar motor sequences Careful key placement Large targets and clear visual feedback Minimizing the need for typing

Table 2.1 : A summary of error types, their cause, and how they can be alleviated

Mental Models

A mental model is a form of mental representation which people construct in their mind when interacting with the world. In other words, the mental model is an "internal scale-model representation of an external reality" [43]. Human beings use mental models to evaluate and explain the current "state of the world" and predict its future state. According to Norman: "In interacting with the environment, with others and with the artefacts of technology, people form internal, mental models of themselves and of the things with which they are interacting" [44]. According to Mayhew, mental models allow people to hypothesise how "visible parts and processes of a system" relate to its "invisible parts and processes" and thereby to choose a course of action [27].

The idea that people form mental models of systems was first put forward by the Scottish psychologist Kenneth Craik (1943) [45]. He suggested that human beings construct internal models of reality, which allows them to anticipate the future state of events in the world [46]. The concept was elaborated further in 1983 by major contributions of two cognitive scientists; namely, Johnson-Laird [47] and Genter and Stevens [48].

Manketlow and Jullian suggest two reasons why user's mental models should be accounted in human-interface design [49]: first, research in cognitive psychology has shown that users are active problem solvers when interacting with the system. "They form goals and expectations; they make inferences and predictions". This supports the idea that users are not directly interacting with the systems, but rather with their mental model of the system. The second reason is that researchers have realised that users' errors with a system cannot be explained as "random" or as merely due to mental and cognitive limitations. Many errors can be explained more reasonably by the theory of mental model and the fact that users may have an inappropriate mental model of a system "from which they make inappropriate inferences and predictions".

Mental models usually lack technical accuracy; therefore, they are normally simpler than the external entities which they represent [44]. However, this does not prevent people from drawing inferences and predictions from them. This is true, even if the system introduces new functions with which people are not very familiar [49]. Even when they draw incorrect inferences, they modify their mental model to achieve the desired result. This helps them to gain experience which allows them to form possibly more accurate mental models when interacting with similar systems in future. Factors such as user's technical background, previous experience with similar systems, and the nature of human information processing system influence mental models [44].

Norman's makes the following general observations about mental models [44]:

1. Mental models are incomplete;
2. People's ability to "run" their models can be severely limited;
3. Mental models are unstable: People forget the details of the system especially when those details have not been utilized for some period;
4. Mental models do not have firm boundaries: similar devices and operations are often confused;
5. Mental models are unscientific and contain "superstitious" patterns; however, people are unwilling to give up on these simple patterns even though they know there are better ways of doing things. This is because the simple patterns cost little in physical effort and save mental effort;
6. Mental models are parsimonious: users often prefer doing extra physical

actions rather than conducting mental planning which can spare those actions. In other words, people are willing to trade off extra physical actions for reduced mental complexity.

According to Norman there are three models of a system [50]:

- The Design model is the designer's mental model of the system and refers to how the designer conceptualises the system in his mind. Ideally, this conceptualization is based on the user's task, requirements, and capabilities;
- User's Model is the user's mental model of the system or the user's perception of how the system works. This perception cannot be formed direct from the design model, but rather from how the user interprets the system image;
- System image is the way the system presents itself to the user (including the documentations and instructions.

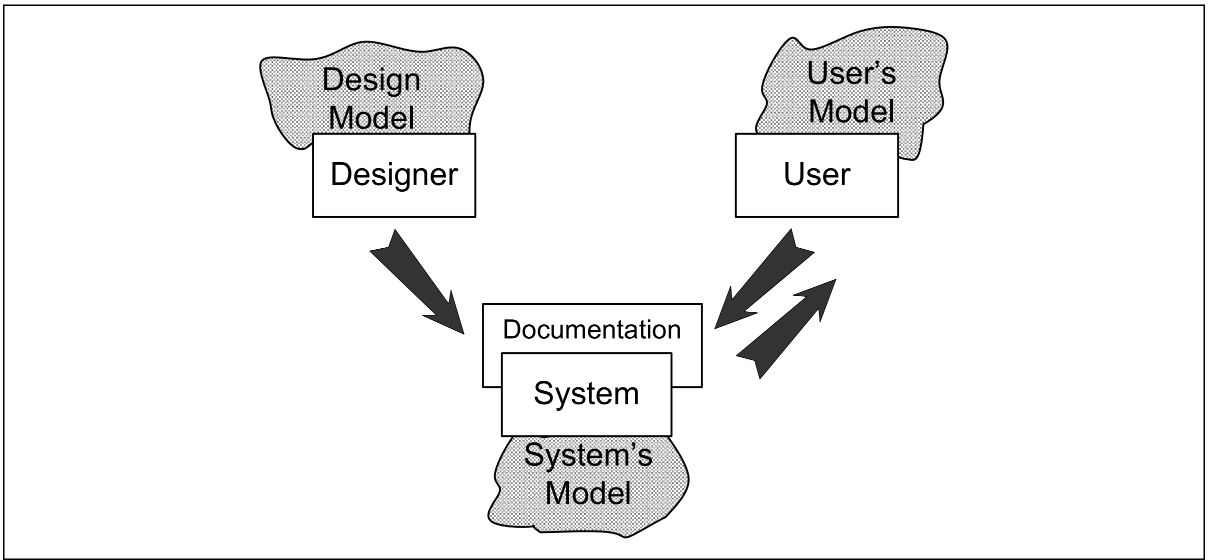


Fig. 2-1: Three models of a system according to Norman [28]

Hence, both designer and user form mental models of the task and the system—although often not matching ones; "It is the nature of this mismatch, which generates many of the problems in system usage" [51]. System image is especially critical because it is the only medium through which the designer and the user communicate. Usability of the system determines how well User's Model and System Image match [43]. Thus, designers must strive to construct systems which develop an appropriate System Image—one which is compatible with the User's Model [44]. To this end, it is always useful to capture and validate the users' models of the system to be designed. Sasse presents a thorough account of this matter by examining different scenarios of how to elicit users' models [52].

Now the question is raised that how designers can ensure that the user acquires an appropriate mental model of a system. The answer gives rise to two important techniques of metaphors and direct manipulation.

2.3 Metaphors

In section 2.2, it was put forth that learning is enhanced by analogy. When users can recognise an analogy between the new information and their prior knowledge, it is easier for them to process the new information and retain it in their long term memory. Human-interaction designers have made effective use of metaphors, which exploit user's prior knowledge of other domains in interaction with systems. According to Booth [53]:

"Metaphors enable learning; they provide short-cuts to understanding complex concepts; they can be used to shape users' behaviour in circumstances that are unfamiliar and that they might otherwise find confusing."

Extensive use of metaphors in a computer user interface has been a commonplace since the advent of GUIs; examples are desktop, folders, trash can, etc. (refer to Schulten's accounts on history of GUI's [15]). Yet, in software systems designed for civil engineers, there is a lot more potential to make use of metaphors from the task domain. As it has been proposed by Chang [54] and later more elaborated by Schulten [15], construction drawings offer an excellent opportunity for use of metaphors in high-performance CAD systems. They can be globally understood by civil engineers without further explanations. The only prerequisite is understanding the rule and legends of standard construction drawings, which civil engineers basically know by education. In this context, metaphors not only facilitate user's learning and understanding of the interface, but also have the potential to eliminate use of local texts and characters.

It must be noted, however, that use of graphics in forms of metaphor, is not always the best solution. In fact, the emphasis made so far must not be taken to underestimate or to discard the role of text in the interface since "one of the great ironies of graphical user interfaces is that they are not usually very graphical" [11]. There are many concepts that could not be expressed without using text and as Johnson says [11]: "The old saying 'a picture is worth a thousand words', while true in some ways, is an oversimplification: Sometimes a few words are worth more than any number of pictures". Furthermore, graphics tend to take up more screen space than text and this may result in cluttered screens in the system. Another problem with graphics is that they may be vague to the user in that the user does not understand the meaning intended by the designer, whereas textual clues can be very precise. Finally, even though users grasp the graphical analogy rapidly, they may draw incorrect conclusions about permissible actions [55]. As a result, metaphors must be carefully used and indeed tested by the intended users to avoid possible shortcomings. Fortunately, there is ample literature discussing the guidelines of metaphor [56, 57].

Finally, metaphors should not be mistaken for icons. Although icons may take the advantage of graphics from the task domain, they are different from metaphors as the latter introduce a deeper conceptualisation to user. The example is the concept of desktop metaphor: Aside from the mere shape of the desktop icon (), this metaphor instructs users to deal with computer files similar to common office documents. In this sense, the desktop metaphor exploits a deeper analogy to facilitate understanding of many different concepts and tasks within the system. In fact, desktop metaphor has proved to be so successful that GUI-based operating systems such as Microsoft Windows or Apple Macintosh are desktop-based.

For high-performance CAD/CAM systems for steel detailing, use of metaphor of construction/technical drawings will potentially help users to understand relevant tasks better and thereby adds to the simplicity of the system. Chapter 5 of this thesis provides a number of new, innovative examples of how basic elements of construction drawings can be effectively used in commands of high-performance CAD systems.

2.4 Direct-Manipulation

According to Mayhew, “A direct manipulation interface is one in which users perform actions directly on objects. This is a decisive contrast to interfaces in which users indirectly specify actions, parameters, and objects through language (for example, command language or menu interfaces)” [27]. A very common example of direct manipulation is reshaping an object in Microsoft Visio, as shown in Fig. 2-2. Here, the user can perform the reshaping command by use of adjustment handles direct on the object.

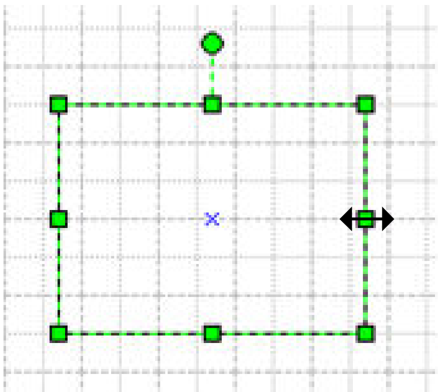


Fig. 2-2: Reshaping a rectangle using direct manipulation in Microsoft Visio

The term direct manipulation was first coined by Shneiderman to describe new interfaces having the following characteristics [58, 59]:

1. Continuous representation of objects;

2. Physical actions or labelled button presses in place of command language;

3. Rapid incremental reversible operations with immediately visible results.

Today, direct manipulation in user interfaces has been realized in a number of ways and through different components of the user interface such as icons, menus, dialogs, etc. Perhaps, its most significant realization for CAD systems is clicking and dragging by aid of adjustment handles, which allows different operations on building objects.

Both research and practice have confirmed various benefits of direct-manipulation. Shneiderman counts several advantages, which may be summarized and paraphrased as follows [55]:

- Visual representation has strong influence on learning speed and retention. Direct manipulation harnesses these strengths resulting in systems whose operation is easy to learn and use.

- Errors are obviously reduced due to visibility of objects and actions of interest as well as elimination of complex syntax. When errors are made, they are easily corrected through reversible actions.

- Reversible actions also foster exploration because the fear of “breaking something” is diminished.

- Direct-manipulation lowers user’s anxiety and reinforces his/her confidence by allowing the user to have the feeling of control and mastery especially because the system responses are predictable and immediate.

Hutchins et al provide a comprehensive definition and discussion of direct-manipulation [60]. According to their definition, there are two aspects to directness, *distance* and *engagement*. They further discuss how direct-manipulation lowers the distance, which by itself divides into articulatory and semantic distance, and enhances engagement.

Mayhew summarizes the advantages to [27]:

- Ease of learning and remembering
- Direct, WYSIWYG (What You See Is What You Get)
- Flexible, easily reversible actions
- Provides context and instant, visual feedback
- Exploits human use of visual-spatial cues
- Less error prone

There can, however, be some disadvantages to this style of interaction [27, 55, 60]: It could be not self-explanatory, inefficient, difficult to design recognizable icons, and icons may take more screen space than words. Regarding the scope of this research, the first two shortcomings are valid mostly to novice or first-time users and, fortunately, not for CAD users, who are already familiar with drawings and graphical symbols used worldwide. The second two shortcomings, on the other hand, are valid regardless of specific application/users.

Altogether, direct manipulation highly adds to directness and intuitiveness of a high-performance CAD system by eliminating commands/menu structures and therefore is very much advisable for an international audience. Moreover, if properly designed, it can also avoid all the abovementioned shortcomings and lead to best efficiency and performance.

2.5 Guidelines of Computer Interface Design

Principles and guidelines of computer interface design are very well-known and established. Several researchers such as Johnson [11], Raskin [21], Mayhew [27], Shneiderman [55], as well as the style guides for specific industry-standard GUI platforms such as Microsoft [61], Macintosh [62], Java look and feel [63], and GNOME [64] provide specific interface design rules. Here, rather than providing these specific design rules, the main guidelines, which constitute the rationale behind many design rules, are presented. Although the eight main principles given here are all according to Johnson [11], they represent guidelines from other sources, too. Furthermore, some examples are provided to show how some of these core principles are violated in existing CAD/CAM programs.

*** Principle 1: Focus on the users and their task not on the technology**

All researchers and official style guides advocate this rule as one of most fundamental ones. This includes understanding users, their tasks (i.e. the intended task domain), their limitations, their preferences and many other aspects that are normally reflected in the user's profile [27]. This process is what Mayhew calls "compatibility" which means the system must be compatible to the user's requirements as well as to other systems [27].

*** Principle 2: Consider function first, presentation later**

Johnson warns that this principle should not be misunderstood as if it meant functionality must precede user-interface in the design process and hence, the designers need to care about functionality first and leave the interface design for later. He adds that this is a common misinterpretation many developers have in design of software systems. To clarify this principle, he says: "software developers should consider the purpose, structure, and function of the user interface—and of the software as a whole—before considering the presentation (the surface appearance) of the user interface. The word 'function' here does not mean 'implementation'—how does it work? It means 'role'—what does it do?" This principle addresses what is known as a "conceptual model" or "the system image". According to Johnson, a conceptual model does not embrace any user-interface components, or screen graphics, but rather "the concepts of the intended users' task domain." For further information on conceptual model, refer to [11, 27].

*** Principle 3: Conform to the user's view of the task**

This principle includes three sub-principles [11]:

- Striving for naturalness

"Natural or Unnatural" are the terms Johnson uses to refer to the tasks which either "belong naturally to the target task domain" or are "artificial and extraneous". Here, task analysis must be carried out to determine whether a function or a feature belongs to the task domain. According to Johnson, software systems must avoid requiring users to "commit unnatural acts because unnatural acts are difficult to learn, easy to forget, time consuming, and annoying." Two examples of unnatural acts in CAD programs are given in sections 5.2 and 5.4.

Another way which a program can violate this principle is by imposing arbitrary restrictions [11]. Examples are:

- in Bocad-3D, file names with upper-case letters are not allowed;

- in Bocad-3D, the undo function for graphical actions is not enabled by default;

- in TeklaStructures, it is not possible to have more than 9 primary windows open simultaneously;

- in TeklaStructures, mirroring/rotating objects is possible only if the mirror plane is perpendicular to "working X-Y plane". In other words, mirroring relative to an arbitrary plane, which is not perpendicular to "working plane", does not lead to any desirable results.

Another factor that increases naturalness is using the user's terminology of the task, which means the system must be designed as to incorporate as much as possible the terms that users apply for their intended task domain [11]. This rule is very often violated in Bocad-3D when the program provides a feedback on an error during an operation. Examples from the author's real life experience with the program are shown in Fig. 2-3. Obviously, a typical user can never be expected to understand such messages nor should he/she be expected to consult a manual or contact the hotline to find out the meaning.

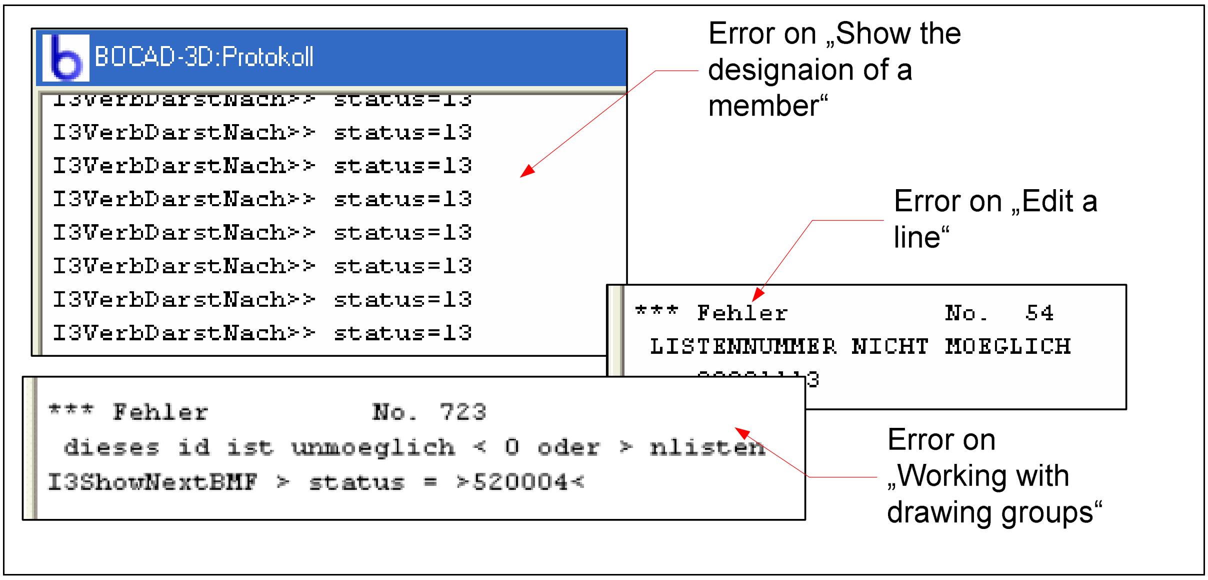


Fig. 2-3: Inadequate "jargon-like" feedback in Bocad-3D

- Use the program internal inside the program (not in the UI)

The program must present users only the terms and presentations which comply with the task domain not from the program internal. The examples shown in Fig. 2-3 violate this rule as well.

- Find the correct point in the power/complexity trade off

This is discussed later in section 2.6.

*** Principle 4: Don't complicate the users' task**

This principle, also known as simplicity, advocates that common tasks within the program must be easy regardless of how complicated a program might be. In Raskin's words [21]: "No matter how complex the overall system, there is no excuse for not keeping simple tasks simple."

Users' task will be complicated if users are given "extra problems" to solve when working with the program [11]. The "extra problems" are those which pertain primarily to the program (or computer technology) and are not from the intended task domain. The more users have to spend time struggling with the program, in the hope of getting their tasks done, the more they are likely to be distracted from their basic goal; i.e. solving the problems of the task domain by using the program [11]. An example of such "extra problems" in TeklaStructures is seen for the commands "to create a weld", for which the user has to assign axis. On the other hand, in Bocad-3D, all the struggles which are considered to be "part of the job" of working with the program are of the nature of "extra problems".

Given the low capacity of working memory and other cognitive resources, such distractions in the task domain of steel detailing, which on its own taxes the memory by demanding a lot of attention, may come at a very high cost for companies, i.e. paying for liabilities as a result of human error. This makes it crucial for high-performance CAD programs to eliminate or minimise the need for spending time on problem solving in the program specific domain.

*** Principle 5: Promote Learning**

Learning or extent of learnability in a program is also addressed as one of the main attributes of usability. The more time users have to spend learning the program, the later they achieve the desired productivity with the program. Therefore, the program must promote learning also because it has a direct influence on user's satisfaction. Factors that promote learning are consistency, familiarity, simplicity, direct manipulation, WYSIWYG (What You See Is What You Get) [11, 27, 55].

Consistency means that similar actions and operation within the program must have similar or if possible the same sequence of action as well as the interface [11, 27]. Consistency fosters development of habits, which allows users to interact unconsciously with the program while focusing on their job. The existing style guides could be used to realise graphical consistency within a program. However, to ensure consistency within the functions, task analysis and usability testing are required. An example in Bocad-3D where the latter form is violated is the “mirror” command. If the user wants to mirror points, he/she has to select the mirror axis/plane and then the objects of interest; whereas, for mirroring members, the sequence of object selection differs; i.e. first the objects should be selected and then the mirror axis/plane. Another example is the command of deleting objects which is object dependent. For this command, the user has to search and select different “delete” commands for different objects of interest such as dimensions, members, points, welds, etc. The consistent design would be to use the keyboard “delete” button as well as assigning one delete command which can function simultaneously with different object snaps.

Another factor which encourages learning within the system is a “low-risk environment” [11], or as Mayhew calls it “robustness” and “protection” [27]. A program which provides a low-risk environment makes it hard for users to make mistakes, and when they do, makes it easy and fast for them to recover from their mistakes. Such a program evidently encourages exploration. In contrast, in a high-risk environment users “tend to stick to their familiar, safe paths”, and this way learning is indeed impeded [11].

*** Principle 6: Deliver information, not just data**

Johnson emphasises the difference between the data and the information on the screen and advises that the interface must present the user the information not just data because it is the information that user needs to know. Therefore, it is wrong to present data on the screen and leave it to the user to find out the information in it on his/her own. In this regard, one of the characteristics of a good user-interface is that is does not just present the data but rather “directs user’s attention to the important information.”

*** Principle 7: Design for responsiveness**

By responsiveness one means how well the system can keep up with the user’s pace of operations within the system [11]. Bickford, by summarising results of many studies on users, concludes that the speed in a program is a factor that most influences users’ satisfaction so much that it precedes other factors such as compatibility, user interface, or price [65]: “when it comes to computers, users hate waiting more than they like anything else.” He further distinguishes the “real (machine) speed” from “the perceived speed” and states that the perceived speed is what actually matters because it addresses how the users will experience the response lag. If they are offered with adequate feedback while the program is processing their tasks, they will have the positive impression as if they did not have to wait long. On the other hand, if the program deprives the user from a feedback of ongoing processes, users will inevitably find it slow.

*** Principle 8: Test it out on users, then fix it**
Even the most careful adherence to guidelines cannot guarantee that the design will be successful. The only way to ensure this is by testing the design on users, receiving their feedback and modify the design (iterative design process).

Aside from these eight principles, Shneiderman speaks of two principles in his "eight golden rules of interface design"[55], six of which are implied in abovementioned eight principles. According to Shneiderman, the system should "enable frequent users to use shortcuts". He reasons this by pointing out that frequent users need to lower the extent of interaction, so that they can increase the pace of interaction. Hence, the system should provide abbreviations, functions keys, hidden commands, and macro facilities. It should be noted that hidden commands here are *alternative* faster routes not the main path that most users, (including novices), are expected to use. Finally Shneiderman emphasises on control and explains that users, especially experienced users, must have the sense of mastery and control over the system. Therefore, systems must be designed so that the users are "the initiator of actions rather than the responders".

2.6 Tradeoffs

Tradeoffs are in the inherent nature of all designs. Similar to its concept in civil engineering, design of user interfaces is a series of tradeoffs. The reason is that many design requirements are not only intertwined but also turn out to conflict each other; for example, offering the user a maximum control impedes learning; simplicity may sacrifice power [27]; presenting extra help, although interesting for novices, frustrates experts; displaying more information as well as enhanced graphic, although informative, lowers the response time [28].

In short, the optimal choice for one part of the problem may not be optimal to another. This makes the task of UI design indeed very challenging. It is the designer's task to make the hard decision and find the optimal point of trade-off between different features. Norman (1986) makes the following remark about this issue [28]:

"It might be useful to point out that although there may not be any best solution to a problem in which the need of different parts conflict, there is a worse solution. And even if no design is "best" along all dimensions, some designers are clearly better than other—along all dimensions. It clearly is possible to design a bad system. Equally, it is possible to avoid bad design."

Here, two famous tradeoffs of "information versus time" and "power versus complexity" are shortly introduced. For further insight refer to [11, 28, 66].

Information versus time [66]:
This is a basic trade-off and one of the major interface issues which must be

handled: "Factors that increase informativeness tend to decrease the amount of available workspace and system responsiveness." On the other hand, the more informative and complete the display, the more useful when the user has doubts or lacks understanding. As it was mentioned before, what helps novices tends to annoy and impede skilled users.

Power versus Complexity [11]:

This addresses the usual trade-off, which exists between power and complexity. This happens because for every feature, function or capability in a computer-based product or service, there must be a way for users to invoke or control it.

There are many instances in Bocad-3D where it seems that the designer has not been able to find the correct point on this trade-off. A very simple (yet interesting) example is shown in Fig. 2-4 for the command of "Adding a point". Obviously, there should not be anything especial with this very basic command. Corresponding task analysis seems to be very straight forward: The user wants to generate a new point via relative dimensions (x,y,z) to an existing point. As the figure shows, there are at least three redundant features which do not belong to the task domain and are very seldom used. (It should be noted that in this figure, the sign () does not belong to the interface of Bocad-3D. This comes from the software (Hardcopy) [67], which was used in this thesis to generate high quality screenshots).

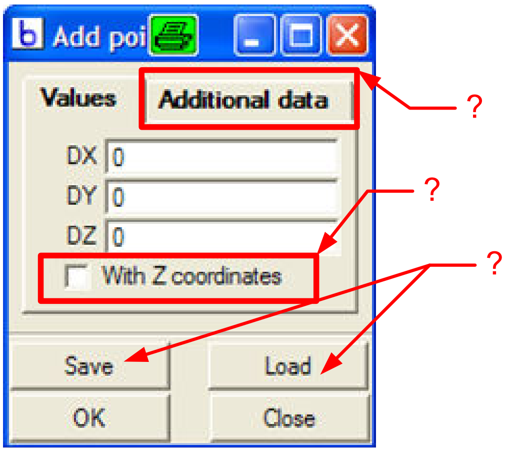


Fig. 2-4: Dialog box in Bocad-3D for adding points relative to an existing point

In Johnson's words [11]: "Computer programmers tend to believe 'the more options, the more controls, the more power, the better'. In contrast, most people who use computer products and services want just enough power or functionality to do their work—no more, no less. …The mere presence of less important features can make more important ones harder to find and use, causing common and simpler tasks to become harder."

In order to deal with complexities, Johnson proposes "sensible defaults, template or canned solutions, progressive disclosure, and generic commands". Here, progressive means "hide detail and complexity until the user needs it". This rule is very practical in high-performance CAD/CAM systems since there are normally a lot of details to deal with, especially in program's macros. An example of such complexities is shown in Fig. 2-5 for a "haunch connection" in both Bocad-3D and TeklaStructures. Both programs effectively take an advantage of metaphor of construction drawings, which obviously enhances efficiency. However, it is apparent that Bocad-3D presents much more information per screen. In this case, all the information pertaining to bolts, welds, and much of the connection geometry is given at the front level. This violates the rule of "progressive disclosure".

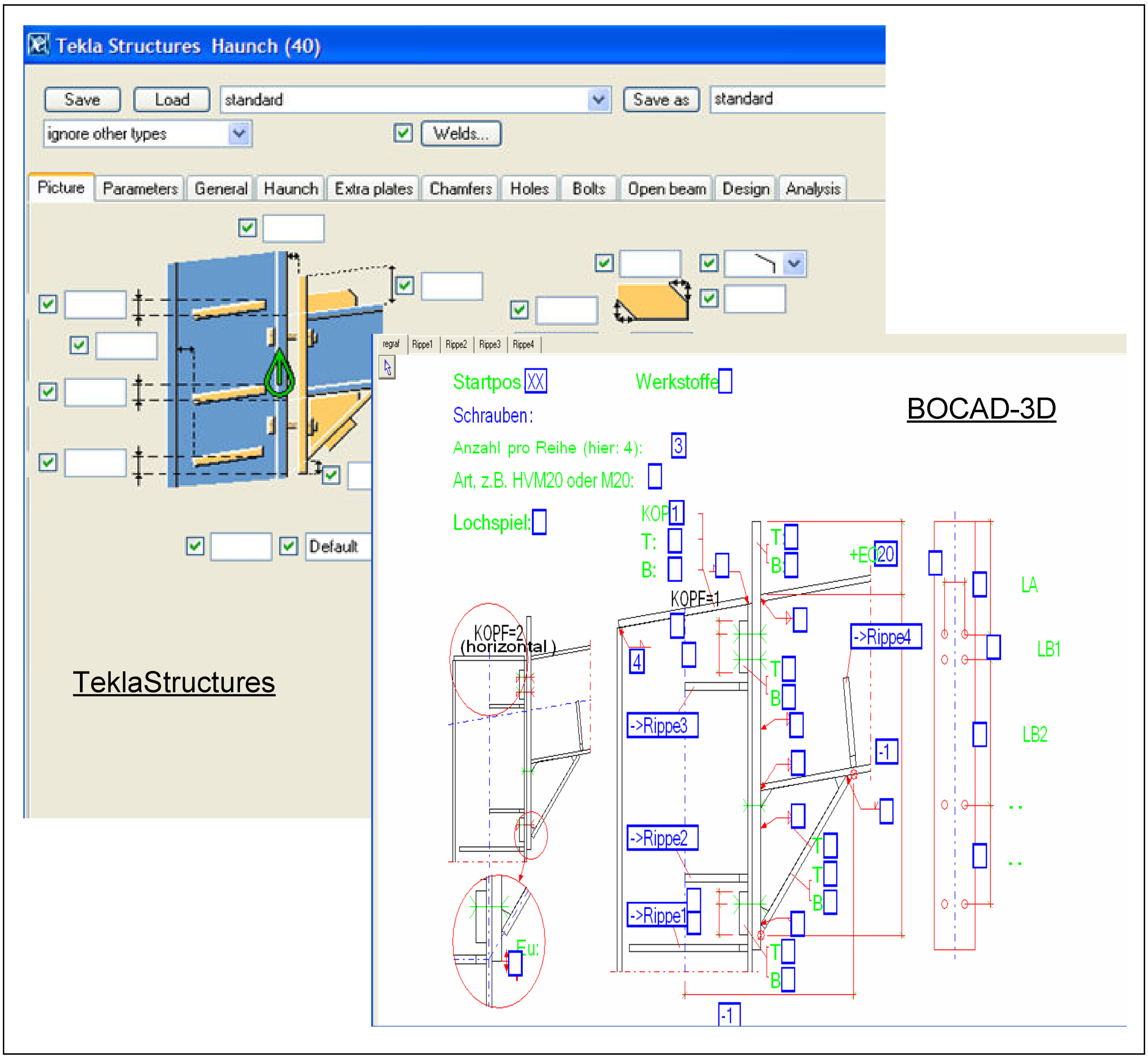


Fig. 2-5: The macro command for "haunch connection"

Apparently, Bocad-3D believes this to be the choice of expert users because presenting information at deeper layers consumes more time. Although this is true in general, it holds mostly for frequent routine tasks, such as "View/Display Settings" (See section 5.1), where learning and recall are facilitated by regular iterations; in such cases, presenting information in different layers is definitely time consuming. However, given the complexity and variety of connections in steel structures, users may often need to apply a different macro each time. In other words, “individual” connections or macros cannot be categorised as frequent commands, as the results of the survey in Chapter 3 also indicate. Therefore, this type of presenting information—showing so many choices per screen—in macros increases the “acquisition time” as well as the error rate, as shown by empirical studies such as Miller’s [68] (“Acquisition time” is the time users need to find the subject of interest). In contrast to Bocad-3D, TeklaStructures presents the same information in an organised, structured fashion, which facilitates learning—as it was mentioned in section 2.2. Therefore, the approach taken by TeklaStructures will be found easier to use because it provides a step by step solution.

3. Research Method

In order to propose user interface solutions for steel detailing CAD systems in developing countries, two steps must be taken: first, to develop a user's profile of corresponding users in developing countries and second, to carry out a task analysis of commands in high-performance CAD/CAM systems.

3.1 User Profile of CAD/CAM engineers in developing countries

The role of a user profile is to collect all relevant information about the users of a system, which can facilitate objective, informed decisions about the different aspects of the user interface. There are four major aspects to a user profile according to Mayhew [27]:

- Psychological Characteristics;
- Knowledge and Experience;
- Job and Task Characteristics;
- Physical Characteristics.

Each of these aspects is decomposed to sub-characteristics. While many of these characteristics tend to be similar among users in Europe and in developing countries, some of them mark the decisive which will be of primary importance within the scope of this research. Here, the discussion is narrowed down to the specific case of Iranian users.

Among the characteristics listed above, physical characteristics seem to be the least relevant in the present investigation. For other characteristics, Mayhew [27] considers several sub-characteristics as shown in Fig. 3-1.

*** Psychological Characteristics**

Certain psychological characteristics such as motivation and attitude affect user's performance [27]. Although the designer of a CAD interactive system cannot control the attitudes or the level of motivation that users in developing countries bring to the system, CAD systems can be designed to minimize the negative emotions such as frustration and boredom and maximise the positive emotions, which lead to job satisfaction as well as enhanced motivation and attitude [27]. In case of Iranian users, these requirements call for simplicity, strong feedback, and low risk environment. Moreover, since they are mandatory users, they need to immediately experience some benefit from using the system. On the other hand, since they lack practical experience of steel detailing because of very theoretical university education in developing countries (see section 6.3), the system should be consistent, predictable, simple to understand and operate; it must also give them immediate sense of control, mastery, and achievement [27].

Psychological Characteristics		
Attitude	**Motivation**	
Negative to neutral	Moderate to high	
Knowledge and Experience		
Education	**System experience**	**Task experience**
University graduate	Novice	Novice
Application experience	**Native Language**	**Use of other languages**
Previous experience with AutoCAD Primary OS: Microsoft Windows Previous experience with MS Office	Persian (Farsi)	Intermediate level of English No familiarity with other European languages
Job and Task Characteristics		
Frequency of use	**System use**	**Primary training**
High	Mandatory	Mandatory formal
Other tools	**Parallel use of other programs**	**Task importance**
Telephone, Calculator Other programs	AutoCAD, Adobe	High

Fig. 3-1: User profile of CAD/CAM engineers in Iran

Although Iranian users may have moderate to high motivation for using the system, it is likely that their trust of the system is fragile. This means, errors (including system bugs), unexpected responses from the system, frequent undesirable results, hidden effects coupled with ill-structured feedback, inefficient use of basic commands, high-risk environment, etc. will negatively affect their job satisfaction and undermine their motivation in the system use.

Designers must not forget the differences in executive intentions behind the application of high-level CAD/CAM systems in German enterprises in comparison to that in Iranian enterprises. Use of such advanced systems in Germany, as a highly developed country, provides the only survival chance in the market to obtain flawless results in due time. In other words, the enterprise which lacks these tools is definitely off the competitive edge. However, in Iran, as a developing country, use of such systems tends to be more optional than mandatory. The customer does not expect error-free results in time, but rather a low price. In short, for German enterprise, using these systems is the make or break factor, whereas for Iranian enterprises it tends to be a choice between “good and better”. Hence, Iranian enterprises will typically approach these systems first as to try them and examine if these systems can offer better functionality than otherwise and equally important, if they can conveniently use this functionality. This is why usability concerns play an extra significant role when designing a CAD system for Iran (or developing countries alike). This marks one difference concerning attitude between Iranian and German users: Iranian users are likely to approach the system with moderate to high motivation, albeit with sceptic to neutral attitude towards the system. Thus, extra attention must be given to design systems, which allow control, promote learning and provide low-risk environment.

*** Knowledge and experience**

Within the scope of this research, it is assumed that Iranian CAD/CAM users are typical university graduates of Civil Engineering, although technical drafters may also be assigned to this job, which is not actually advisable (See discussions in section 6.3).

Task experience refers to the knowledge of the task domain [27], which is steel detailing in this case. Graduates of Civil Engineering departments in Iran are typically novices (inexperienced) concerning steel detailing know-how (see section 6.3).

System experience refers to user's experience with the particular system [27]. Normally, graduates of civil engineering in Iran do not have any previous experience with high-performance CAD systems.

Application experience refers to user's previous experience with similar systems [27]. This is indeed the point which has impact on mental models of Iranian users. Nearly all Iranian CAD/CAM engineers have previous experience with AutoCAD. Moreover, as an Operating System, they all use Microsoft Windows. Each of these prior experiences should provide designers with specific clue about how the system image should be. The prior experience with the operating system determines primarily the specific "look and feel" of the system. That means, for Iranian users, it is far more helpful to develop systems, which behave like Microsoft Windows. In this case, the XP version is preferable, since it is widespread in Iran.

Prior experience with AutoCAD causes that users form some certain expectations from similar programs. Most fundamental ones are zooming and panning, object-verb versus verb-object paradigm, object snap-settings, dimensions or use of keyboard. These are all characteristics, which Iranian users expect to see in other CAD programs. Therefore, the more a new system behaves more or less the same, the more it is likely to leave a positive impression already from early encounters. On the other hand, AutoCAD is one of the basic CAD systems and even companies working primarily with high-performance CAD systems such as Bocad-3D or TeklaStructures need to use AutoCAD. Hence, the more compatibility in the user interfaces across systems, the less learning burden there will be for the new system and the fewer the errors that will be made as users go back and forth between systems. Everyone who works with Bocad-3D certainly knows the strange experience of switching to AutoCAD in between; unless the user concentrates, he/she pans when he/she actually wants to zoom; he/she deletes the objects when he/she actually wants to discard the selection!

Finally, as to the subject of program language, it should be noted that since software systems with English user interfaces (including MS windows) are very prevalent in Iran, it seems that Iranians are used to English as the program language; hence, translation and change of the program character set from English to Farsi does not seem to be necessary.

*** Job and task**

A very important aspect in this category is the learning habit of Iranian users in their interaction with computer systems. Although it is possible to organise mandatory training periods (and it is actually advisable according to section 6.8), one must not underestimate the importance of self-learning and training on the job. This type of learning actually constitutes basic learning habits of Iranian users when interacting with programs.

Iranian CAD/CAM users have a high tendency to explore within the program and learn it by try and error. Reading manuals and instructions as supplementary material is also common as long as the manual is informative and carefully organised. Thus, a software system designed for Iranian users must be highly explorable by encouraging users to explore the interface. Of course, it should be noted that this type of learning, i.e. learning by exploration and try and error, is not something to be limited to Iranian users. This is recognised as a general rule among researchers and has been confirmed by many studies and surveys; for example, Nielson reports of a survey of business professionals who were experienced computer users. It was found out that 4 of 6 highest-rated usability characteristics (out of 21 characteristics in the survey) related to exploratory learning: easy-to-understand error message, possible to do useful work with program before having learned all of it, availability of undo, and confirming question before execution of risky commands [9]. Therefore, exploratory learning is not only an important rule in general, but also has a significant effect given the learning habit of Iranians.

To be explorable by users, a system must meet certain prerequisites such as:

- clear explicit feedback presenting the user all effects which might be hidden from the user, or which the user is likely to overlook;
- helpful warning and error messages whose purpose is not only mere warning but also guidance of the user to recover from errors;
- effective undo/redo function so that the user can easily recover from undesired effects;
- reliability and system stability.

In short, the designers must design the system such that it relieves the user from the unpleasant feeling of “breaking something”.

3.2 Analysis of Commands in High-performance CAD/CAM System

As discussed in Chapter 2, task analysis is the second fundamental step for user-centred design. The method used in this research, is perhaps the most effective one in order to capture the needs and day-to-day tasks of engineers working with high-performance CAD systems for steel detailing. For this purpose and as a part of this research, the author participated in real life projects as a project engineer at engineering company ICW GmbH for one year. He had active participation in projects such as: European Court of Justice in Luxemburg, Porsche Museum in Stuttgart, football stadium in Zurich, and Fiddlers Ferry Power Station in Warrington, England, which were all done using high-performance CAD/CAM systems to be transferred to developing countries. This valuable experience gave the author a thorough insight and knowledge of the task domain as well as the requirements that CAD engineers must fulfil in their daily working life. In this regard, a survey was also carried out in this company with the target program Bocad-3D. The objective of this survey was to obtain reliable statistical data about the frequency distribution of commands in a high-performance CAD system and thereby to prioritise the efforts necessary to develop a text-free graphical user interface for international use.

Survey Method

The survey was conducted within 17 working days in February 2006. The sampling method used was simple random sampling (SRS). Simple random sampling is drawing of a sample from a sampling frame in which (a) the units in the sample have been drawn independently from each other and (b) each unit in the sample has been drawn with the same probability [69]. In a SRS, members of the target population are selected one at a time and independently and because of this equality of opportunity, random samples are considered relatively unbiased [70].

In this survey, the user and the observation time were selected for each observation session by using random numbers. Time was divided into five-minute intervals during typical office time (8:00-18:00). In order to disturb least with their work, it was decided to observe each user not more than three times a day and each time with user’s consent. If at any specific session the user was either not at his desk or not working with the program, the sample would be excluded from that very session and no replacement would be considered for that day. At each sampling session, the user’s work was video-taped for 5 minutes to be analysed later on. All together 445 minutes (89 observations) were recorded from 7 working engineers. Since it was not feasible at all time to focus the camera fast enough to recognise the commands, a very quiet audio commentary was integrated within observations.

Survey Results

Within 445 minutes of observation 5455 samples (commands) were collected which comprised 95 types of commands. In order to form a better understanding of the user's task, it was decided to analyse the user's actions from three different perspectives, which are explained below.

User's actions based on general working/task modes

User's actions at a high-performance CAD system comprise basically two working modes; i.e. modelling and generation of output (drawings, parts lists, and CNC-data). In other words, users of an advanced CAD system spend time either constructing the 3-D product-model or controlling the output generated automatically. Thus, "Modelling" actually covers all the actions and commands to accomplish the building model and develop the model to the state ready for automatic generation of parts lists, CNC data, and drawings. "Output", on the other hand, is the mode in which the user modifies or completes the output. In this survey, the user was working in "modelling mode" at 73% of observations, and the rest in the "output mode", as shown in Fig. 3-2.

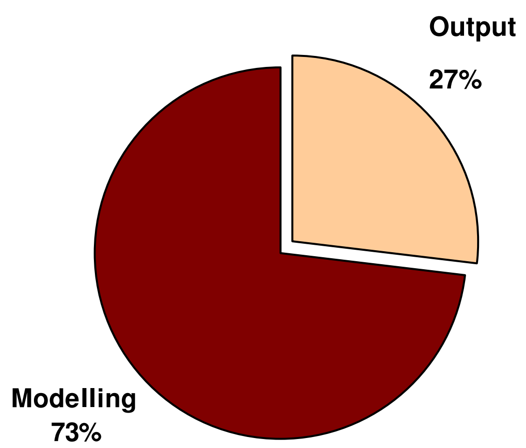


Fig. 3-2: Observed distribution of user's work based on working modes

User's actions based on specific functionality

To enable users to accomplish their two basic tasks, CAD systems provide various commands offering different functionalities. In this survey, 95 different command types in Bocad-3D were observed. These commands can be divided into 8 different categories based on their similar functionalities. Each category may as well include different command types or sub-categories. This logic structure of the research is shown in Table 3.1. These eight categories were defined as follows:

- **Construction:** All commands during the modelling phase, by which the user creates or edits building components such as profile members, plates, bolts and welds. In other words, this category of commands offers the possibility to create or modify the components of the product model. This category is composed of four different sub-categories: Edit, Copy/move, Connection, and Creating members.

Construction		**411**	**Graphic**		**445**
Edit		**151**	**Dimensioning**		**174**
Contour-edit		58	Dimension		111
Cut Off		29	Dim-move		26
Adjust		26	Dim-del		21
Member property Edit		17	Dim-edit		14
Adjust on Member		10	Dim-bolts		2
Punch		6	**Text**		**133**
Stretch		4	Textbox-move		92
Lengthen		1	Text-edit		15
Copy/Move		**103**	Text		17
Copy via points		54	Textbox-edit		9
Move via dimension		27	**Copy/Move**		**61**
Move via points		12	Delete		30
Mirror		8	Move		19
Copy via dimension		2	Copy		9
Connection		**68**	Rotate		1
Bolting		33	**View**		**54**
Delete Connection		12	View-move		26
Bolt-edit		7	Open-group		12
Welding		10	View-copy		5
Others		6	View-delete		3
Members		**63**	Others		8
Delete		23	**Drawing**		**23**
Macro	-	22	Line offset		13
Create a profile		18	Line		4
			Rectangle		5
			Circle		1

Table 3.1: Categories of commands and their frequency

Points		**322**	**View**		**235**
Intersection		90	View-section		65
Mid-points		59	Display settings		67
Member-points		30	View- member		71
Bolt-points		25	View-Axis		10
Add-points		31	Display Welds		7
Point parallel		20	View-3D		6
Perpendicular		20	View-window		4
Delete		17	Others		5
Point-extend		11	**File**		**24**
Others		19	Open-drawing	-	10
Info/Check		**200**	Save-drawing		8
Points distance		63	Save		2
Members-info		51	Browse drawings		3
Search		33	Print		1
View Assembly		18	**Output**		**9**
Member ID Check		11	Generate-drawings	-	7
Blink	-	8	Generate-lists		2
Bolt-info		7	**Navigation**		**3683**
Others		9	Zooming	-	3681
			3D-rotate		2

Table 3.1. Continued

- **Graphic:** All commands by which the user creates or modifies graphical elements—complementing the standard output of a CAD system. Thus, these commands basically have no effect on the building product model but are needed to modify the drawings in a way individual clients demand. This category is divided into five sub-categories: Dimensioning, Text, graphical Copy/Move, View, and 2D drawing.

- **Points:** This category includes all commands that user needs to create or recognise reference points such as end points or mid points. Such points are typically needed during the construction of the model.

- **View:** All commands that user applies in order to change the display settings of elements or to have additional views.

- **Info/Check:** During the construction of the model, users often need to check some information about the components. Commands such as member info, search, bolt-info etc. fall into this category.

- **File:** This category addresses all commands relating to management of the program specific files or documents.

- **Output:** Commands that are used to generate automatic output such as parts lists, drawings, and CNC-data.

- **Navigation**: Commands which enable the user to navigate within the model. According to results, this category mostly consists of zooming.

User's actions based on productivity

The eight categories can be abstracted into two more general categories, i.e. productive commands and preparatory commands:

Productive Commands: These are defined as commands which basically enable the user either to construct the model or to generate output, i.e. drawings or lists. Therefore, this category mostly comprises main categories of Construction, Graphic, and Output.

Preparatory Commands: These are defined as commands which the user carries out to reach the mental state in which he/she can be sure of right results before executing the next productive command. These are called "preparatory", because they basically set the stage for the state of execution; a typical example of such preparation will be: gathering the right information, setting the right view, changing the display settings, checking the current state of the model etc. Therefore, this category mostly comprises main categories of Points, View, Info/Check, File, and Navigation.

The classification of commands based on working mode, functionality and productivity has two major advantages: (a) It paves the way for a better and more comprehensive understanding of the user's actions and needs; (b) It allows comparative studies among different systems such as Bocad-3D and TeklaStructures. In other words, results of a similar survey on TeklaStructures can be compared with this survey, since both systems are similar at least in terms of user's tasks and productivity although with differences in specific functionalities, i.e. commands.

Survey Analysis

By collecting and sorting the data and summing up the number of repetition of each command, the frequency of each command will be obtained. Relative frequencies (proportions) are calculated which are more helpful for drawing comparisons. A proportion is the number of observation or responses with a given characteristic divided by the total number of observations [71]. For the purpose of this study, ranking of commands in terms of their frequency is of importance. Scientific questions are how reliable are the results or how similar will be the results of another survey using SRS or how much are these results repeatable? The answer could be achieved through calculating the confidence interval for the population parameter.

Confidence interval is a plausible range, which with a certain level of confidence includes the unknown population parameter (in this case each command's frequency). The width of the confidence interval gives us some idea about how certain we are about the unknown parameter. A very wide interval indicates that more data must be collected before anything very definite can be said about the parameter. Theoretically, the true value in each case is a mean of an infinite number of test results, but since it is not feasible to carry out so many tests, confidence intervals are calculated instead to indicate the level of certainty about the range which includes the true value [72]. For example, a confidence interval of 95% indicates that if independent samples are taken repeatedly from the same population, and a confidence interval is calculated for each sample, then 95% of the intervals will include the unknown population parameter.

Considering the sampling method used, the following assumptions are made:

- All commands are treated as independent random variables;
- All commands are assumed to have Bernoulli distribution since they have "yes/no" or "observed/not observed" nature. In other words, at each particular observation, each individual command is either observed with the probability or proportion of $\hat{p}$ or not observed with the proportion of $(1-\hat{p})$ [69, 73].

Based on central limit theorem, the sum of a sufficiently large number of independent random variables has a distribution that is approximately normal [69]. By "sufficiently large", one means a sample of size n>30 [69, 73] and sample sizes larger than 100 yield sufficiently accurate results [72]. Therefore, even though the commands are assumed to have Bernoulli distribution in this study, it is possible to calculate the confidence intervals same as normal distribution. Thus, considering the variables to have a Bernoulli distribution and the confidence level of 95%, the Equation 3-1 yields the confidence interval. A more detailed account of this formula is given in Appendix A.

$$CI = (L,U) = \left(\hat{p} - 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{n}} \right) \qquad \text{Equation 3-1}$$

in which: CI is the Confidence Interval; L is the Lower limit of CI; U is the Upper limit of CI; $\hat{p}$ is the Calculated proportion or probability and n is the sample size.

Results based on specific functionality (commands)

Table 3.2 presents the relative frequencies as well as confidence intervals for the eight main categories. As it is shown in Fig. 3-3, “navigation” has a substantial higher frequency than other commands. This is because in Bocad-3D, despite existence of a “Panning” function, users do not use it. The reason is, the pan function is neither direct nor effective and does not comply with common software such as Microsoft Word/Excel or AutoCAD. As a result, frequent zooming is used to compensate this shortcoming. In order to have more balanced results, it was decided to exclude navigation (zooming) for the sake of analysis. Results are given in Table 3.3 and presented in Fig. 3-4.

	Occurrences	Lower Limit %	$\hat{p}$ %	Upper Limit %	(U-L) %
Navigation	*3777*	*68,01*	*69,24*	*70,46*	*2,45*
Graphic	*469*	*7,85*	*8,60*	*9,34*	*1,49*
Construction	*411*	*6,83*	*7,53*	*8,23*	*1,40*
Points	*325*	*5,33*	*5,96*	*6,59*	*1,26*
View	*235*	*3,77*	*4,31*	*4,85*	*1,08*
Info+check	*200*	*3,17*	*3,67*	*4,17*	*1,00*
File	*28*	*0,32*	*0,51*	*0,70*	*0,38*
Output	*10*	*0,07*	*0,18*	*0,30*	*0,23*
	5455		*100*		

Table 3.2**:** Relative frequencies and confidence intervals for command categories including navigation

Calculation of confidence intervals, as in Table 3.3, allows a better interpretation of results. For instance, for the category of Graphic, it is possible to conclude that with certainty of 95%, this category includes 26 to 30% of all commands excluding navigation and in this particular series of observations it is 28%. Likewise, the results further reveal that for most categories, their occurrence probability lies in an acceptable range with certainty of 95% and therefore, it can be concluded that enough samples have been collected.

	Occurrences	Lower Limit %	$\hat{p}$ %	Upper Limit %	(U-L) %
Graphic	*469*	*25,8*	*27,9*	*30,1*	*4,3*
Construction	*411*	*22,4*	*24,5*	*26,6*	*4,1*
Points	*325*	*17,5*	*19,4*	*21,3*	*3,8*
View	*235*	*12,3*	*14,0*	*15,7*	*3,3*
Info+check	*200*	*10,4*	*11,9*	*13,5*	*3,1*
File	*28*	*1,1*	*1,7*	*2,3*	*1,2*
Output	*10*	*0,2*	*0,6*	*1,0*	*0,7*
	1678		*27,9*		

Table 3.3: Calculated relative frequency values as well as confidence intervals for command categories excluding navigation

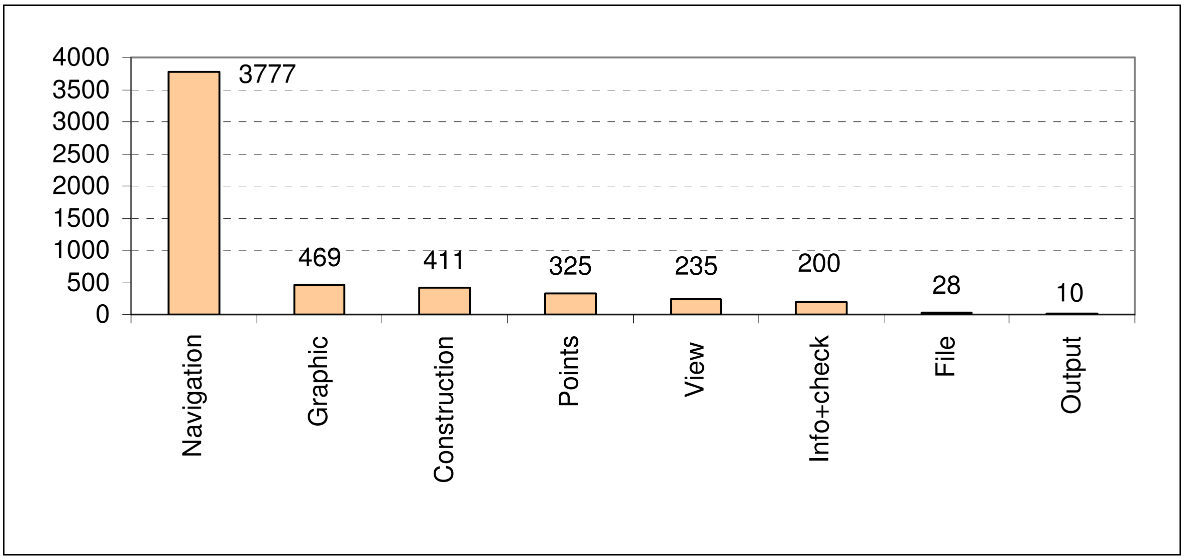


Fig. 3-3: Number of occurrences including navigation (zooming)

Fig. 3-5 and Fig. 3-6 show the distribution of commands as well as their relative frequency in sub-categories of construction and graphic, respectively. Confidence intervals were calculated also for each category/sub-category. Selected results are presented for five main commands' categories in Table 3.4.

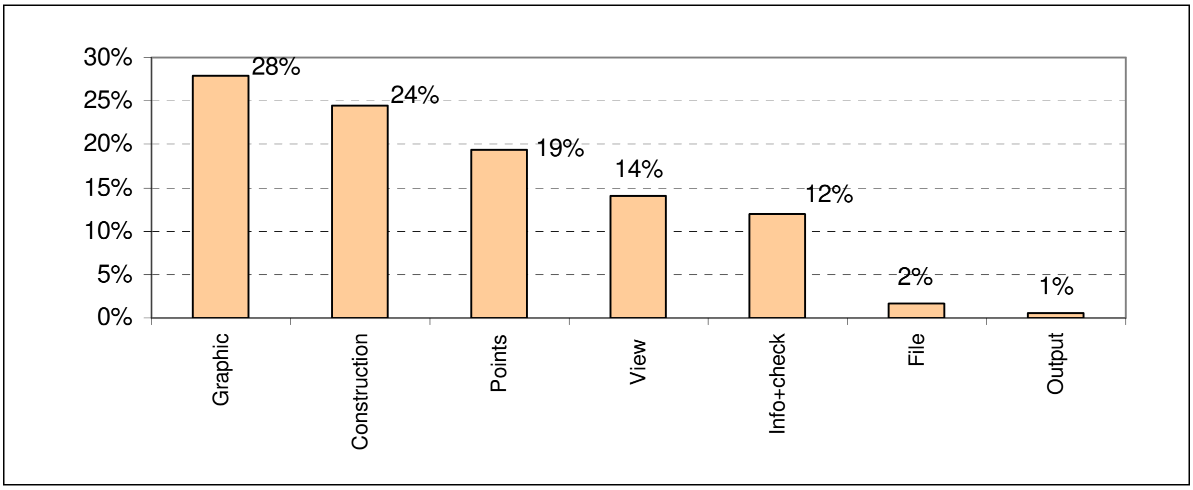


Fig. 3-4: Distribution of command categories excluding Navigation

Results based on "Productivity" as well as "working modes"

In this section, frequency of commands' categories in both working modes is investigated. To do this, observation sessions were first classified based on their working mode—similar to the results presented in section 3.1. Afterwards, frequency of different command categories was considered in each working mode.

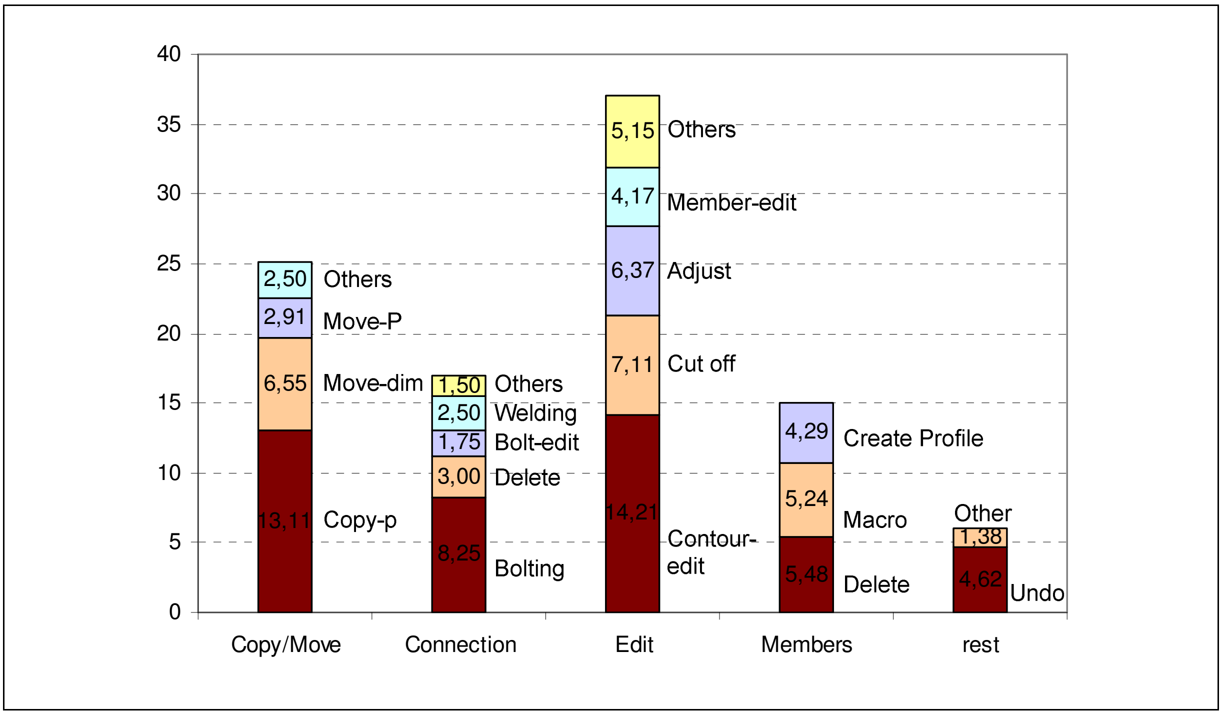


Fig. 3-5: Category of Construction: Sub-categories with their most frequent commands

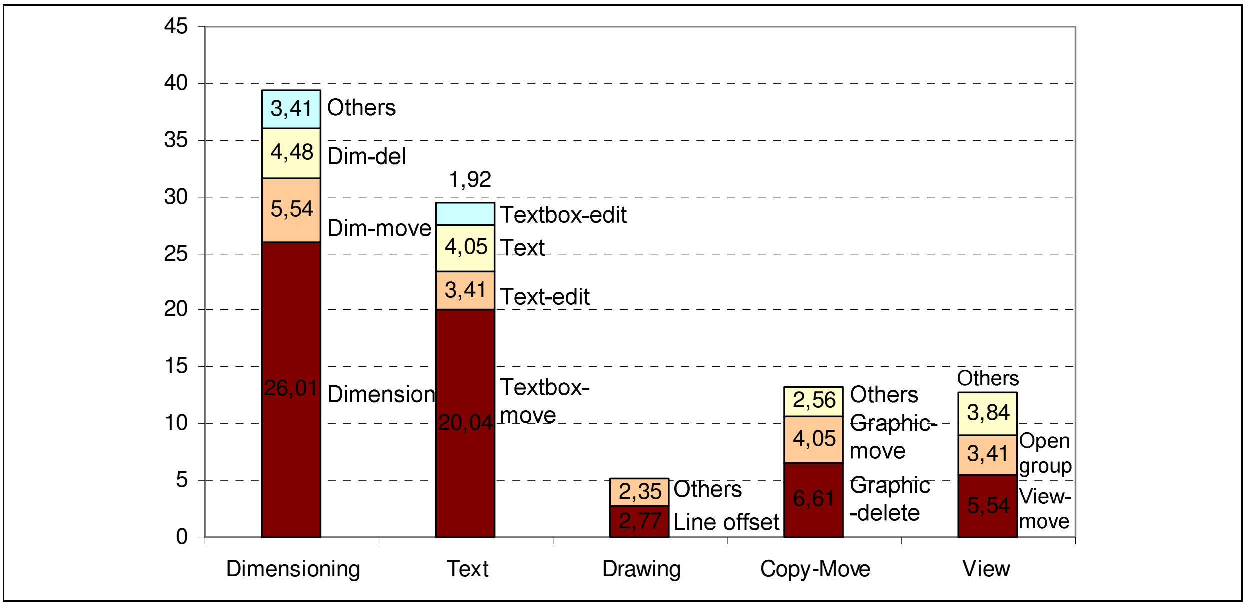


Fig. 3-6: Category of Graphic: Sub-categories with their most frequent commands

Table 3.5 presents values of means and standard deviations of each command category in both working mode. The tabular values represent the observed data in terms of location or central tendency as well as dispersion or variability [69]. Here again, in order to have an estimate of accuracy and repeatability of the data, the confidence intervals were calculated for means, using the properties of a unit normal distribution (Equation 3.2): $CI = (L,U) = \left(\overline{X} - 1.96\frac{\sigma}{\sqrt{n}}, \overline{X} + 1.96\frac{\sigma}{\sqrt{n}}\right)$ Equation 3.2

Categories	Sub-Categories	Occurrences	Lower Limit %	$\hat{p}$ %	Upper Limit %	U-L
Construction (n=411)	Edit	151	32,08	36,74	41,40	9,32
	Copy/Move	103	20,87	25,06	29,25	8,38
	Connection	68	12,95	16,55	20,14	7,18
	Members	63	11,85	15,33	18,81	6,97
	Others	26	3,97	6,33	8,68	4,71
Graphic (n=469)	Dimensioning	185	35,02	39,45	43,87	8,85
	Text	138	25,30	29,42	33,55	8,25
	Copy-Move	62	10,15	13,22	16,29	6,13
	View	60	9,77	12,79	15,82	6,05
	Drawing	24	3,12	5,12	7,11	3,99
Points (n=322)	Intersection	90	23,05	27,95	32,85	9,80
	Mid points	59	14,10	18,32	22,55	8,45
	Member-points	30	6,14	9,32	12,49	6,35
	Bolt-points	25	4,84	7,76	10,69	5,85
	Add point	31	6,41	9,63	12,85	6,44
	Point parallel	20	3,57	6,21	8,85	5,27
	Perpendicular	20	3,57	6,21	8,85	5,27
	Delete	17	2,84	5,28	7,72	4,89
	Point-extend	11	1,43	3,42	5,40	3,97
	Other	19	3,33	5,90	8,47	5,15
View (n=235)	View-section	65	20,14	27,66	35,18	15,03
	Display settings	67	20,92	28,51	36,10	15,17
	View-member	71	22,50	30,21	37,93	15,43
	View-axis	10	0,86	4,26	7,65	6,78
	Show-weld	7	0,12	2,98	5,84	5,71
	View-3D	6	-0,10	2,55	5,20	5,30
	View-window	4	-0,47	1,70	3,88	4,35
	Others	5	-0,30	2,13	4,55	4,85
Info/Check (n=200)	Points-distance	63	25,06	31,50	37,94	12,88
	Member-info	51	19,46	25,50	31,54	12,08
	Search	33	11,36	16,50	21,64	10,29
	View Assembly	18	5,03	9,00	12,97	7,93
	Member ID Check	11	2,34	5,50	8,66	6,32
	Blink	8	1,28	4,00	6,72	5,43
	Bolt-info	7	0,95	3,50	6,05	5,09
	Others	9	1,63	4,50	7,37	5,75

Table 3.4: Probability values and confidence intervals of five main categories

Based on Table 3.5, one can conclude that the user carries out 5-7 construction commands (in five minute intervals), for which he/she carries out on average 44-55 “preparatory commands”. Table 3.6 presents relative frequency values of commands in both working modes. Moreover, Fig. 3-7 shows the same values as well as the percentage of “Productive” or “Preparatory” commands in each working mode. It should be noted that in calculation of productive commands in modelling mode, graphical commands are also considered as preparatory commands, because, as the observations also indicated, commands such as dimensioning or line are mostly used and therefore have preparatory functionality.

	Lower Limit		$\overline{X}$		Upper Limit		σ	
Construction	5,14	0,00	6,29	0,08	7,45	0,19	4,75	0,28
Graphic	0,69	10,20	1,48	15,54	2,27	20,89	3,24	13,36
View	2,55	0,08	3,34	0,75	4,12	1,42	3,23	1,66
File	0,00	0,45	0,06	1,00	0,12	1,55	0,24	1,38
Output	0,00	0,09	0,00	0,42	0,00	0,74	0,00	0,81
Navigation	37,01	29,66	43,05	40,79	49,08	51,93	24,82	27,83
Points	3,37	0,12	4,69	0,83	6,02	1,54	5,46	1,77
Info+check	2,15	0,07	2,89	0,50	3,64	0,93	3,07	1,08

Table 3.5: Values of means and standard deviations of commands' frequency

☐ Modelling mode ☐ Drawing mode

	No		**Lower Limit**		$\hat{p}$		**Upper Limit**		**U-L**	
Construction	409	2	9,25	0,00	10,18	0,00	11,12	0,00	1,87	0,00
Graphic	96	373	1,92	23,67	2,39	25,94	2,86	28,20	0,94	4,53
View	217	18	4,70	0,68	5,40	1,25	6,10	1,83	1,40	1,15
File	4	24	0,00	1,01	0,10	1,67	0,20	2,33	0,20	1,32
Output	0	10	0,00	0,27	0,00	0,70	0,00	1,12	0,00	0,86
Navigation	2798	979	68,23	65,67	69,65	68,08	71,08	70,49	2,84	4,82
Points	305	20	6,77	0,79	7,59	1,39	8,41	2,00	1,64	1,21
Info+check	188	12	4,03	0,36	4,68	0,83	5,33	1,30	1,31	0,94
	4017	1438								

Table 3.6: Calculated probability values and confidence intervals for command categories in two different working modes

☐ Modelling mode ☐ Drawing mode

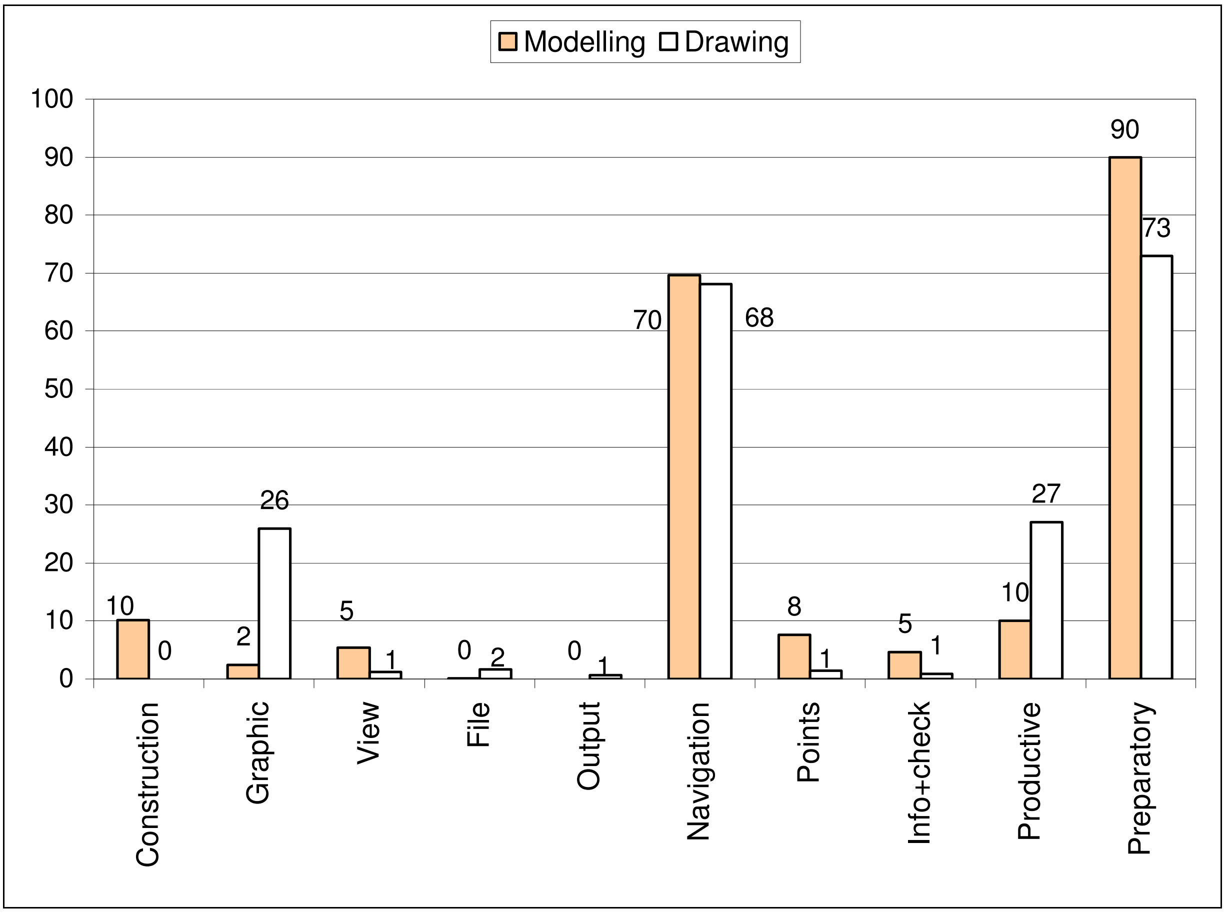


Fig. 3-7: Relative frequency of commands' categories in two working modes

As it was expected, in modelling mode, the extent of preparatory commands is higher (90%) than it is in drawing mode (73%). In modelling, users generally need to work more cautiously when working with construction commands, whereas working with graphical commands in drawing mode is more of routine tasks. According to the results in Table 3.6, preparatory commands (excluding navigation) are twice as frequent as productive commands in modelling mode.

In the end, a summary of all commands with their relative frequency are sorted in Table 3.7 and Table 3.8 based on modelling and drawing modes, respectively. Some commands are very frequent in both modes. Moreover, commands belonging to categories of "View", "Points" and "Info/Check" are also frequent in both working modes.

Category	Commands	Occurrences	Lower Limit	$\hat{p}$ %	Upper Limit	U-L
Construction (n=409)	Contour-edit	58	13,3	14,2	15,1	1,9
	Copy-p	54	12,3	13,2	14,1	1,8
	Bolting	33	7,3	8,1	8,8	1,4
	Cut off	29	6,4	7,1	7,8	1,4
	Move-dim	27	5,9	6,6	7,3	1,3
	Adjust	26	5,7	6,4	7,0	1,3
	Member-del	23	5,0	5,6	6,2	1,2
	Macro	22	4,8	5,4	6,0	1,2
	Undo	20	4,3	4,9	5,5	1,1
	Create profile	18	3,9	4,4	4,9	1,1
	Member-edit	17	3,6	4,2	4,7	1,1
	Connection-del	12	2,5	2,9	3,4	0,9
	Move-p	11	2,3	2,7	3,1	0,9
	Adjust on member	10	2,0	2,4	2,9	0,8
	Welding	10	2,0	2,4	2,9	0,8
	Mirror	8	1,6	2,0	2,3	0,7
	Bolt-edit	7	1,4	1,7	2,1	0,7
	Punch	6	1,1	1,5	1,8	0,6
	Other	18	3,9	4,4	4,9	1,1
Graphic (n=96)	Dimension	48	48,7	50,0	51,3	2,7
	Graphic-delete	14	13,6	14,6	15,5	1,9
	Line offset	12	11,6	12,5	13,4	1,8
	Textbox-move	6	5,6	6,3	6,9	1,3
	Dimension-delete	4	3,6	4,2	4,7	1,1
	Other (20)	12	11,6	12,5	13,4	1,8
View (n=217)	View-member	71	31,5	32,7	34,0	2,5
	View-section	59	26,0	27,2	28,4	2,4
	Display settings	58	25,6	26,7	27,9	2,3
	Others (7)	29	12,5	13,4	14,3	1,8
Points (n=305)	Intersection	86	27,0	28,2	29,4	2,4
	Mid points	59	18,3	19,3	20,4	2,1
	Member points	30	9,0	9,8	10,6	1,6
	Add points	30	9,0	9,8	10,6	1,6
	Bolt-points	25	7,5	8,2	8,9	1,5
	Point parallel	19	5,6	6,2	6,9	1,3
	Perpendicular	15	4,3	4,9	5,5	1,1
	Others (7)	41	12,5	13,4	14,3	1,8
Check/Info (n=188)	Points-distance	63	32,3	33,5	34,8	2,5
	Member-info	49	24,9	26,1	27,2	2,3
	Search	27	13,4	14,4	15,3	1,9
	View-hp	18	8,8	9,6	10,4	1,6
	Members-id-check	11	5,2	5,9	6,5	1,2
	Other (5)	20	9,8	10,6	11,5	1,6

Table 3.7: Relative frequency values and Confidence Intervals in modelling mode

Categories	Commands	Occurrences	Lower Limit	$\hat{p}$	Upper Limit	U-L
Graphic (n=373)	Textbox-move	88	19,28	23,59	27,90	8,62
	Dimension	74	15,79	19,84	23,89	8,09
	Dimension-move	25	4,16	6,70	9,24	5,08
	View-move	24	3,94	6,43	8,92	4,98
	Text	18	2,65	4,83	7,00	4,35
	Graphic-move	17	2,44	4,56	6,67	4,23
	Graphic-delete	17	2,44	4,56	6,67	4,23
	Dimension-del	17	2,44	4,56	6,67	4,23
	Text-edit	16	2,23	4,29	6,35	4,11
	Open-group	14	1,82	3,75	5,68	3,86
	Dim-edit	14	1,82	3,75	5,68	3,86
	Other (14)	49	9,71	13,14	16,56	6,86
View (n=18)*	Display settings	9	-	50,00	-	-
	View-section	6	-	33,33	-	-
	Other (8)	3	-	16,67	-	-
Points (n=20)*	Points delete	6	-	30,00	-	-
	Intersection	5	-	25,00	-	-
	Perpendicular	5	-	25,00	-	-
	Other (11)	4	-	20,00	-	-
Check/Info* (n=12)	Search	6	-	50,00	-	-
	Blink	4	-	33,33	-	-
	Other (8)	2	-	16,67	-	-
* For these categories no confidence interval was calculated since n<25						

Table 3.8: Relative frequency values and Confidence Intervals, drawing mode

Concluding Remarks

Results of such surveys are helpful in user interface design to decide upon priorities and efficiency. Practical guidelines especially on text-free strategies are given based on these results in chapter 5.

In general, when commands are categorised depending on their functionality and sorted based on their frequency, it can in fact reveal the users' needs and priorities in accomplishing their two basic tasks of modelling and output. This can show which functions or commands have to be available first to the user and which functions or commands have to be more direct. Therefore, with these results at hand, it is possible to address the issue of availability and hierarchy in user interface design.

Preparatory commands form 90% of all commands in modelling mode and 75% in drawing mode. Zooming excluded, these accounts are 66% and 29%, respectively. Because of this substantial difference between productive and preparatory commands in a high-performance CAD system, more attention must be given to preparatory commands. Obviously, developers of high-performance

CAD systems have been successful to make the essential functionalities of construction commands and automatic generation of output very efficient. On the other hand, these developers confirmed that they had never done scientific research about the frequency of commands and the time invested by the users. Therefore, the dominant importance of preparatory commands came to a surprise of leading software enterprises.

If preparatory commands are not direct enough, they can mislead the user; for example, if the user can not conveniently generate the desired working plane, he/she may start constructing at the wrong elevation and later on he/she will have to spend time modifying his/her work; Or if the user has to search too long to find the desired function, or if he/she has to fight the function to work, he/she will get distracted from his/her basic immediate goal. Furthermore, the users control and evaluate the state of the system through preparatory commands, in order to see if they have received the right results. A very common case is where they carry out the final quality check of the model. The results indicate that in 16% of all observations, the user was only controlling and checking without performing any significant changes to the model.

Based on the results, given in Table 3-7 and 3-8, following commands in each category have high priority for a more efficient text-free solution and/or directness:

Construction: Contour-edit, Cut off, Adjust, Edit member property, Copy/move, Bolting, Create profile, Macro.

Graphic: Dimension, Dimension-move, Textbox-move, Text-edit, Text, Copy/move, View-move.

Points: Line-intersection, Mid-points, Member points, Bolt points, Add points, Points parallel, and Perpendicular.

View: View-section, Display settings and View member.

Info/Check: Member info, Search, View-assembly, and Members identity check.

Navigation: Zooming and Panning.

In addition to proposing a text-free solution for each individual command mentioned above, the placement and its availability to the user must also be considered. This is also a factor, which contributes to the usability of the software. Generally, the more frequent the command, the closer or more available it must be to the user. There are a number of ways to achieve this, such as:

- Enhanced directness using adjustment handles

This is the subject of section 4.4 and will be elaborated there.

- The use of layers

This is in fact the most basic and fundamental solution appropriate also for novice users. Here, the more frequent commands must be placed at front layers visible to the user, whereas the less frequent ones should appear at more remote layers.

- Icons/menu arrangement

The right arrangement of commands in the icon/menu bar will make them easier and more available for the user to reach. It is suggested that the more frequent commands be placed near the top in a menu bar and/or a vertical icon bar and to left and centre in a horizontal icon bar.

- The use of a customized icon box

In either working modes, it is often the case that the user may need to use a certain set of commands repeatedly. The idea of a customized icon box is proposed to assist the user in such cases and save him the time and effort needed for many mouse travels across the screen. This can be implemented as a “flying” window, in which the user can drag and drop or simply load the desired icons. The frequent commands resulting from the analysis may be used as default icons in this icon box.

- The use of keyboard short-cuts

For very frequent commands it will be reasonable to provide shortcut keys. Shortcut keys can be quicker than travelling the pointer to the right icon or menu and clicking. This will be explained more in section 4.1.

In order to provide international text-free solutions, it is important to consider commands in terms of both frequency and functionality. It should be noted that results of this survey highlights this significance in terms of frequency and not functionality. Some commands such as Redo, Library or Drawings/Lists generation certainly offer significant functionality even though they are not really frequent.

4. Basic GUI Components in High-Performance CAD Systems

In this chapter, basic GUI components which appear and/or are needed frequently are discussed based on the principles and guidelines discussed in Chapter 2.

4.1 User's Input

Mouse

As a Windows-based application, a high-performance CAD system for steel detailing is characterised for its WIMP (Windows, Icon, Menu, Pointing device) interface. Among different input devices for a CAD program, the mouse is indeed the most popular one and facilitates the following functions:

- A pointing device for positioning the cursor;
- Selection of objects;
- Main device for user's direct manipulation actions such as drag and drop or use of adjustment handles;
- Conveying different contexts of applications within the program by assuming different cursor/pointer shapes;
- Opening contextual/pop-up menus ;
- Navigation (zooming and panning) using the scroll wheel.

Zooming and panning

According to the survey conducted in this research, zooming and panning is the most frequent command in steel detailing CAD systems. For many users familiar with AutoCAD, such as Iranian users, both functions are facilitated using the mouse wheel: zooming by rolling the mouse wheel and panning by pressing and dragging it. This is indeed the most direct method and is used also in TeklaStructures. Moreover, in TeklaStructures by aid of the keyboard, the user can also use 3D rotation and navigate efficiently within the model. In fact, an efficient zooming and panning function can substantially lower the frequency of other preparatory commands such as "section" or "standard view from a profile". Bocad-3D offers a solution, which even though very brilliant in its time, seems not so effective today given the advent of the mouse wheel: to zoom in, the user must determine the desired area (in form of a rectangle) by pressing and releasing the middle mouse button (or scroll wheel). Depending on the original state and the desired zooming level, user may have to repeat the process several times. Clicking only once on the mouse wheel brings the previous zooming level (zoom out). Panning in the program is facilitated by drawing wiry rectangles in horizontal/vertical directions using the same method. As it is seen, the method is not as efficient as using the mouse wheel, especially for panning which is limited only to orthogonal directions.

AutoCAD further provides extra tools that offer extensive zooming functions such as aerial view or dynamic view, which allow speedy navigation within a view (Fig. 4-1). These functions are useful especially when working with views, in which many elements have to be displayed at a time, and hence consecutive zooming and panning would take up a lot of processing resources resulting in lowered system's response time.

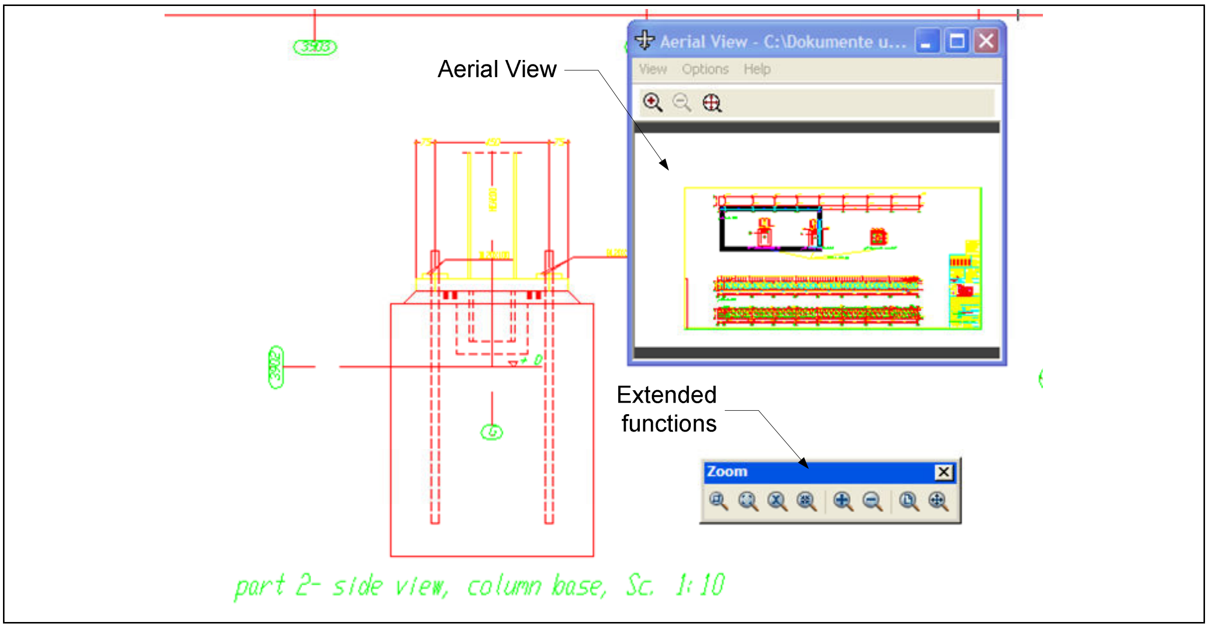


Fig. 4-1: Extended navigation tools in AutoCAD

Keyboard

Although CAD systems are designed to be operated primarily by pointing at, clicking on, and dragging objects on the screen using mainly the mouse, they require the use of keyboard for the text or data entry. This means that users very often have to switch back and forth between the keyboard and the mouse. Therefore, CAD users usually work with one hand on the mouse and the other on the keyboard (or in the vicinity ready to operate). Johnson considers several reasons why some users may prefer to keep their hand on the keyboard and thereby believes that all GUI-based applications must be designed to be operated as well without a pointing device [11]. Maybe, the most important reason for CAD users is to save time; in the time it takes to move the mouse to an on-screen control component (such as a menu or a button) to invoke a command, most users can execute several commands from the keyboard.

Another important advantage of using the keyboard is that it maintains user's visual focus on the screen. If successive commands are to be executed on an object, the user can favourably use the keyboard without losing visual attention as a result of looking away to screen borders to find the required on-screen control component. Therefore, the use of keyboard can enhance efficiency.

In the framework of this research, use of keyboard plays even a more significant role for users in developing countries who are already familiar with AutoCAD, which allows direct keyboard entry for many commands. According to author's observation in Iranian engineering offices, keyboard is often used when working with AutoCAD. Therefore, many users in developing countries are likely to prefer using both hands when working with a CAD program. Finally using the keyboard has an "expert look" and is therefore motivating for novice users.

4.2 Choice boxes and Dialogs

Within CAD programs, users interact often simultaneously with a number of primary, secondary and utility windows. In fact, much of what a user does in the program is done through his/her interaction with different forms of dialog boxes. Therefore, correct design of these interface elements can effectively add to the usability of the system. As it is subsequently shown, in both market leading steel detailing CAD programs, there are instances of usability issues concerning these elements. Especially Bocad-3D suffers from usability shortcomings in this respect. As a result, the present section aims to discuss at first some usability considerations about these elements and demonstrate their corresponding shortcomings in existing programs. In each case, improvements and innovative workable (text-free) solutions are proposed. Major Human Interface Guidelines (HIG) referred here are from Microsoft [61], Apple [62], Java [63] and GENOM [64] with primary focus on Microsoft given the target users in this research.

Before starting with the discussion, some basic terms should be explained. According to the Java Look and Feel Guidelines [63]:

"A primary window is a window in which users' main interaction with the data or document takes place. An application can use any number of primary windows, which can be opened, closed, minimised, or resized independently.

A secondary window [also known as dialog box] is a supportive window that is dependent on a primary window or another secondary window. In the secondary window, users can view and provide additional information about actions or objects in the primary window.

A utility window is a window whose contents affect an active primary window. Unlike secondary windows, utility windows remain open when primary windows are closed or minimized"

In a high-performance CAD program, primary windows are indeed windows in which views of the model are generated and shown (this also includes drawings). However, apart from primary windows, users interact frequently with different forms of secondary windows such as dialog boxes, which are used to elicit a response from the user or to obtain additional information needed to carry out a particular command or task [61]. There are different types of dialog boxes for different sorts of interaction. Here, the focus is mainly on dialog boxes or choice dialog boxes, which are used to enter additional settings for commands within high-performance CAD systems.

A *dialog box* is a secondary window that allows users to perform a command, asks users a question, or provides users with information or progress feedback. A typical dialog box consists of a title bar (to identify the command, feature, or program where a dialog box came from), an optional main instruction (to explain the user's objective with the dialog box), various controls in the content area (to present options), and commit buttons to indicate how the user wants to commit the task [61]. Dialog boxes should be carefully designed to offer the intended flexibility, otherwise they become an easy way to annoy users, interrupt their work flow and make the program feel indirect and tedious to use [61].

Design concerns about dialog boxes include their modality, display location, their automatic versus manual display, commit buttons, and save/load setting buttons, which are specific to high-performance CAD programs.

Dialog boxes have two fundamental types:

Modal dialog boxes require users to complete and close before continuing with the owner window;

Modeless dialog boxes allow users to switch between the dialog box and the owner window as desired.

All human interface guidelines agree on the point that user’s interaction with the system must be designed as modeless as possible. Apple Interface Guidelines explains [62]:

“As much as possible, allow users to do whatever they want at all times. Avoid using modes that lock them into one operation and prevent them from working on anything else until that operation is completed. “

This must also be applied to the design of dialog boxes since designing a modal dialog box, where it is not necessary, restricts users and breaks their work flow by demanding attention. There are useful guidelines, which can effectively help interaction designers upon the decision whether a dialog box must be modal. For example, according to GNOME:

*"Use an application modal window only if allowing interaction with other parts of the application while the window is open could cause data loss or some other serious problem. Provide a clear way of leaving the modal window, such as a **Cancel** button in an alert. "*

or according to Microsoft:

"A modal dialog box is best used for critical or infrequent, one-off tasks that require completion before continuing. By contrast, a modeless dialog box is best used for frequent, repetitive, or on-going tasks. "

Despite these clear guidelines, there are instances of modal dialog boxes in Bocad-3D where they should actually be modeless due to their frequent use (e.g., the dialog box for "display settings" or the "owned widows" in the command "search members").

Controls

Many dialogs in a CAD program are in form of "Choice dialogs", which present users with a set of choices to specify a command more completely. As a certain design rule, these dialog boxes (choice dialogs) must have only a close button (or additionally a help button) presented in the title-bar since there is no need to minimize, restore or maximize these secondary windows. Ignoring this rule, as in Bocad-3D, is a design mistake: As it is shown in Fig. 4-2a, this can sometimes become annoying because it forces the user to look carefully to select the right control; otherwise, to the user's surprise, the box may maximize to the whole screen (Fig. 4-2b). By contrast, the design rule allows users to reluctantly move the mouse towards the top right corner and click the desired control. Unfortunately, this wrong design in Bocad-3D appears in all dialog boxes in the program even for simple dialogs that ask user a question!

Display Location

One issue concerning choice boxes is their display location. By default, the program always displays a window in the same location where it was last accessed, whether it is its default location or the location to which the user might have moved the window. Although this is a right choice in general, there are exceptions to it namely for contextual dialogs and owned windows. For these dialogs, Microsoft sets following rules [61]:

"If a dialog is contextual, display it near the object from which it was launched. However, place it out of the way (preferably offset down and to the right) so that the object is not covered by the dialog. "

"If a window is an owned window, initially display it 'centered' on top of the owner window. For subsequent display, consider displaying it in its last location (relative

to the owner window) if doing so is likely to be more convenient."

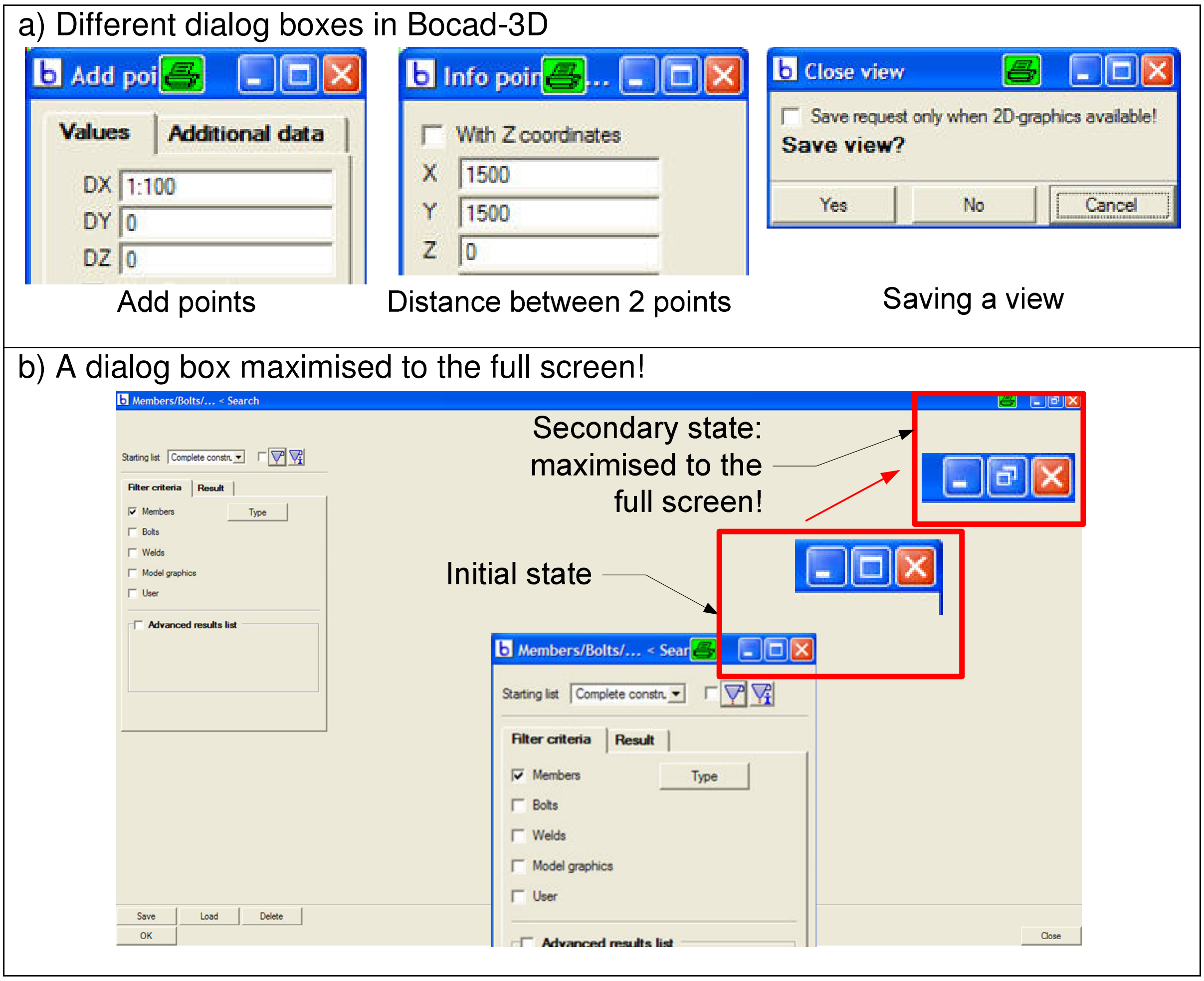


Fig. 4-2: Dialog boxes in Bocad-3D having controls of primary windows

These rules already explain the possible drawbacks that arise, should the software stick to the general rule only, which is the case in Bocad-3D: In case of contextual dialogs, they may appear randomly relative to the target object, because they track the last accessed location only, which may potentially be irrelevant to the currently selected object. Therefore, they may display too far from the object or they may simply cover it.

In case of owned windows, if they are not displayed relative to the owner window but rather to their last accessed absolute location, they may appear again too far from the owner window. On the other hand, as Microsoft explains, this too must be done "if doing so is likely to be more convenient". There are in fact cases in Bocad-3D, where it is clearly not convenient for user to have owned windows displayed in their last relative location. Fig. 4-3 demonstrates a typical example of this drawback taken from author's real-life working experience with Bocad-3D. As it is shown, such cases usually result in excessive mouse journey across the screen lowering efficiency. Therefore, for dialog boxes with owned windows, it is better to display the owned windows not only relative to their owner window, but also their relative location must be fixed. Indeed, there will be limitations concerning original location of the owner window relative to screen borders, but displaying owned windows near the cursor often seems to be a good choice.

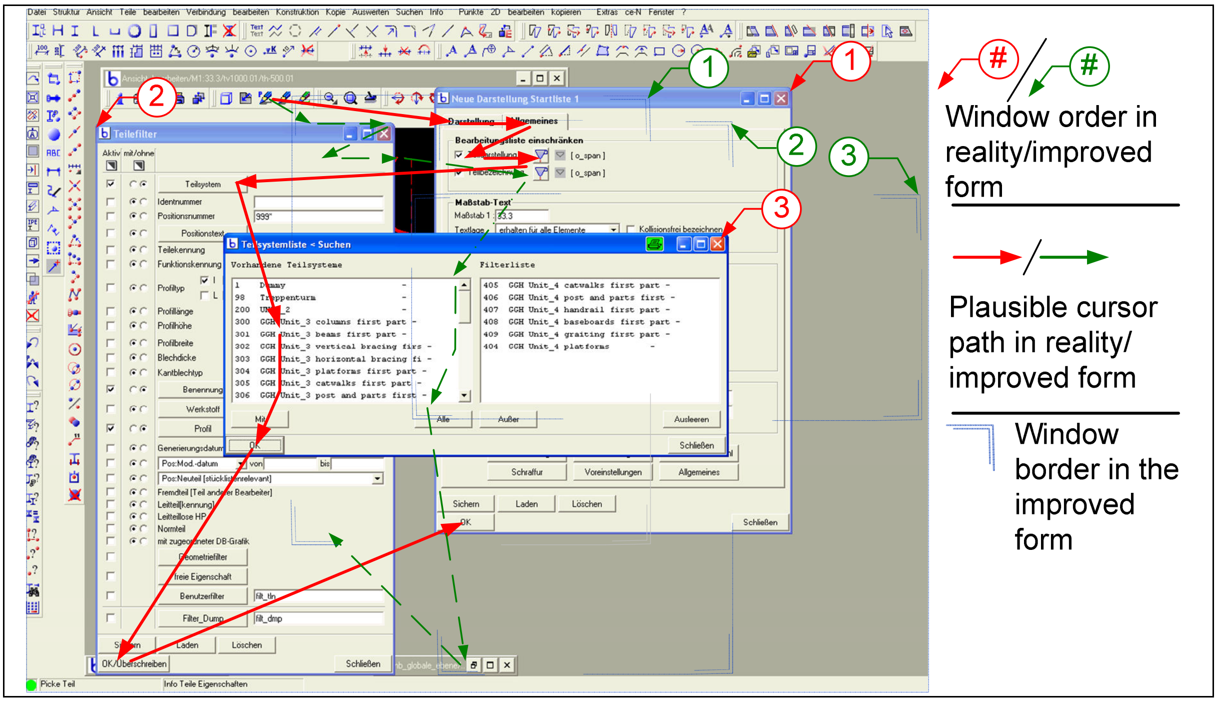


Fig. 4-3: Bad positioning of owned windows in Bocad-3D

Automatic vs. Manual Display

This part focuses on the decision where and how dialog boxes should automatically/manually appear. This matters because it could become annoying to allow all choice boxes appear immediately on the screen as user invokes their command (Automatic display). On the other hand, it is inefficient to hide them all at first and display them only upon further inquiry (Manual display). Therefore, a right decision has to be made about the automatic/manual display of each individual command's dialog box. Microsoft provides the following guideline in this case:

"For choice dialogs, annoyance factor is normally low, because they are user-initiated and need a response, but could be high if users rarely change default values."

Based on this rule, every command has to be considered individually given its characteristic and whether users often need to change its default values. Only based on such considerations, it is possible to decide if a command's dialog box must appear automatically or rather by user's further inquiry. As a general rule, dialog boxes of commands for which user will most probably enter new settings must open automatically. Typical examples are search, display settings, point parallel, copy/move by dimension or rotate. In the same category are the commands through which user creates new objects within the program, such as profiles, bolts and welds. Finally, commonly used commands, which register user's previously given settings, must open automatically when the settings are likely to change (e.g. the dialog box for "member length edit" in Bocad-3D).

Although both CAD programs seem to consider these rules, there are also instances where the rule has been violated. In both programs, when user wants to create a profile/bolt/weld/connection, he/she has to invoke the dialog box manually. When working with the dialog box for "member length edit" in Bocad-3D, users often have to change the previous values or at least check them because the command registers the values. Nevertheless, they have to invoke this dialog box manually.

Another point that needs attention in both programs is the manual way, which displays the dialog boxes. Both programs provide rather an unfamiliar way to invoke the commands' boxes (choice dialog boxes for commands): In Bocad-3D, user must use the mouse right-click button and in TeklaStructures, he/she must double-click on the icon instead. Either method is rarely seen in other programs. Therefore, a typical user is almost unlikely to find the choice dialog box on his/her own. Especially the method used in TeklaStructures is to some extent even misleading because this is what computer users are accustomed to when interacting with folder icons in Microsoft Windows but not with internal program icons. Although the method used in Bocad-3D is somewhat superior, it has the drawback that it completely overrides other uses of mouse right-click menu during the command, i.e. accommodation of some useful options in the command's context.

To conclude this section, it is necessary to make the right decision if the command must appear automatically or rather manually through the user's further effort. If latter is the case, probably the most familiar way for users to invoke the dialog box manually is to place an option for it in the mouse-right click menu when the command is active.

Response/Commit Buttons

Response buttons (or commit buttons as in Microsoft terminology) are used to respond to a dialog. Typically, there are three types of commit buttons for choice dialogs: Cancel, OK and Apply.

Clicking Cancel means abandon all changes, cancel the task, close the window, and return the environment to its previous state, leaving no side effect.

Clicking OK means apply all settings in the window, close the window.

Guidelines provide different rules about the role of the Apply button: GNOME and Microsoft suggest that clicking the Apply button, applies the pending changes, but does not close the window should the user wish to change his/her mind. Apple on the other hand, considers a preview role for Apply while at the same time does not dismiss the window. The user can decide whether to accept the previewed changes by clicking OK or to reject them by clicking Cancel. Although in a common windows application this seems to be a good idea, in a CAD program with a decent Undo/Redo function, this will be redundant. Hence, the guideline offered by Microsoft or GNOME is in fact a better choice.

In choice of commit buttons, both Bocad-3D and TeklaStructures have a categorical approach. Bocad-3D normally does not include an Apply button; TeklaStructures uses the same set of commit buttons (Fig. 4-4), which appear in all command dialogs regardless of specific characteristics of the command they are representing.

Fig. 4-4: Commit buttons in TeklaStructures

Apart from its language dependency, this set up in TeklaStructures has three drawbacks: (1) According to Microsoft, Apply button (or any other button with a similar role) are to be used only in property windows not in choice dialogs; (2) “Modify”, “Apply” and “OK” present different/none/ or even wrong functionalities in different command boxes, see Fig. 4-5. In some commands, “Modify” actually has the same role as Apply, whereas “Apply”, in some cases, does not apply changes on the object (practically no effect). “OK” on the other hand, sometimes takes the role of cancel by abandoning the changes and dismissing the box! This is obviously a design mistake because the program leaves it to the user to deduce what the button does in the addressed command and whether it applies changes; (3) “Apply” and “Modify”, as well as “Get”, are by default enabled, which is wrong; they must enable only when user inserts changes.

Fig. 4-6 shows a recommended text-free solution for commit buttons developed in this thesis. Another useful aspect, which should be considered for commit buttons, is their exclusive access keys in Microsoft Windows, i.e. assigning Enter for OK and Esc for Cancel. However, Enter is basically the access key for the dialog default commit button, which can be OK, Yes or any other similar command. In fact assigning a default commit button is another useful possibility, for it facilitates user’s interaction via the keyboard, too. There are also certain design considerations for choosing the dialog default button which can be found in human interface guidelines.

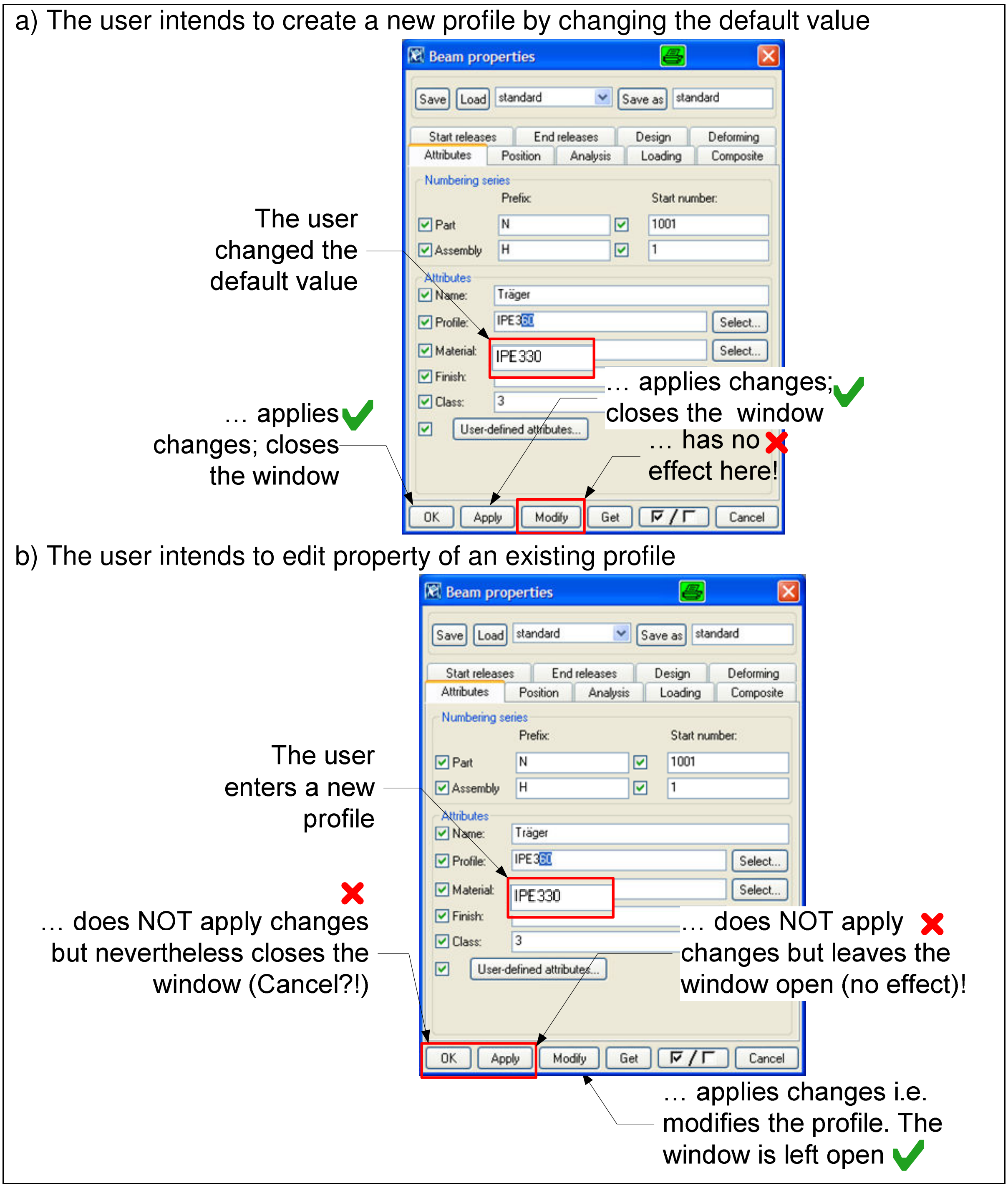


Fig. 4-5: Commit buttons in TeklaStructures with different functions in apparently same dialog boxes

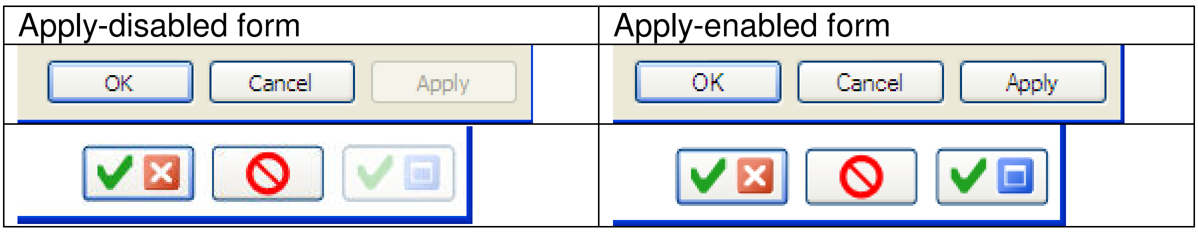


Fig. 4-6: Commit buttons in Microsoft Windows compared to the text-free solution

Save/Load Settings

These buttons are actually specific to high-performance CAD systems. It is sometimes very useful for users to save the settings in a command window and load them for later use. Alternatively, the user may want to load a setting from another element; for example, when the user intends to create a new profile and simply wants to load similar settings from an existing one. This function is especially very common in commands such as view properties, sections properties, own connection/components or creating a profile. In both TeklaStructures and Bocad-3D, this function has been foreseen, as shown in Fig. 4-7a and b, respectively. There are usually some standard settings embedded in both programs and the user has the possibility to add and save his/her settings with new names.

a. TeklaStructures

b. Bocad-3D

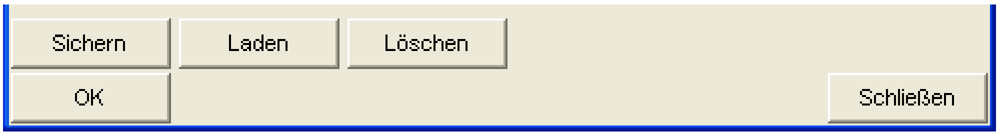


Fig. 4-7: Saving/loading user’s settings. a. Bocad-3D, b. TeklaStructures

The design of this function is language-dependent. However, by placing this function at top of a dialog-box and using a drop-down menu, the function has gained a better visibility as well as ergonomics in TeklaStructures. This way, the user can work with this function at the same layer (Fig. 4-8a), whereas in Bocad-3D by opening up a new window the user is transferred to a new layer, see Fig. 4-8b. As a result, a novice user will gain a better understanding of this function and an expert user will apply it with slightly more efficiency. However, apart from language dependency, this function can yet be improved as shown in the following.

1. “Save” and “Save as” as two buttons are superfluous. By providing the adequate visual feedback to user in the text-field, the user will be informed if he/she wants to save with a new name or overwrite the existing one. For this sake, an auto-completion function as in Microsoft Windows as well as a warning for a possible overwriting are proposed.

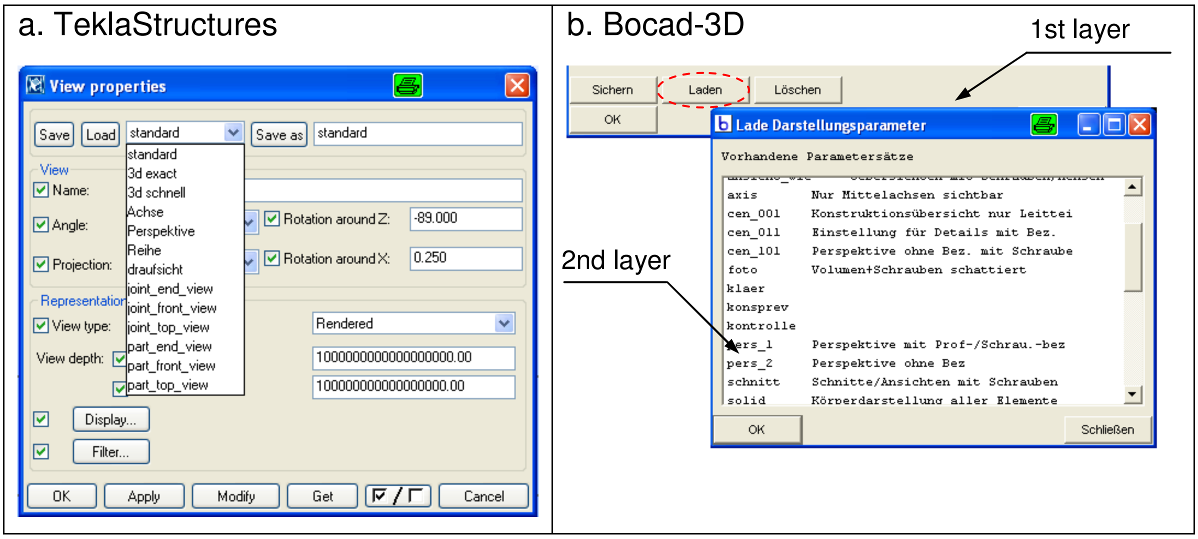


Fig. 4-8: Saving/loading user's settings in (a) TeklaStructures, (b) Bocad-3D

2. The optimal location of a button next to a drop-down field is to the right of the field. This makes the function faster, as it requires the shorter mouse travel as shown schematically in Fig. 4-9.

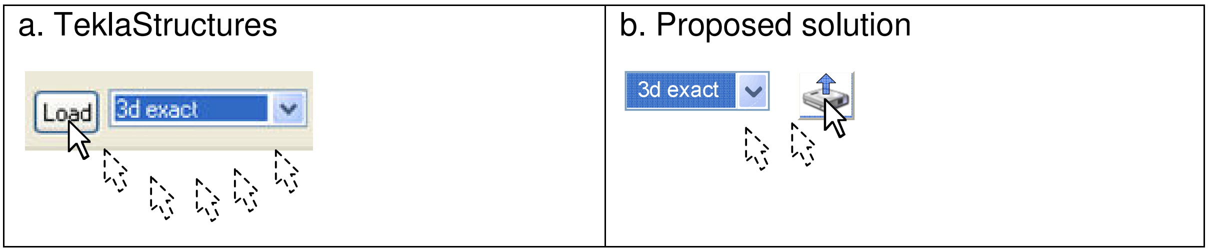


Fig. 4-9: Optimised location of the execution button next to a drop-down field

3. Both buttons must have auto-enabling behaviour; they should enable only when the corresponding function is available for execution. This is missing in TeklaStructures and as it is shown in Fig. 4-10a, even when user selects another setting, no change takes place in the state of "Load" button and therefore the user will not have the chance to realise the dependency of the drop-down menu and the "Load" button. As a result, by the very first experiences the user does not receive the desired result. In the proposed solution, however, the "Load" button in the default form is disabled and enables only when a setting has been selected (Fig. 4-10b). This provides an effective hint to the user to press the button intuitively in order to have the setting taken effect.

3. The use of a "Delete" button in this function could actually be abandoned since it is seldom used. If users are not happy with a setting, it will not disturb with their work because they can simply overwrite it. Thus, the "Delete" button is not really necessary.

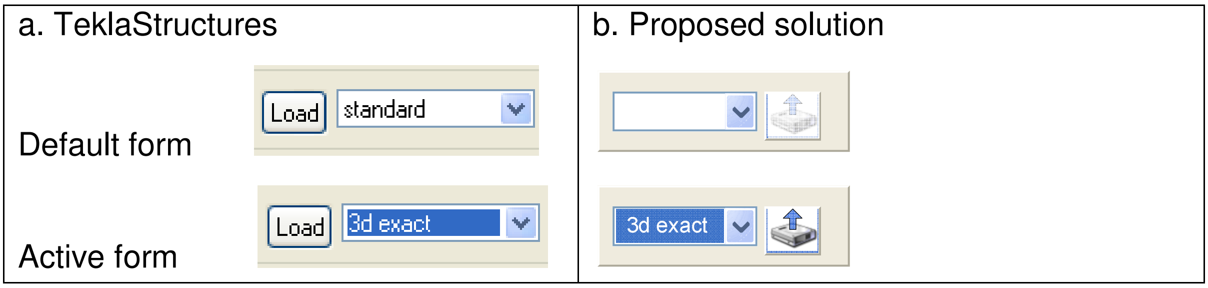


Fig. 4-10: Auto-activation of a button

4. For loading a setting from an existing element, a button with the common mouse cursor can be accommodated within this block. The mouse pointer is considered to be an appropriate choice for this function since it has affordance of selecting. In other words, by seeing this item within this block the user is likely to understand that he/she must select an object to load the settings. Moreover, consistent use of this icon in other commands facilitates its quick learning. Fig. 4-11 presents a text-free solution for this function.

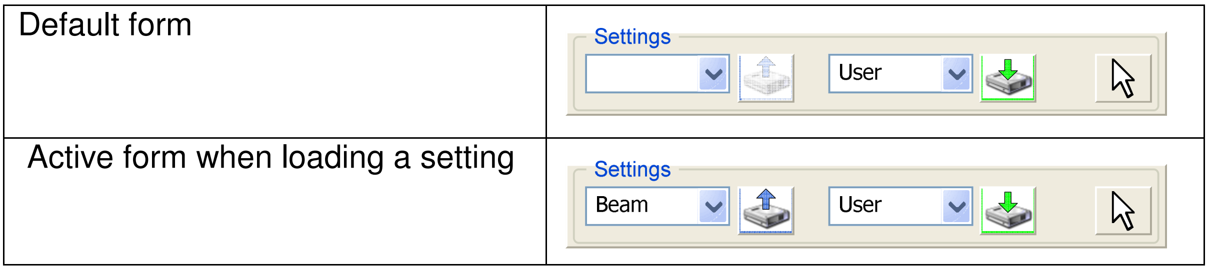


Fig. 4-11: Text-free function for saving/loading user's settings

4.3 Selection

Selection is definitely one of the most basic functions in CAD programs. Nearly all commands require that user selects one or some types of objects. This is usually done by direct mouse input through a single click or a lasso—as a freehand selection. Moreover, in all windows applications, it is possible to adjust selections; i.e. to add elements or remove from a selection using keyboard modifiers (Ctrl and Shift) with the mouse.

Fundamental Types of Selection

- Selection via "Single mouse click"

"Single mouse click" on any object selects that object; "Single click" on another object will make a new selection by replacing the old one.

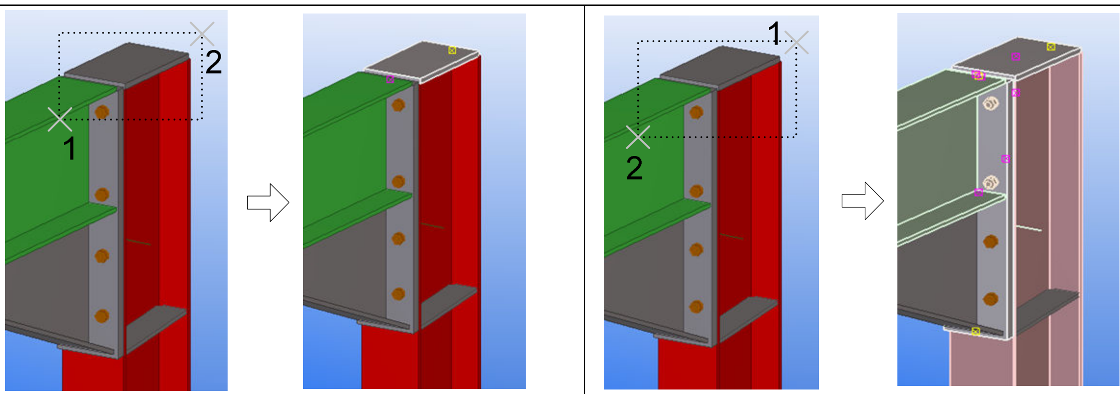
- Selection via lasso

A lasso or a freehand selection is a common practice in all GUI applications. By a lasso, the user determines a range by drawing a bounding outline (sometimes referred to as marquee). The outline is typically a rectangle but arbitrary shapes as in graphic programs are possible.

In CAD programs, it is also very useful to make a distinction whether an object is totally enclosed or only intersected by the bounding region to be affected by the selection operation. This can save the time for adjusting the right view where the entire object lies within the bounding region. Both in AutoCAD and in TeklaStructures[1] this distinction is made by direction of a lasso:

- If the user drags the cursor from left to right to create a lasso, only the objects that are entirely enclosed will be selected, see Fig. 4-12a;
- If the user drags the cursor from right to left to create a lasso, objects that the lasso encloses or crosses will be selected, see Fig. 4-12b.

AutoCAD calls these two lasso selections as “enclosing window” and “crossing window” selection.



a. Objects selected using a window selection method

b. Objects selected using crossing selection method.

Fig. 4-12: *Window selection* method versus *crossing selection* method

Modifying a Selection

The Ctrl key is the toggle or modifier for selections in all window applications. If the user presses the Ctrl key while making a new selection, it preserves any existing selection and adds or subtracts the new object to it. In Windows, the Ctrl key has a toggle function. This means that clicking on an object while having the Ctrl key pressed toggles the selection state of an object; i.e. if it is not selected, selects it and if it is already selected, cancels it. Thus, by using this key, the user can add or remove an object to an existing selection.

Using the same method in a CAD program has the advantage that it complies with users’ prior experience and hence, they can use it without any problem. However, this method has a drawback on efficiency, as in a CAD system, the users most often need to select multiple objects and therefore would have to hold down the

[1] Apparently this function exists also in Bocad-3D but it does not always function correctly!

Ctrl key during the selection. Although this alone may not be very bad, the problem rises when during selection, the user wants to move his/her hand on the keyboard in order to adjust other settings, such as to switch the snap or to change the view. At this point, they must always bear in mind to hold down the Ctrl key; otherwise, selecting a new object without the Ctrl Key pressed, clears all previous selections. A better alternative, as used in Bocad-3D and to some extent in AutoCAD, is to make the single mouse selection by default “accumulative”. This means that the user can accumulate an infinite number of objects within a selection only via mouse input—by “single-click” or a lasso. The key modifiers can and should then be used to remove an object from a selection. On the other hand, in order to avoid any possible inconsistency with user’s background experiences with windows, the “toggle” function as in Windows should be preserved for the Ctrl key. This is on the one hand consistent with windows as well as AutoCAD and on the other hand does not exhibit the abovementioned drawback on efficiency.

Table 4.1 summarises the selection methods as well as their corresponding advantages and disadvantages in AutoCAD, TeklaStructures and Bocad-3D.

	Multiple Selection	**Add/remove from a selection**	**Advantage**	**Disadvantage**
AutoCAD	Without key-modifiers	Shift key removes from a selection; Ctrl key has no function.	Efficient	Inconsistent with Windows in mouse selection as well as functions of Shift and Ctrl keys. The problem is however alleviated by using an effective mouse pointer.
TeklaStructures	With key-modifiers	Shift key adds to selection; Ctrl key has a toggle function.	Consistent with Windows	Not very efficient.
Bocad-3D	Without key-modifiers	Ctrl key has the toggle function	Efficient, Ctrl key Consistent with Windows	For users with no experience with AutoCAD, it may seem a little strange at first especially because the mouse pointer indicates no change.

Table 4.1: Selection methods in AutoCAD, TeklaStructures and Bocad-3D

Selection Feedback

It is necessary to provide a feedback when the user selects an object. This is normally done by highlighting the selected object through alteration of its colour. Moreover, CAD systems usually provide a textual feedback about the number of objects in the selection. Fig. 4-13 shows this type of feedback in Bocad-3D and

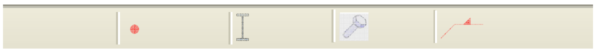
TeklaStructures. It is also sometimes useful to cue the user not only about the number but also about the type of objects being selected. This can help the user to recognise if he/she has selected the right object(s) or if, for example, a part together with its welds and bolts has been selected. This feedback can be best presented in the status bar as in Fig. 4-14. On the other hand, since this option is not crucial and may also take up resources, its presentation should be left to the user. This way the user can get the advantage of this option when he/she needs it and can have it hidden otherwise.

a. Bocad-3D

b. TeklaStructures

Fig. 4-13: Feedback on the status bar indicating the number of objects being selected

a. Default/Inactive form



b. Active form

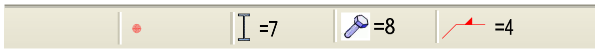


Fig. 4-14: Text-free feedback on the status bar about the type and number of objects being selected

Task Analysis of Selection

In order to enhance the usability of the selection function, it is necessary to perform a task analysis. A task analysis of selection reveals following points:

- Deselection: The user has just selected one or some objects, but realises that it is not what he/she wanted. Therefore, he/she needs to deselect it/them;
- Reselection: The user has already deselected one or some objects but changes his/her mind. Therefore, he/she needs to reselect it/them;
- Clearing a selection: The user wants to clear the selection without cancelling the command;
- Previous selection: The user wants to operate on previously selected objects. In the most effective form as in Bocad-3D, this can be defined as window/command dependent. This means that the objects will vary depending on in which window and for which command they were last selected;
- Single Deselection: The user wants to deselect one or some objects from a group of selected objects but not necessarily in the same order they have been just selected;
- Different input: Some commands like copy require input at two or even three stages. A function that sometimes can be very helpful is the possibility to step back into a previous input stage. This will give user full flexibility: If they are in the second or even the third stage of input and realise that they want to modify

their input from a previous stage, they can easily step back into the desired stage, modify their input and continue with the next stage without cancelling the command and invoking it again.

These functions are missing in TeklaStructures; therefore, the user sometimes experiences difficulties when after selecting a number of objects, he/she selects a wrong object without having the Shift-key pressed. This will clear the selection and in the absence of such functionality, the user will have to pick all the objects once again. By contrast, Bocad-3D offers a full flexibility concerning these tasks. Nonetheless, their solution has following drawbacks:

The functions are accessible only through key combinations. There is no graphical or textual clue whatsoever to help the user learn about such functions. Providing hidden functions, as such, has the drawback that they can be accessed only by remembering and typing rather than seeing and pointing.

If the user wants to clear a selection, he/she needs to press the “Del” key. This is completely against the normal function of this key in all other window programs. Although the idea of clearing a selection seems to be somehow similar to deletion, it is wrong to assign such a role to the “Del” key since in all window programs this key has the unique function of deleting not clearing. Therefore, apart from the fact that this setting is strange and definitely not direct, it leads to partial confusion of the user when working simultaneously with Bocad-3D and AutoCAD for example.

Recommended Solution

To overcome the discussed shortcomings, following solutions are proposed:

- Graphical solution

A graphical package for selection, which addresses the analysed tasks, can be placed next to the selection feedback (Fig. 4-14) on the status bar. A metaphor that probably best implies these tasks to the user is those used in Windows media player for forward/backward, beginning/end of a playing list. However, it is important to notice that simply presenting these symbols is not sufficient and to be most effective they must direct users’ attention toward their possible functionalities. This could be done by automatic activating/deactivating, which is explained in Table 4.2. Fig. 4-15 shows a variant of how these symbols may look like on the status bar.

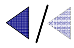
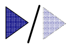
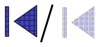
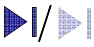
Task/Function	Proposed Symbol, Active/Default form	Must be Active only when
Deselection		User has just selected one or some objects.
Reselection		User has just deselected one or some objects.
Clearing selection and/or stepping back into a previous input stage.		There is an active selection.
Previous selection		1) There is no active selection but from the last time that the command was invoked, objects have been selected in this window. 2) The user has just deselected some objects but would want to reselect them all at once.

Table 4.2: Graphical text-free solutions for “Selection”

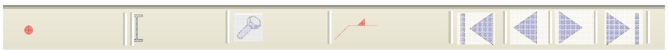
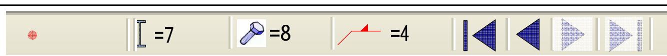
a. Default form	
b. Active form	

Fig. 4-15: Selection package next to selection feedback on the status bar

- Keyboard solution

Table 4.3 presents keyboard accelerators to address the discussed task scenarios in a selection. As an alternative, the use of arrow keys is also presented.

Task/Function	Keyboard solution	Alternative
Deselection	U as in Undo.	Arrow key left.
Reselection	R as in Redo.	Arrow key right.
Clearing selection	“0”	Arrow key down.
Previous selection	P as in Previous.	Arrow key Up.
Single deselection	Ctrl key as the toggling function.	None.

Table 4.3: Keyboard solutions for “Selection”

Selection Filters

High-performance CAD systems offer functions to filter desired objects during a selection. This will facilitate users’ task and will usually save the time needed for filtering the desired objects by other methods—such as “search function”. Moreover, this function avoids any potential errors which may occur when the user mistakenly selects the wrong objects and does not realise it.

Depending on different elements within the program, high-performance CAD systems offer different object filter toolbars, as shown in Fig. 4-16 in Bocad-3D and in TeklaStructures. Here the following points should be considered:

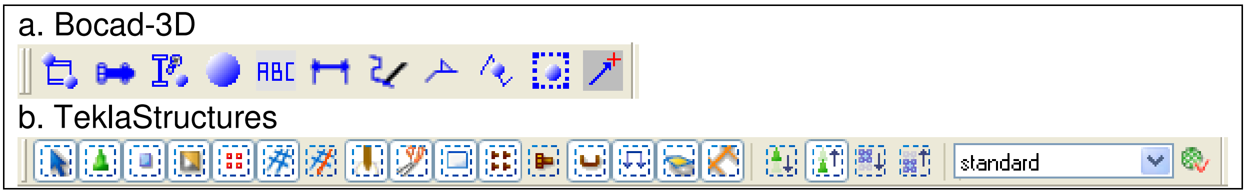


Fig. 4-16: Object filters toolbar in Bocad-3D and in TeklaStructures

- Feedback and visibility

The selection filter should be visible to user at any instant. This is done effectively in TeklaStructures by showing the active filter as a depressed button on the selection toolbar (Fig. 4-17a). The textual feedback on the taskbar, as in Bocad-3D, is not as effective (Fig. 4-17b).

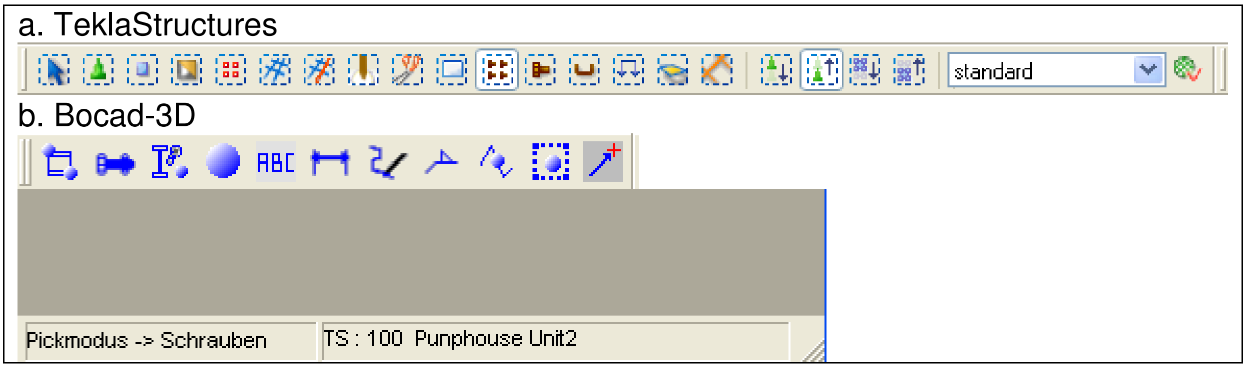


Fig. 4-17: Feedback for object filters in (a) TeklaStructures, (b) in Bocad-3D

- Sensible filtering defaults for each command

For each command a sensible default should be set so that the user would not have to spend time adjusting the filter. A wrong or an inadequate default setting can be very annoying; for example, in Bocad-3D for the command “move Text-leader point” the default setting is “any” which means the user can basically select any object! When working fast, it can happen that they mistakenly pick a profile for example and receive an error message. To get rid of it, they will have to invoke the command again.

For some commands however, the default settings can even be locked; i.e. the users cannot change them. This can be done for the commands that deal with certain object types such as all “editing” commands; for example, if it is weld editing or dimension editing, the filter object must be set automatically to weld or dimension.

It should be noted that defining such a setting, is not against the principle of “giving control to user”, since in such cases it makes no sense to give him/her control over something he/she would not need. On the contrary, this is a case where extra control means extra concerns or extra problems for the user to solve. For example, if the user wants to move a dimension and invokes the corresponding command for it, they never want and never think if anything other than dimensions can/should be picked. Thus, it will be extraneous to give users the possibility to change the correct natural setting. Another typical example is the command "delete points". Here again, it makes no sense to be able to set object-filter to members, as in Bocad-3D!

Defining such settings within the software adds to a low-risk environment as the users can see that the software automatically blocks the lines where the user may make a mistake. A thorough analysis on each command can guarantee the right results and users will never even realise that they have no control in changing such settings.

- User-defined combinations

The user must have complete control to determine/select any combination of object types. In other words, it must be possible to select one or more object types together and selecting one object type should not cancel another object type.

- Keyboard accelerators

It is useful to define a keyboard accelerator for frequently used settings such as: any/all, parts, bolts, components (as in TeklaStructures), and points. The best choice for this sake will probably be the first letter of the word (in English) for each object type such as: All, Parts, and Bolts.

4.4 Adjustment Handles

As it is common in all window applications, adjustment handles allow resizing or reshaping of elements. They appear as small boxes on the corners and edges of a selected object (see Fig. 4-18a). Hovering on a handle results in changing the cursor shape indicating the possible operation to the user. Typical cursor shapes are shown in Fig. 4-18b. These elements highly contribute to direct manipulation style of interaction within a program.

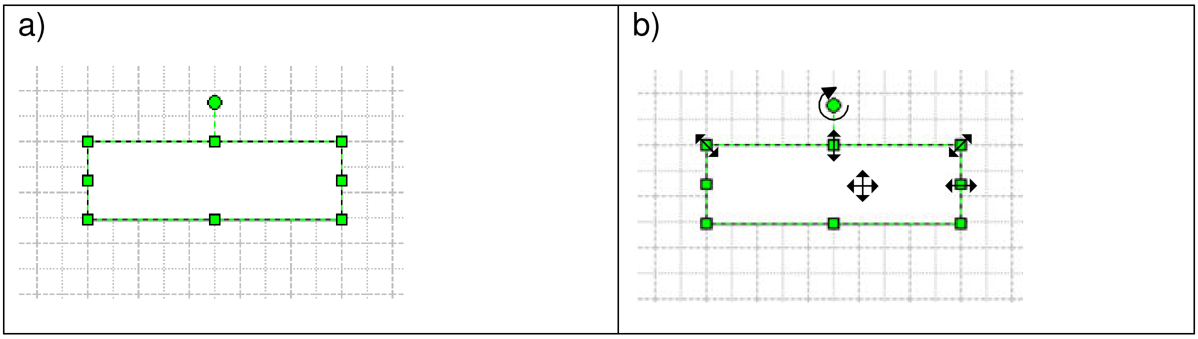


Fig. 4-18: Adjustment handles in Microsoft Visio

In CAD programs, these tools offer a splendid chance not only because all users are familiar with their function but also because they allow efficient performance of many useful operations on an object. Chang (2001) introduced application of these elements in steel detailing CAD programs [54]. Pegels and Weckmann discuss their significance in net-based collaboration (2007) [75]. Recently Schulten developed a prototype (Fig. 4-19) which demonstrates how these interface elements can be successfully implemented in CAD programs (2007) [15].

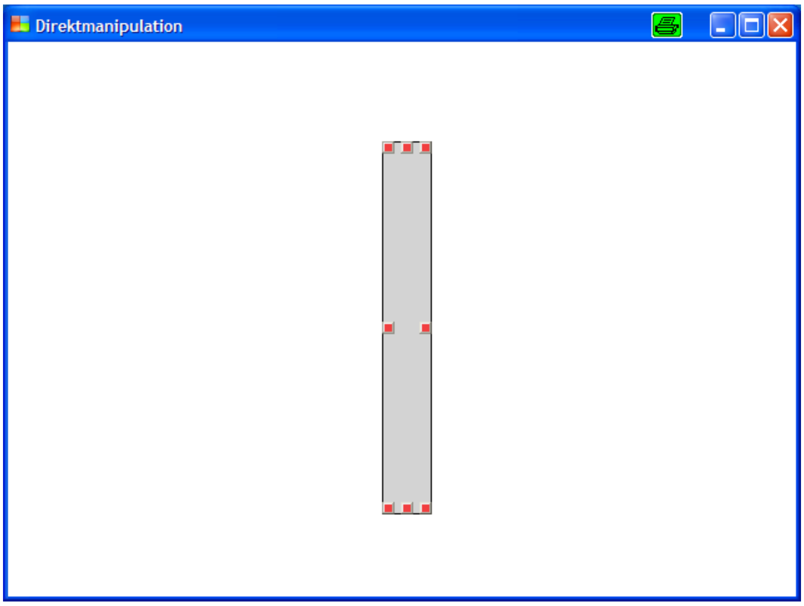


Fig. 4-19: A prototype for adjustment handles on objects, Schulten [15]

Although in Bocad-3D there are a few instances of direct manipulation interaction, adjustment handles are basically not used—probably due to earnest implementation and programming problems for interactive handles. In TeklaStructures, these tools are somehow missing in modelling mode, but the program uses them effectively in drawing mode.

Adjustment handles can have variety of important applications in a 3D CAD program for steel detailing; they can be used on both 2D and 3D objects. They can even be used to determine view borders (or as it is proposed in section 5.2, to determine section's region). As a contribution to previous work on these tools, more detailed guidelines are given here concerning their design. Furthermore, section 5.5 will present a practical example of how these elements can be used in a CAD program for steel detailing.

Graphical/Display Concerns

Feedback

As a basic prerequisite for interaction with these tools, a real-time feedback must be provided to indicate the ongoing changes. Fig. 4-20 presents two different styles for this. The second style seems to be more advantageous as it presents more information to user showing a trace of original status.

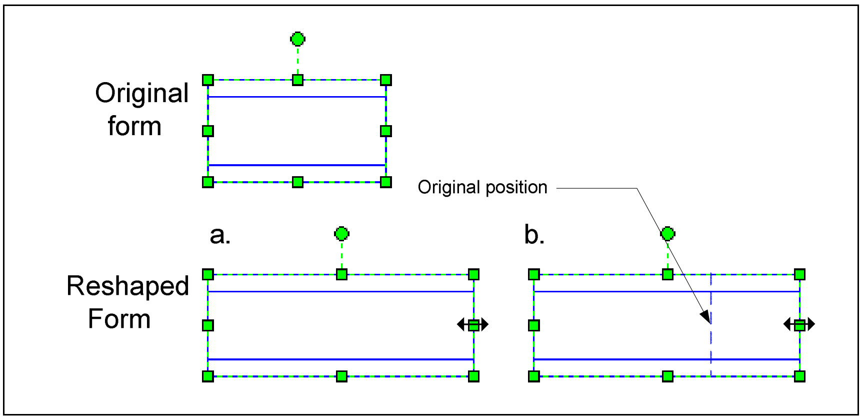


Fig. 4-20:
Two styles for direct-manipulation using adjustment handles

Alteration of the mouse cursor

As the experience in TeklaStructures as well as AutoCAD shows, it does not seem to be necessary to display common changes of the mouse cursor (Fig. 4-18b) for adjustment handles because the users of CAD systems are likely to be familiar enough with these tools and their possible operations. Therefore, the mere display of the handles on objects would be sufficient.

Zooming level issues

Presence and availability of adjustment handles must be a function of the zooming level. This means that in views with a low zooming level, either not all adjustment handles must be displayed or not all corresponding operations must be available. In views with low zooming level only move or copy must be available since these are the only logical choices that the user may want to do at low zooming levels using these tools. Violating this rule gives rise to a problem seen in TeklaStructures when editing drawings as shown in Fig. 4-21.

At the presented zooming level, although the user intends to move the selected view to another location on the drawing, he/she fails to do so because the program misunderstands the user's intention and instead resizes the selected view. Whereas resizing is an operation, which the user does/can do more effectively in much closer zooms where he/she can keep track of the ongoing changes. At this zooming level, the user may just want to move the whole view.

Adjustment Handles for Members

Fig. 4-22 presents adjustment handles on an I-profile together with their possible actions that they offer. These handles facilitate many adjustment operations direct on the profile. In addition, Chang proposes an extra handle to be displayed on the profile for changing its size [54]. Although this seems to be an interesting idea, it has a drawback that it takes screen space for an infrequent task.

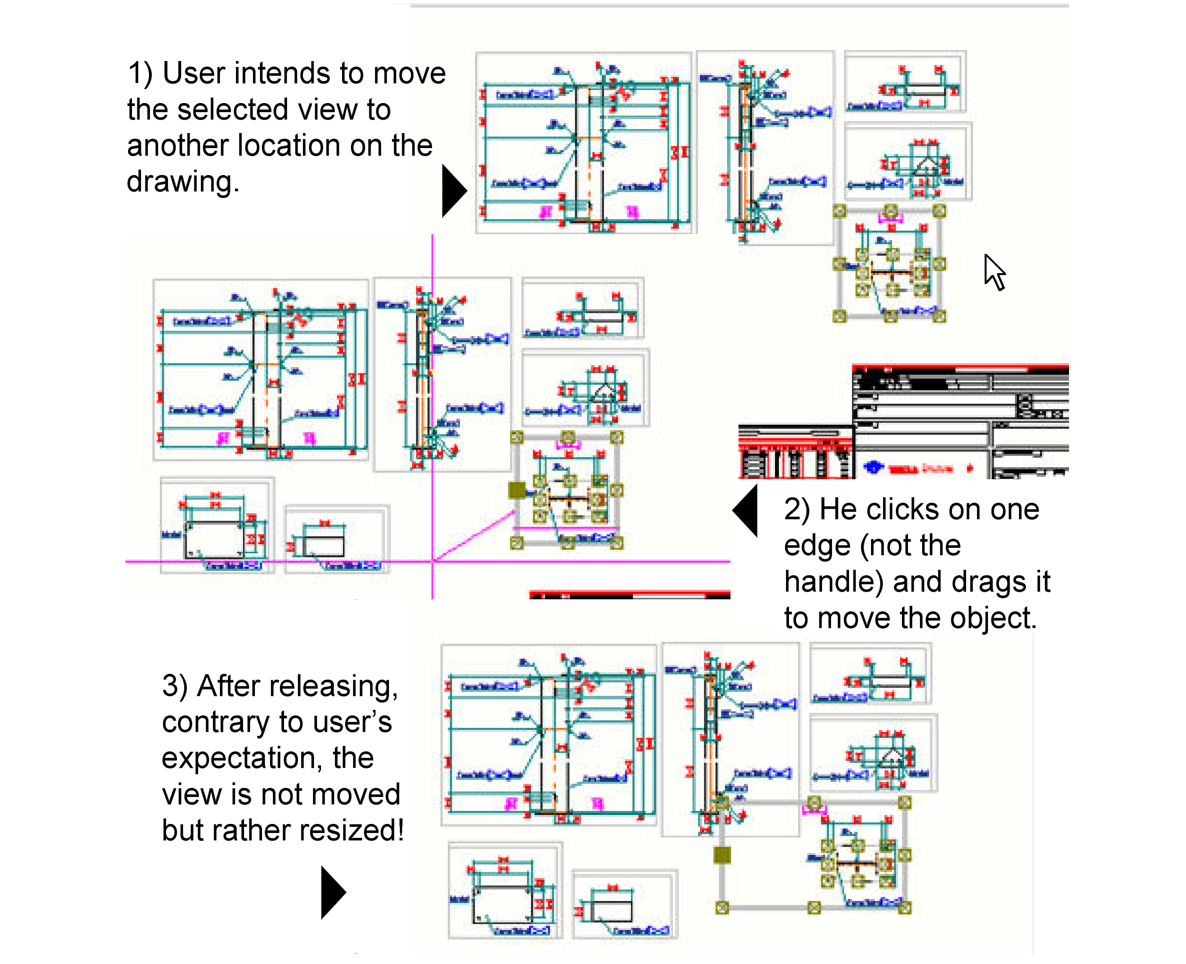


Fig. 4-21: Shortcoming in TeklaStructures regarding adjustment handles

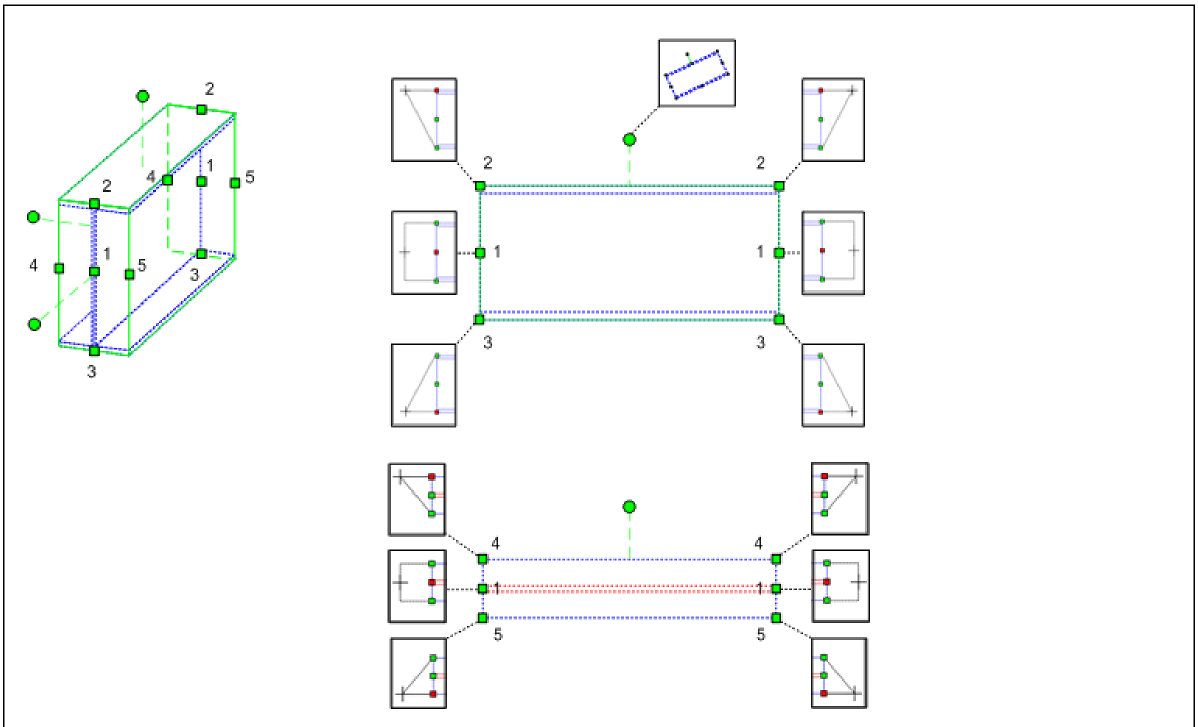


Fig. 4-22: Adjustment handles and their corresponding operation on an I-profile

It should be noted that for members with irregular geometry, such as a profile with an inclined cut at one end, adjustment handles must appear on corners of their outline perimeter. This may contradict users' general expectation of these elements in window applications because they normally appear on perimeter of a rectangle surrounding the object. However, the required accuracy in a CAD program necessitates that the user has direct access to points of interest. Thus, this idea is advisable because it improves efficiency by increasing user's control.

Moving/Copying and Rotation

It is a common practice in windows applications to allow copying and moving of elements by direct manipulation. Normal mouse cursors for "move" "copy" are ✥ and ✥+, respectively. This can be offered also in CAD programs for steel detailing provided that adequate control as well as graphical feedback is given to guide the user in selecting origin and destination points. Moreover, as in Microsoft Visio, it is useful to introduce an adjustment handle for rotating the element. The rotation centre must also be adjustable by the user through direct manipulation. This is shown in Fig. 4-23.

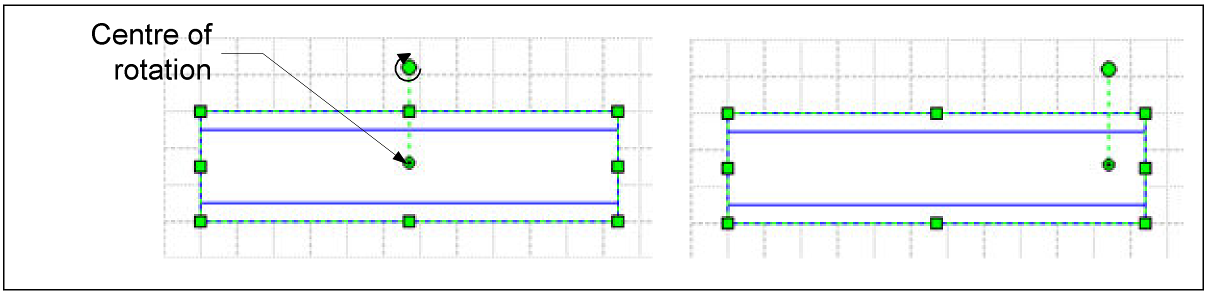


Fig. 4-23: Adjustment handle for rotation in Microsoft Visio

Extended Controls

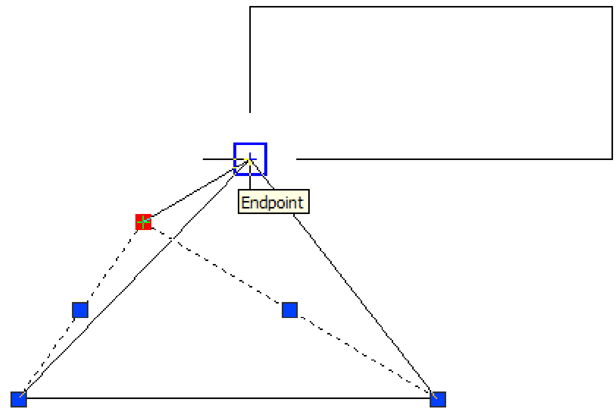
Tasks that are facilitated by adjustment handles in many window applications do not basically require a high accuracy and as a result these are done typically by freehand operation and subsequently controlled by visual inspection. By contrast, in a CAD program for steel detailing, every single operation, including resizing and reshaping by direct manipulation, demands a high precision. As a result, free hand operations and visual inspections alone will indeed fall short of goals. In this aspect, AutoCAD offers two adequate tools that facilitate complete control over the results:

- Use of adjustment handles together with object snap: As the user drags a handle, object-snaps actively offer possible destination points, see Fig. 4-24a. In this figure, the user intends to reshape the triangle such that its top corner coincides with lower left corner of the rectangle. In a 3D CAD program for steel detailing, this option can be especially useful for adjusting a member to another member.

- Allowing numerical entry via keyboard: After the user selects a handle, the program allows a keyboard entry of a value in the direction that the user determines by moving the mouse as shown in Fig. 4-24b. This setup is especially useful for shortening or lengthening an object by value, which is often the case.

To implement the latter operation in CAD programs, which normally do not accommodate a command-line, the solution shown in Fig. 4-25 is proposed: When user selects an adjustment handle, a text-box appears which not only displays the incremental change of length through dragging, but also allows the user to enter a numerical value. Same as AutoCAD, this must be done automatically as the user enters the value on the keyboard and presses the enter key.

a. Object-snaps together with adjustment handles



b. Allowing numerical entry together with adjustment handles

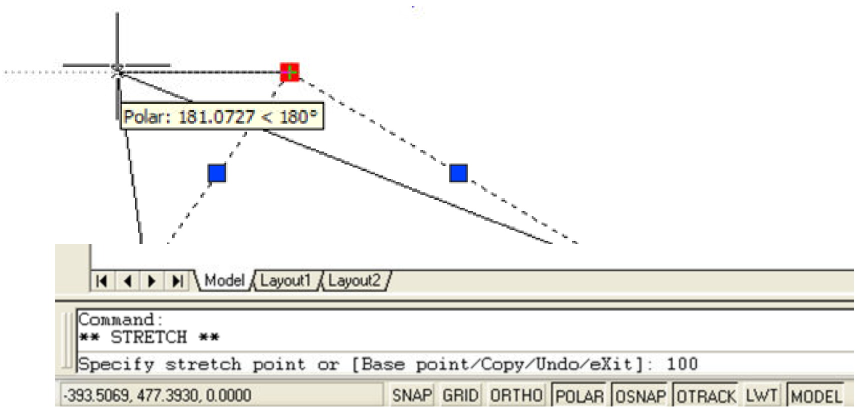


Fig. 4-24: Object-snaps or numerical entry with adjustment handles in AutoCAD

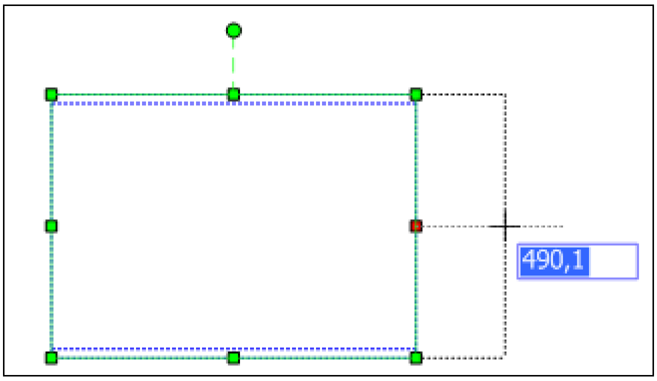


Fig. 4-25: Proposal for accurate use of adjustment handles

Selection of Multiple Objects

When multiple objects are selected, a control must appear that embraces all selected objects as shown in Fig. 4-26. In this case, adjusting the selected controls via handles must result in resizing or reshaping the selected elements by preserving their proportion to each other; for example, when an I-profile together with its end-plate are selected, lengthening the control must result in lengthening the profile only accompanied by the end-plate. Fig. 4-26 presents further examples for this case.

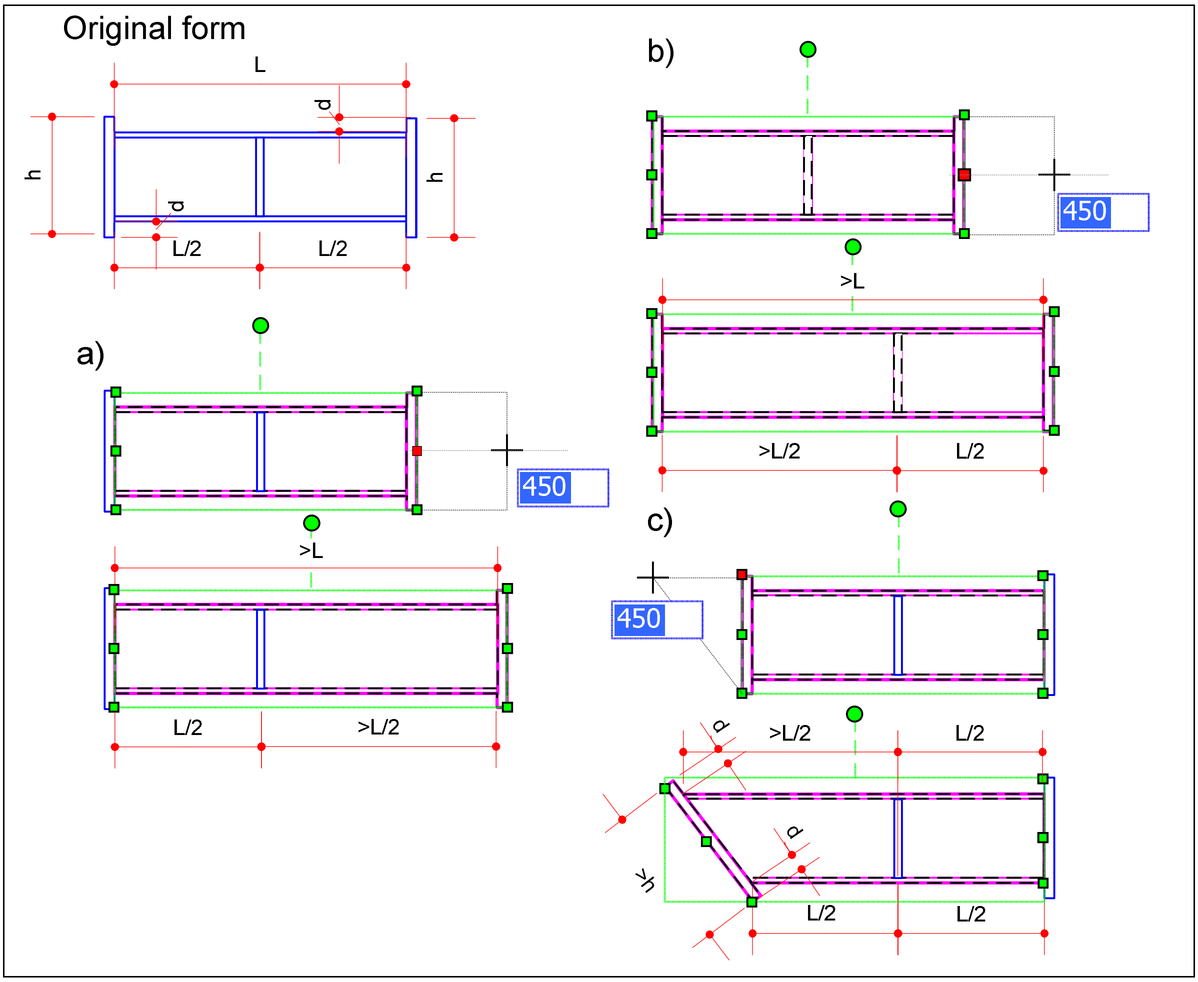


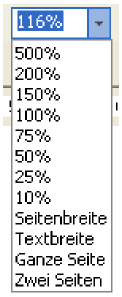
Fig. 4-26: Adjustment handles working on multiple objects

4.5 Text-boxes, Drop-down Lists and Combo-boxes

Text-boxes, lists and combo boxes are typical GUI components used in any CAD program. These elements are basically used to receive input from the user. Different types of text-boxes are shown in Fig. 4-27. In general, these elements are composed of a caption title, input field, and in case of lists or combo boxes, a drop down list.

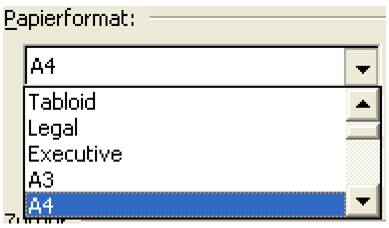
The purpose of this section is to provide solutions that enhance the efficiency of working with these elements. Moreover, it is intended to emphasis that even in these elements, it is sometimes possible to avoid text, as in lists with graphical elements.

Text-box: Only textual entry; No selection.



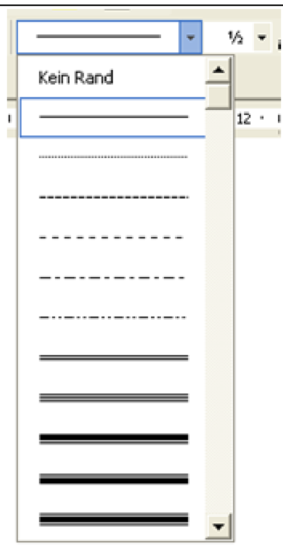
Combo box :

Textual entry as well as selection



Drop-down lists with textual elements:

Only selection; No textual entry.



Drop-down lists with graphical elements:

Only selection; No textual entry

Fig. 4-27: Text-box, lists and Combo box with different input alternatives

General Design Considerations

In design of drop-down fields, the following general considerations should be taken into account:

- Drop-down lists have the advantage that they eliminate any chance of typographical errors by the user. However, since they force the user to search for the desired option, it becomes troublesome when the list is relatively long, such as a list of profiles. In such cases, combo boxes can function more efficiently because the user has the possibility to type down the desired information instead of searching for it in the drop-down list. Therefore, combo boxes should be applied as much as possible when dealing with long textual lists.

- Combo boxes must support an auto-completion function. Moreover, application of "wild cards" such as "*" and "?" must be possible in text fields as well as combo boxes;

- Instead of providing a complete list, minimum relevant elements must be displayed to the user; for example, if the user wants to select a profile for a search function in the least, only the applied profiles in the model must be listed. Fig. 4-28 shows an example of such shortcoming in Bocad-3D. In this example, since the list does not support an auto-completion function, the user will have to look for the desired profile in a long list. In practice, what happens is that users barely use the list and instead type the desired profile in the given field, which makes the application of the list almost aimless. In this case, although enlisting all possible profiles is easier from the programming perspective, it is confusing and inefficient from the user's perspective and hence, must be avoided.

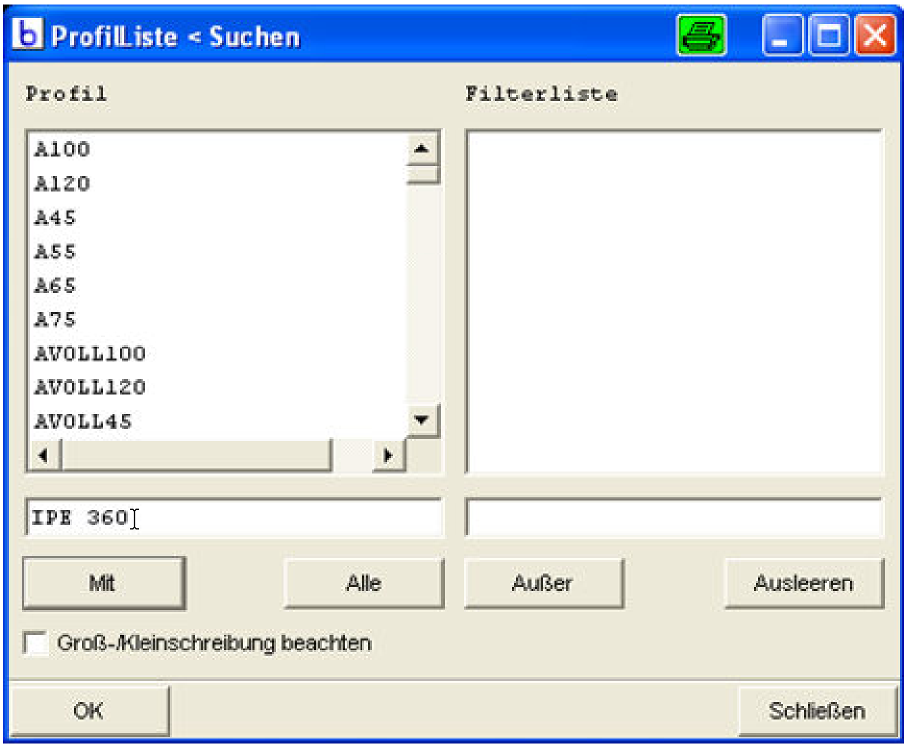


Fig. 4-28: Inefficient selection box in Bocad-3D lacking auto-completion

- In case of names or properties longer than the text-fields, drop-down fields must adjust automatically to cover the whole list, see Fig. 4-29a. Otherwise, it is not possible to read the selection list completely. An alternative to this could be the use of resizing handles, as in Fig. 4-29b. Fig. 4-30 shows an example in Bocad-3D where neither of these methods is used and as a result, the user has no chance to read the options completely; in this example, he/she can not even understand the difference between the shown options—making it difficult for him/her to select one.

It should be noted that the use of horizontal scroll bars (as in Fig. 4-31) would not be very effective either, since the user can see only one part of the option at a time.

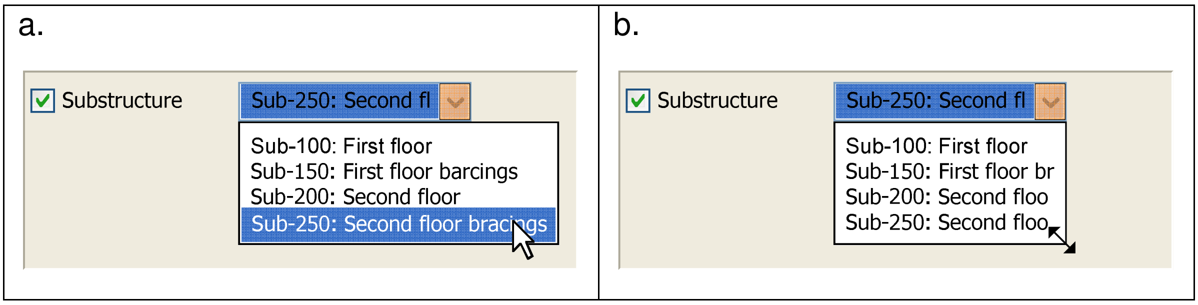


Fig. 4-29: a. Automatic adjustment; b. Manual adjustment using resizing handles

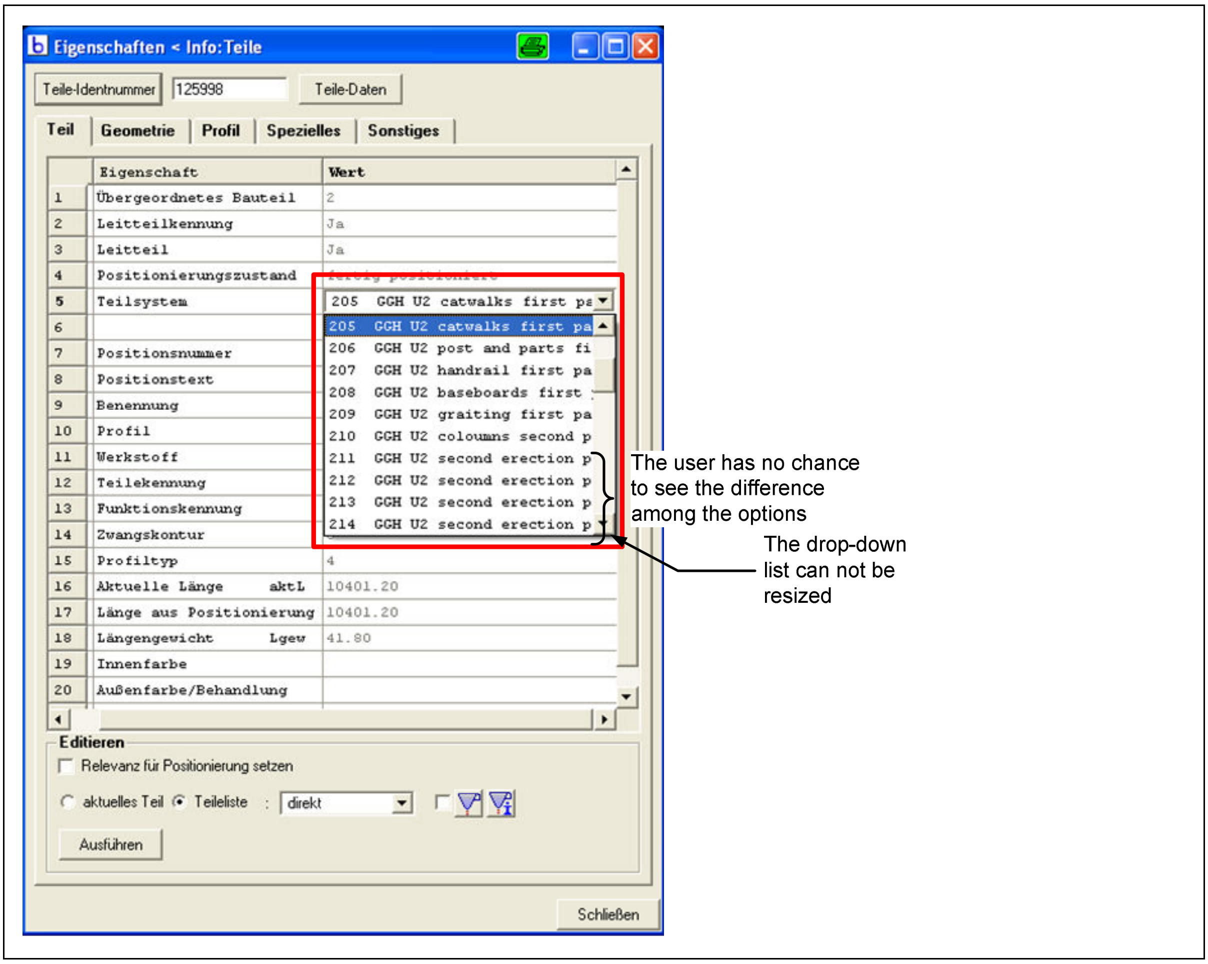


Fig. 4-30: Lack of resizable drop-down list in Bocad-3D

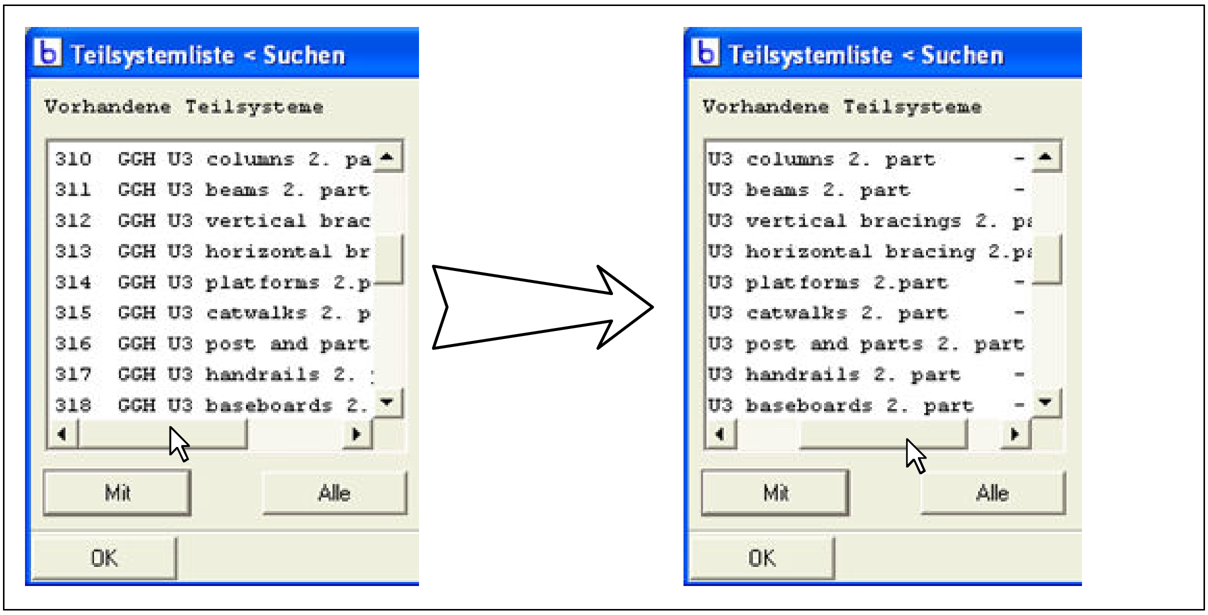


Fig. 4-31: Horizontal scroll bars for lists

Optimised Design Considerations

In order to optimize the user's interaction with these elements in a CAD program, possible ambiguities must first be addressed. At the very first sight, due to their extensive application in all GUI-based programs, users are familiar with these elements. The visual characteristics of these elements will not be vague or confusing for the users. However, what cannot be clear is the following:

1. Of which entity is the required information, i.e. what should be given in the box?

2. Which format for information is acceptable?

To address these ambiguities in the framework of a graphical text-free system, at least one or a combination of the following solutions should be applied:

Combo boxes: Combo boxes with their drop-down fields are useful since the users interact fairly intuitively with them, i.e. dropping down the menu is the very first reaction and as a result the first question will be addressed. By selecting the desired information from a list, the format will be automatically correct. As a new idea, use of combo boxes with multiple selection possibility offers the maximum efficiency. An example is shown in Fig. 4-32.

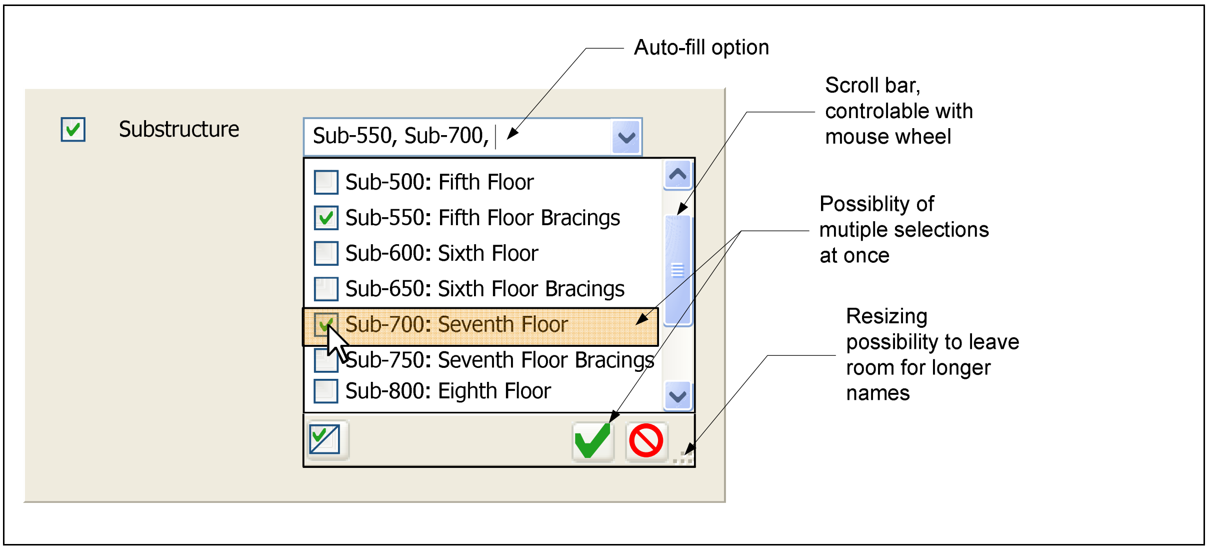


Fig. 4-32: Combo boxes with multiple selections

Use of graphical elements in the list: This is one of the effective methods to optimize the user's interactions. An extensive use of these elements can be seen in the components of TeklaStructures resulting in a productive user's interaction, see Fig. 4-33.

Use of graphical caption for the drop-down menu or the text-box: When possible, a graphical caption will have the advantage to address both interaction questions at once. Fig. 4-34 presents a comparison of both methods.

Use of global engineering legends: This is another effective method to address the possible ambiguities in user's interactions with text-boxes or drop-down lists. Examples are shown in section 5.3.

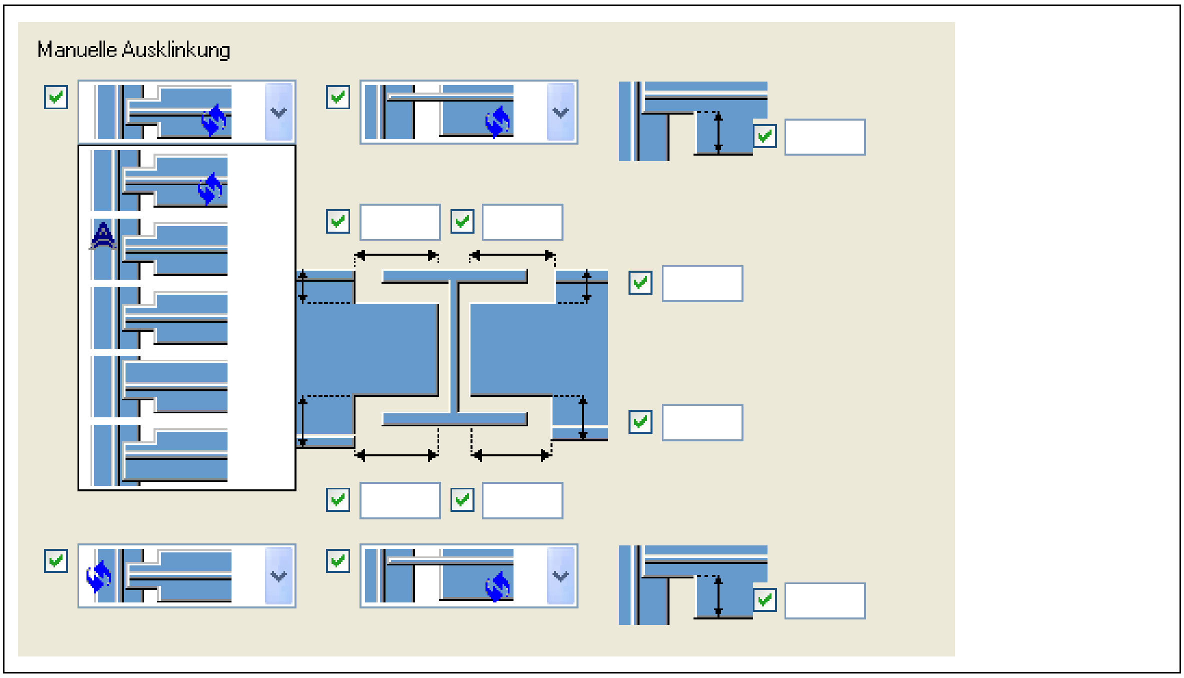


Fig. 4-33: Drop-down lists with graphical elements in TeklaStructures.

Use of format examples (Quick info): Format examples (Quick info) can be very useful to address the question of format, as shown in Fig. 4-35.

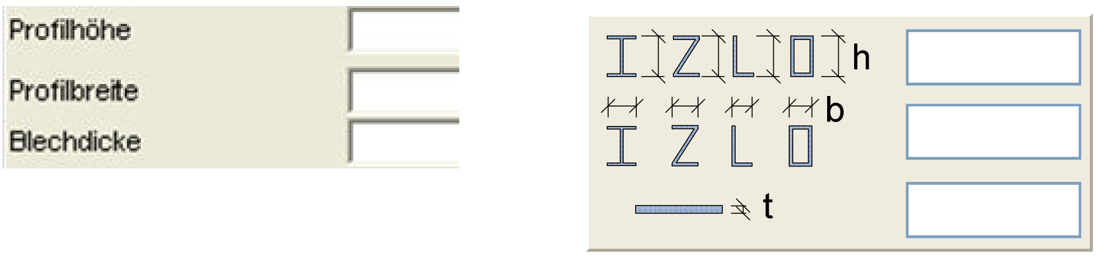


Fig. 4-34: Textual captions vs. Graphical captions for text-boxes

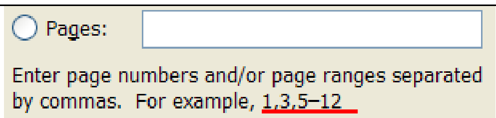


Fig. 4-35: Quick info in Microsoft Word

Textual captions with format examples: As the last solution, a textual caption should inevitably be used, but it must at least include a format example. In such cases, format examples can also offer some idea about the entity of the desired information. Fig. 4-36 shows a comparative example.

“Generation date” in search function in Bocad-3D

Fig. 4-36: Format example

5. Layout Prototypes

In this chapter, a number of complete design layouts are developed for some of the most frequent commands in a high-performance CAD system.

5.1 View/Display Settings

View/Display setting is one of the most frequent preparatory commands. Major tasks to be covered in this function are:

- Defining/adjusting the view borders;
- Adjusting the elements' display/marks;

As next, each of these tasks will be elaborated and their corresponding solution will be provided.

Defining/adjusting the view borders

This means that the user intends to extend or reduce the borders of the available view. The interface provided in both programs for this task is basically textual. However, a text-free solution for this task is relatively easy to realise, as shown in Fig. 5-1. Moreover, adjustment handles can be used very effectively in x-y directions: By clicking and dragging available handles on the view borders, the view extends or reduces correspondingly, while at the same time the user receives a real-time feedback.

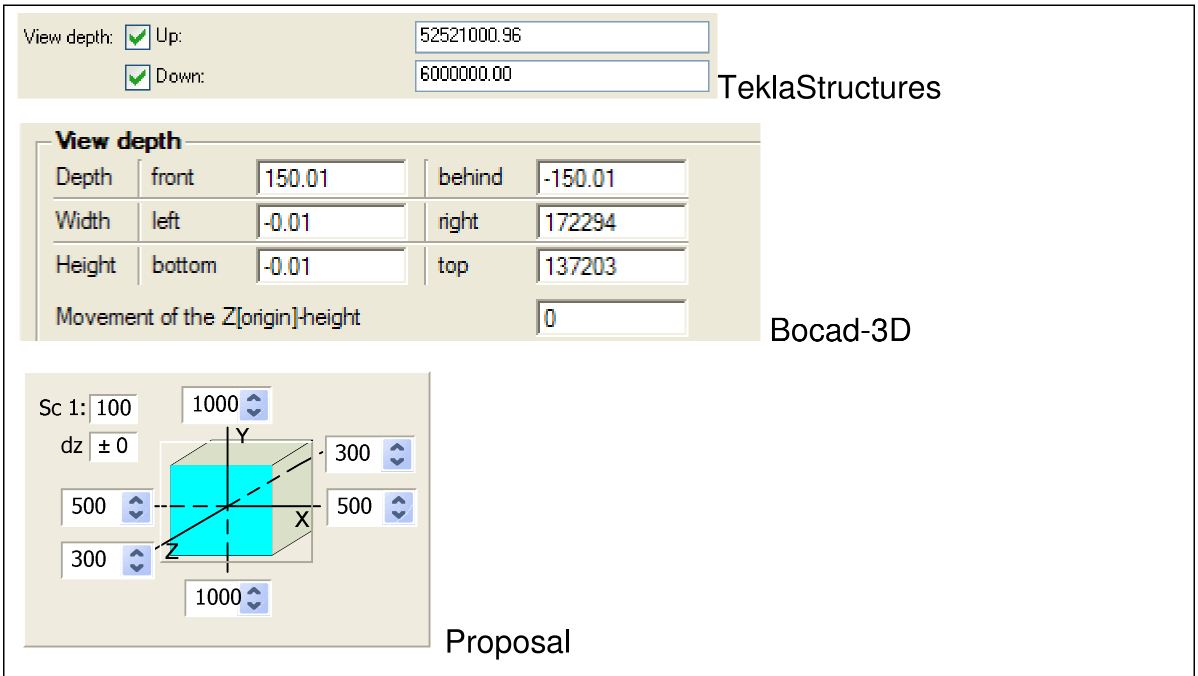


Fig. 5-1: Alternatives for defining/adjusting the view borders

Adjusting the elements' display/mark

This is a frequent task which is also needed for interactive generation of drawings or details. Fig. 5-2 and Fig. 5-3 present examples of this feature in Bocad-3D and TeklaStructures.

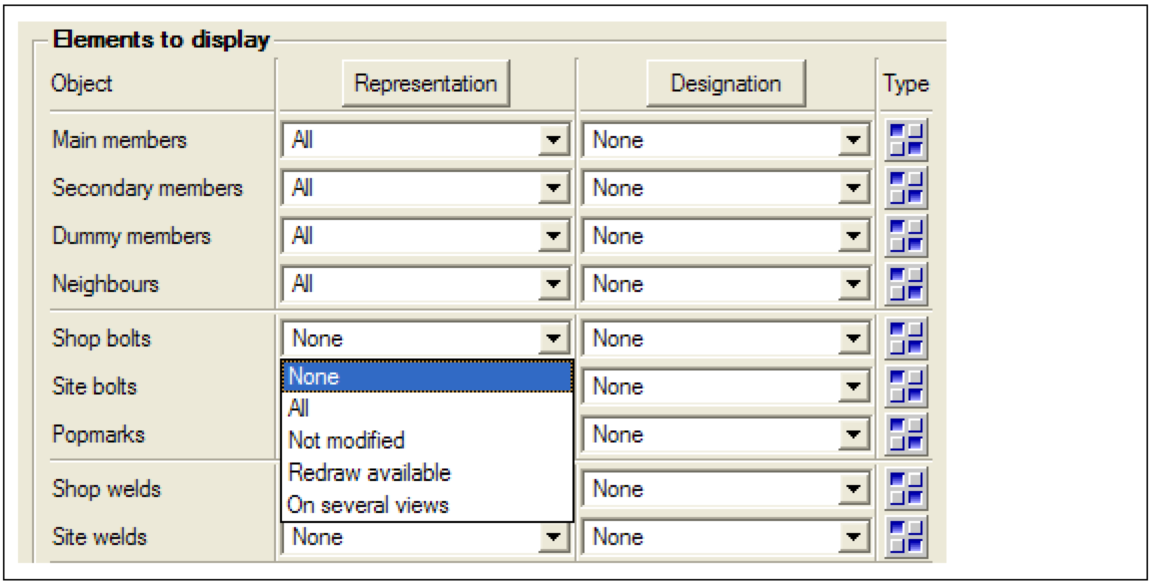


Fig. 5-2. Dialog box for "View settings" in Bocad-3D

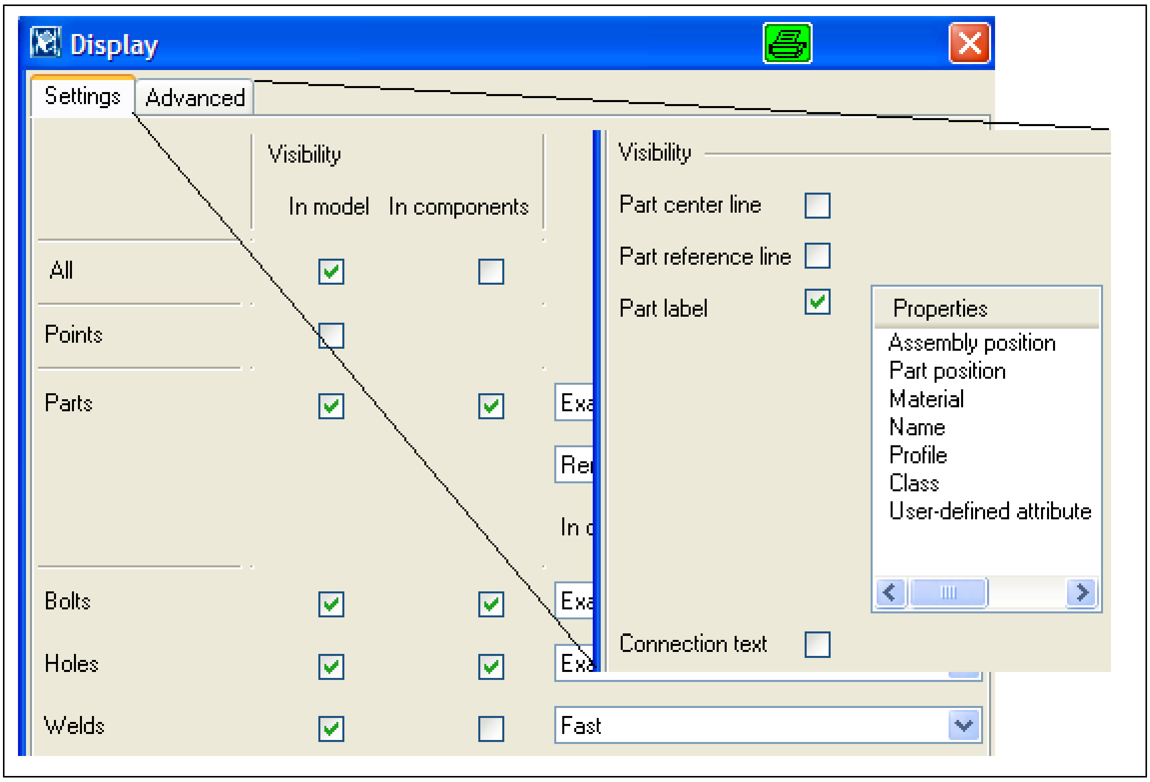


Fig. 5-3: Dialog box for "View settings" in TeklaStructures

In both programs, this feature is textual and language dependent. It will certainly take a lot of time, especially in Bocad-3D with all its comprehensive options in this feature, until users get accustomed to this dialog box and work efficiently with it. Moreover, nearly all options provided in the first layer of these dialog boxes have the same level of transparency to the user regardless of their frequency or functionality; for example, in Bocad-3D an option such as "DB graphics", which is barely used, is presented in the same layer with a frequent option such as "Main members".

In TeklaStructures there is no way for user to see mark of bolts or welds. This leads to a shortcoming that the user will not receive any direct feedback when creating these elements. Moreover, the user is deprived of having an overview over the elements' properties, which makes controls and quality checks relatively inefficient; for example, in order to control a series of bolts, it is necessary to click on every single set of them. Furthermore member marks are not displayed by default; in addition, when user wants to have them displayed, the feature is not flexible at all because the marks will be shown only in yellow, without leader lines, only at the middle of the member and not necessarily without overlapping. As a result, this feature can not be considered very helpful in TeklaStructures.

Another noticeable drawback regarding this task in Bocad-3D is the placement of the "filter function" in the third layer relative to the working layer; it requires at least four mouse clicks to invoke the command, as shown in Fig. 5-4. Hence, despite its frequent use, users have to go through a long mouse journey to invoke the command. TeklaStructures offers a better access for this function: Instead of going to screen borders for clicking the corresponding icon as in Bocad-3D, user can load the function simply by double-clicking on the view. Alternatively, he/she can load the command also from the mouse right-click menu. Moreover, the embedded filtering function is just one click away, after the command box is loaded.

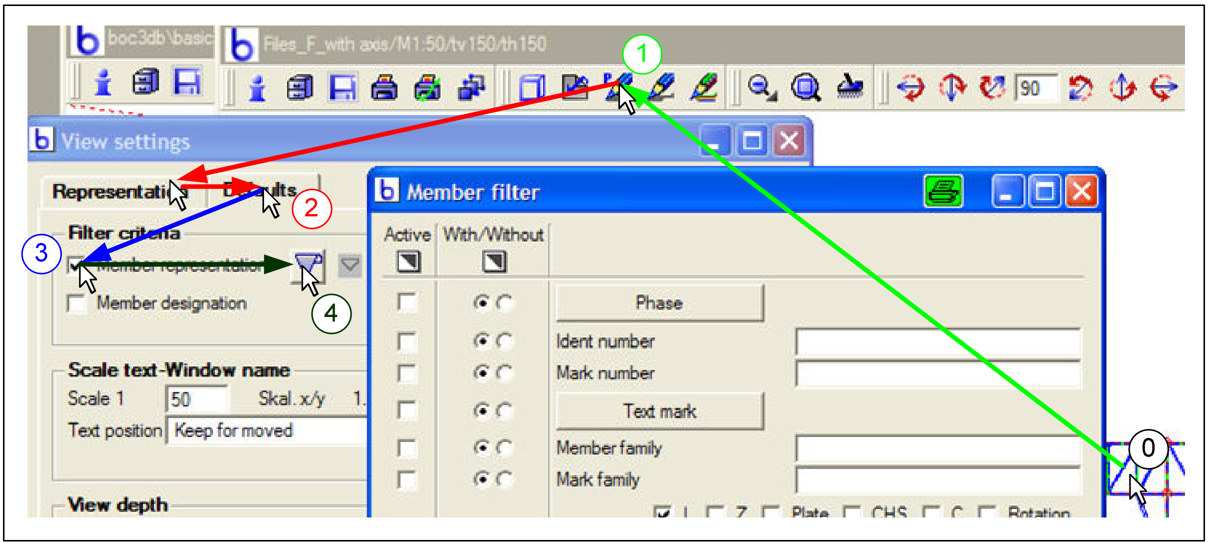


Fig. 5-4: Mouse journey for display/mark filtering in Bocad-3D

In an attempt to provide a text-free dialog box for this feature, several interesting icons and metaphors can be used such as bolts, welds, members, or a schematic text with a leader line to indicate the mark for different elements. Moreover, a metaphor of an on/off light bulb (/), as in AutoCAD, can be used effectively to indicate activating/deactivating the target mark/representation. In addition to these two features, Bocad-3D offers two more helpful features (redraw available, do not refresh), which can become especially helpful for manual editing of drawings. For these features, the refresh icon from Internet Explorer can be used to indicate "redraw/refresh available": (); () can be used to indicate "keep existing or do not refresh". Although both of these latter features may be hard to guess for novice users, once explained, they can find and use it quickly since the nature of the presented metaphors will help them recognise the function more quickly in comparison to textual options only. Moreover, a textual hint presented in the mouse tool-tip can always be helpful. This is shown together with the proposed solution in Fig. 5-5. In the proposed solution, the beforementioned shortcomings have been overcome:

- The textual/language dependency is significantly lowered;
- Most options are readily recognised by graphic symbols/metaphors even by novice users;
- Based on the task analysis, more frequent features such as filtering are now on the top level while at the same time providing adequate visual cues to explore inner layers for extended features.

Since the features provided in this command are very often used, it is a good practice to provide them also in the working view. As in Microsoft Visio (Fig. 5-6), this can be done by providing small windows with auto-hide/floating feature on the screen borders. This will have a further advantage that user will not need to look for such features as he/she has the most frequent ones right at his/her disposal. For this purpose, text-free solutions in Fig. 5-7 are proposed.

Finally, even for more complicated concepts regarding the marks, it is possible to provide a graphical solution for the user; for example, to save space and/or enhance legibility either in the drawings or during construction of the model, users will sometimes need to set one the following functions:

- Marks vertical or horizontal;
- Marks on the element edge if possible;
- General location (left, right, top, bottom) of the mark relative to the object;
- Merging same marks in the specified area.

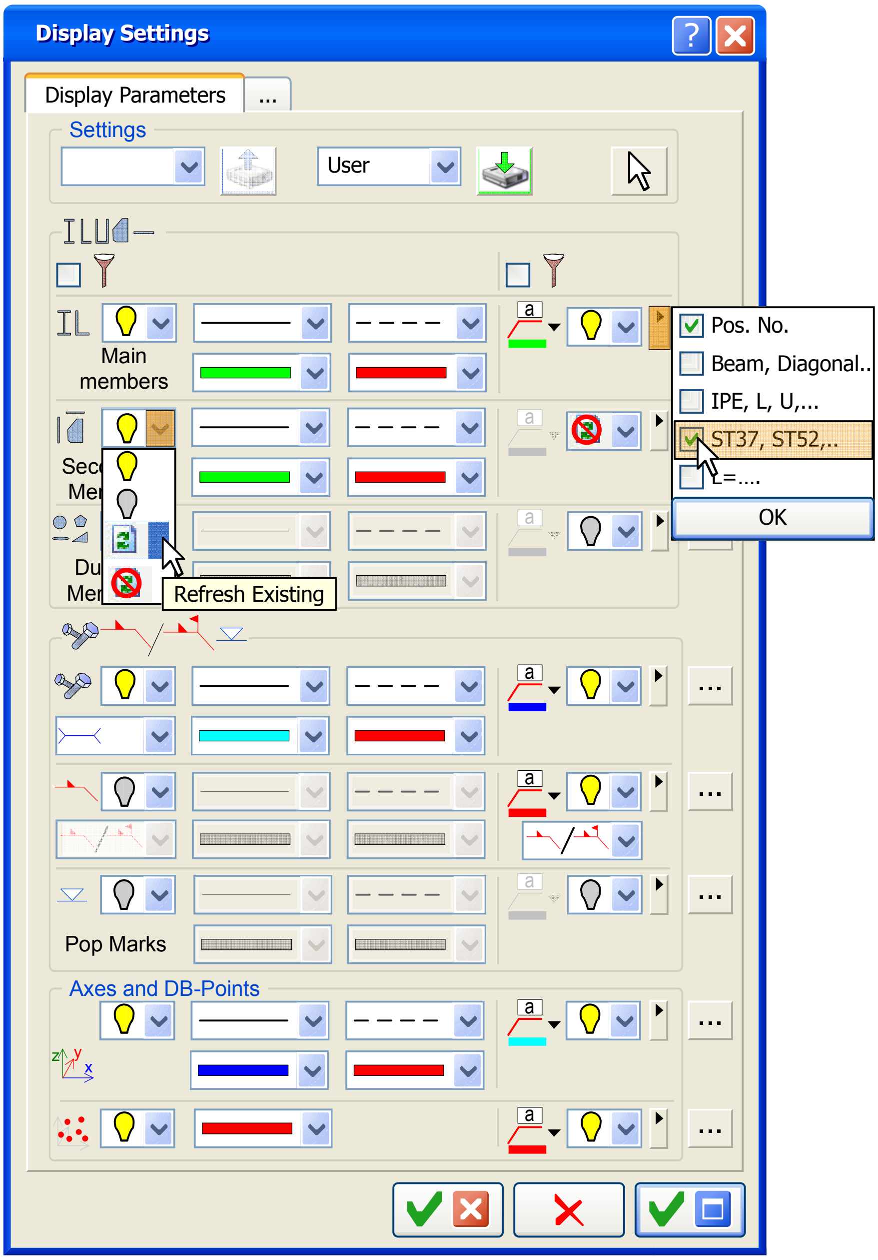


Fig. 5-5: Proposed solution for “View/Display Settings”

Fig. 5-8 through Fig. 5-11 present these features in both programs as well their proposed solutions in this work. Bocad-3D has a textual approach for these tools. TeklaStructures offers better solutions in two cases shown in Fig. 5-9 and Fig. 5-10. Yet, they may seem to be slightly ambiguous to a novice user, as he/she may not recognise the edges of the profiles from the available graphic. The proposed solution, therefore, is likely to present more clarity in this respect.

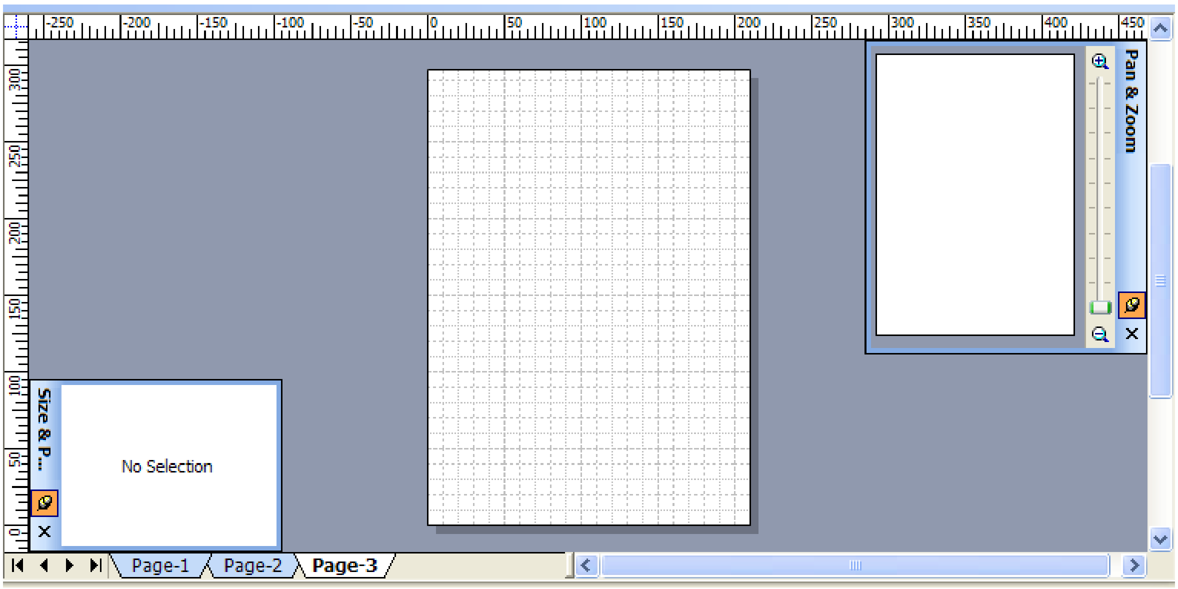


Fig. 5-6: Use of floating windows in the working area, Microsoft Visio

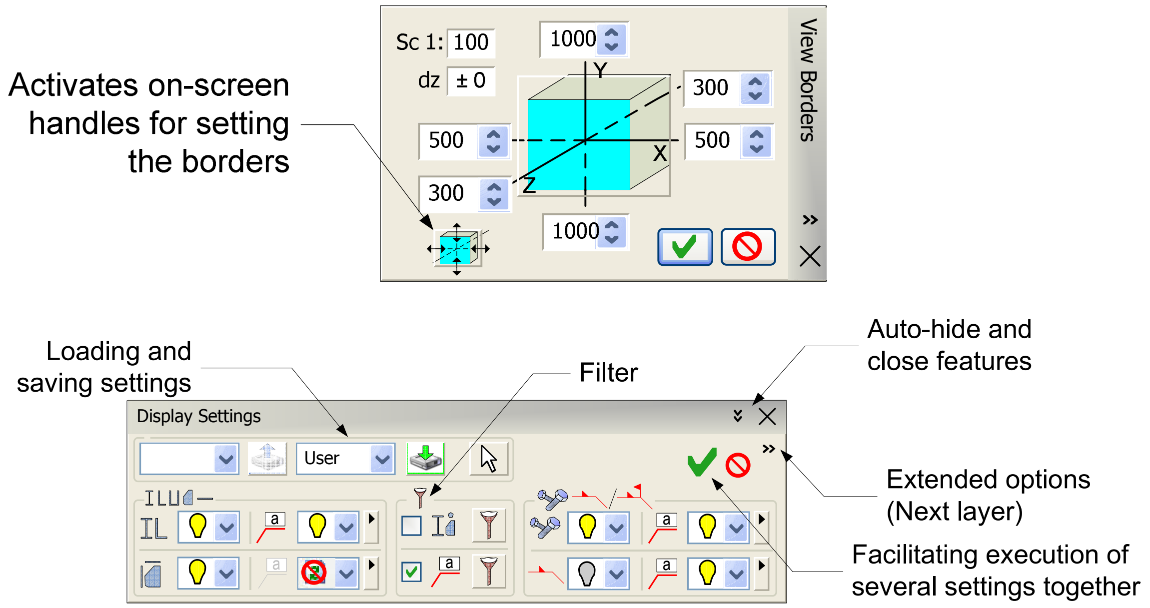


Fig. 5-7: Proposed floating windows for adjusting view borders/properties

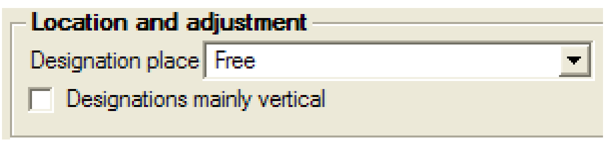
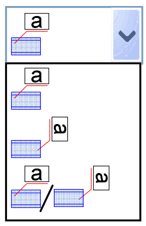
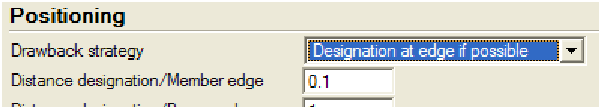
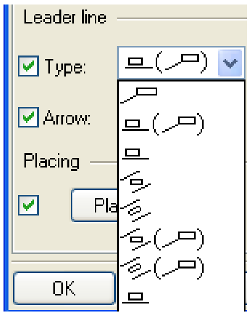
Bocad-3D	TeklaStructures	Present work
	None	

Fig. 5-8: Alternatives for “setting mark vertical or horizontal”

Bocad-3D



TeklaStructures



Proposal

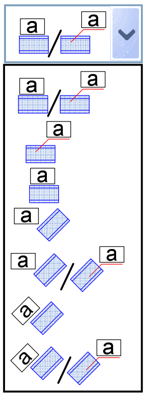
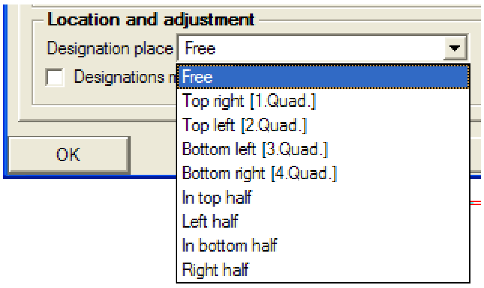
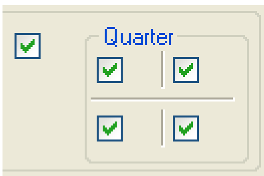


Fig. 5-9: Alternatives for “setting marks on the element’s edge if possible”

Bocad-3D



TeklaStructures



Proposal

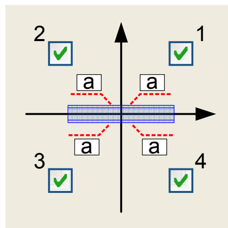
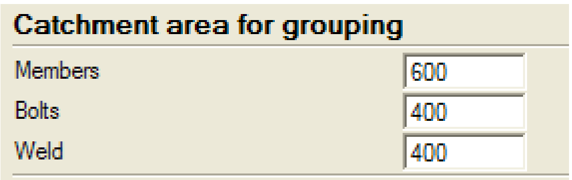
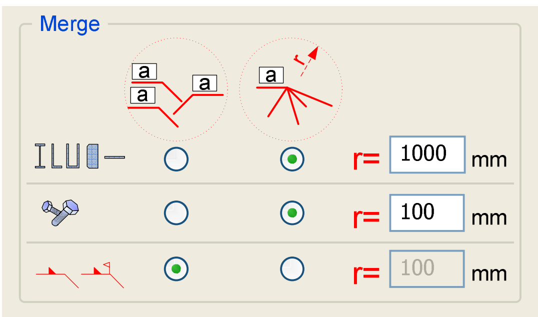


Fig. 5-10: Alternatives for “Setting general location of mark relative to the object”

Bocad-3D



TeklaStructures



Proposal

Fig. 5-11: Alternatives for “merging marks”

5.2 Creating a Section

As results of the survey in this research indicated, this command is one the most frequent preparatory commands and is basically used both in modelling and drawing mode.

Both programs suffer from some drawbacks at this command. The main problem with this command is how to convey the user's purpose to the program about the section's extension (vertical or horizontal) and its direction (left-right, top-bottom), which makes four alternatives—excluding inclined forms. Moreover, width and depth of the section also have to be determined so that the program can subsequently generate an automatic output. To do this in Bocad-3D, the user has to specify the section type at first in the command's contextual dialog box before creating it, see Fig. 5-12. Other extensions are determined by giving numerical values in the text fields. The process has to be repeated or at least controlled each time when the user wants to create a section. This results in distraction from the task. Moreover, except for very expert users, it is difficult to insert sensible/exact numerical values in the provided text-fields.

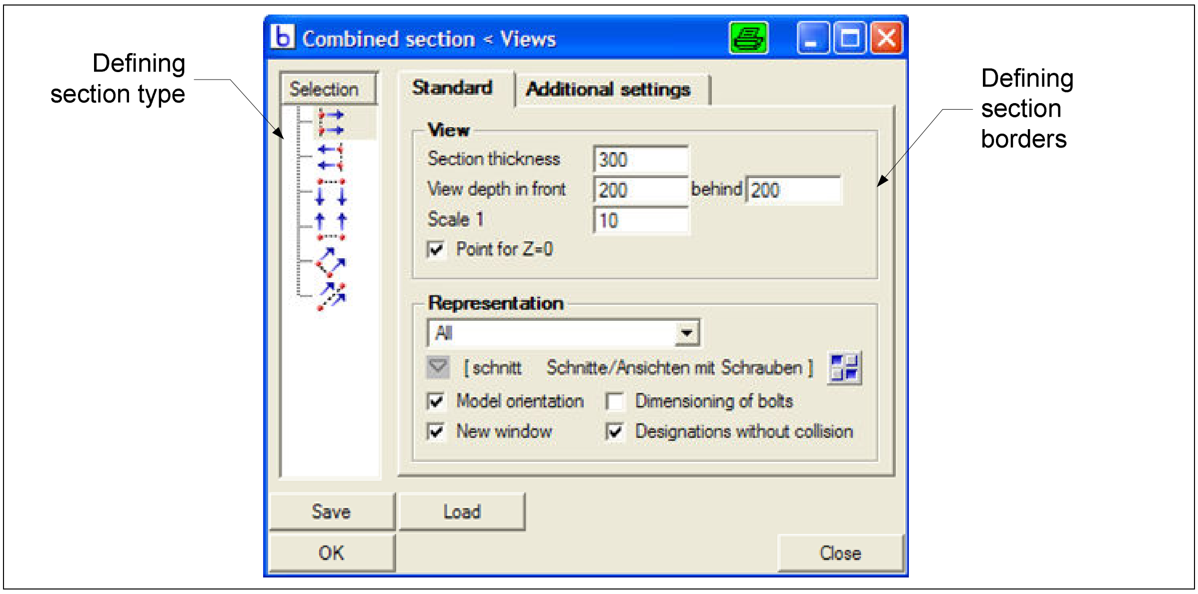


Fig. 5-12: Dialog box for creating a section, Bocad-3D

In TeklaStructures, by use of adequate graphical feedback, this is done more intuitively: After invoking the command, the user can directly click and drag on the screen and control if the desired result is being achieved—without any controlling in-between. In addition, section's width and depth can be given by direct manipulation, as shown in Fig. 5-13.

The downside is that the program idiosyncratically forces the user to follow the "right-hand rule" when drawing a section. This is a typical example of "forcing users committing unnatural tasks" because it does not belong to the task domain to consider the right-hand rule while drawing a section. The result is, for each of

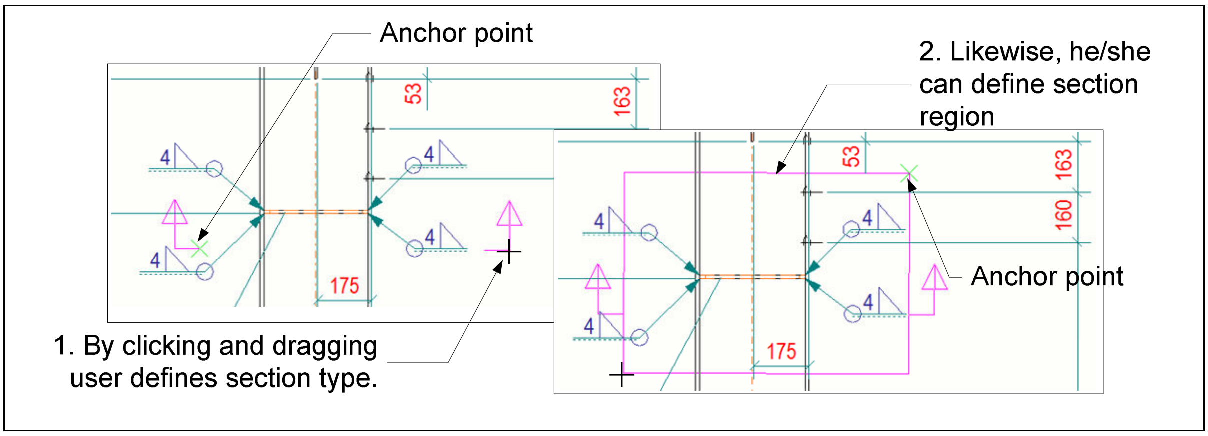


Fig. 5-13: Creating sections by direct manipulation, TeklaStructures

four possible alternatives of a section, there is only one correct first point position; should the user select "a wrong position", there is no way to achieve the desired result. Another drawback of this setup is that it forces users to perform unnatural hand movements. As a natural rule, humans normally draw vertical lines, from top to bottom and horizontal lines from left to right. This can be confirmed by a simple test of observing people doing these tasks. When creating a section in TeklaStructures, although the same rule basically holds, users are forced to do opposite hand movements to get the right result. This is shown in Fig. 5-14. To make it worse, should the user pick "a wrong point", there is no chance to deselect this point and select another point. In this case, the user has to cancel the command and invoke it again.

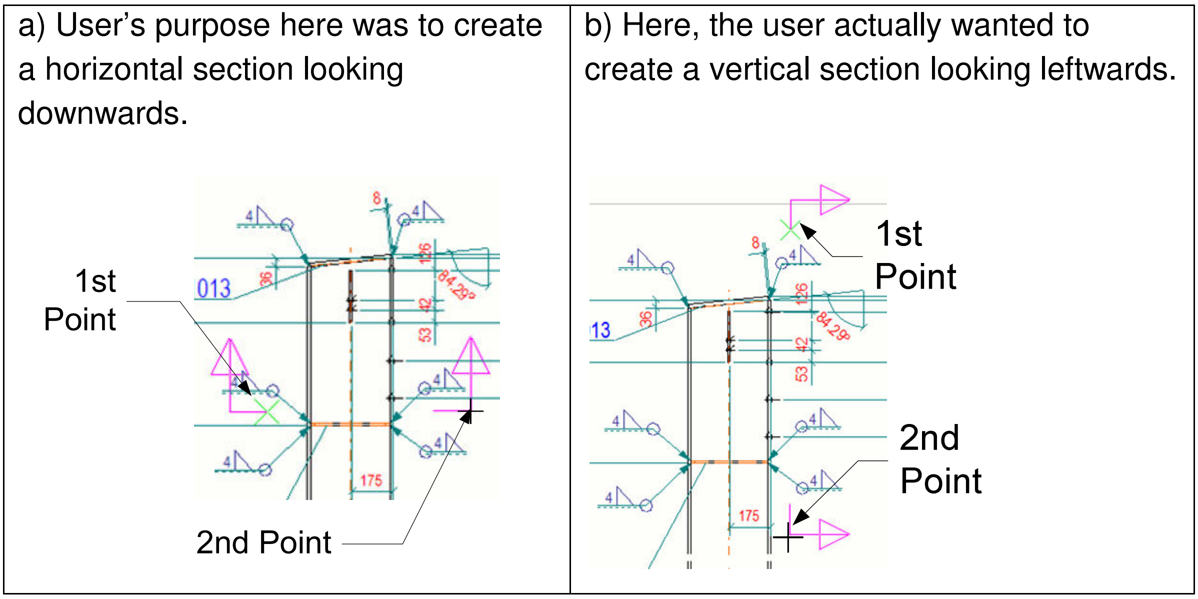


Fig. 5-14: Unnatural hand movements in creating a section, TeklaStructures

To overcome these shortcomings, users' natural hand movements must indeed be considered. When user intends to create a section, at first he/she drags a line with a slight inclination to horizontal or vertical direction. This path is composed of a longer and a shorter distance. The longer distance determines section's extension (vertical-horizontal). The shorter one will determine section's direction (Left-right, top-bottom). Furthermore, the user must be given a graphical clue about the extension and the direction of the section—conforming to the direct manipulation style of interaction. This is shown in Fig. 5-15.

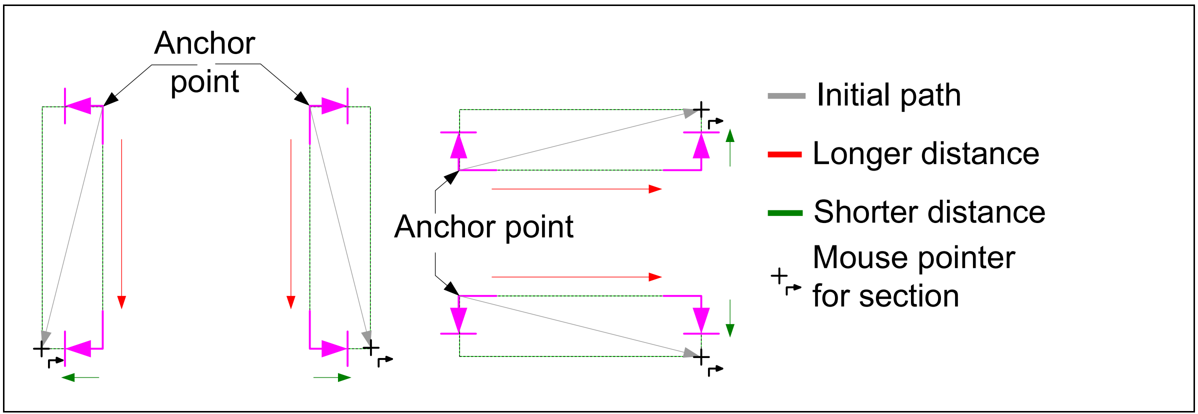


Fig. 5-15: Proposal for "Creating a section" corresponding to initial mouse movement

After this, a rectangle indicates the section borders and extensions, as shown in Fig. 5-16. Moreover, adjustment handles over this rectangle are presented simultaneously i.e. the user would not have to click on the rectangle to see the handles. By dragging and dropping the control handles on the edges of this rectangle, the user can change the view borders of the section, while the depth will be taken automatically from the destination view. In addition, a rotation handle is provided so that the user can freely change the section's extension from left to right or from vertical to horizontal or simply include inclined sections. Therefore, adjustment handles allow recovering from any possible slip which may occur when creating the section and thereby offer full flexibility. When the user has finished adjustments, he/she can create the section view by clicking on the provided text-free on-screen button.

Other features which can potentially enhance usability of this command include:

1. As a very important feature especially for drawings, the CAD program must deal with sections and their designations similar to how Microsoft Word treats different cross references such as figures and tables in a word document. This means that the program must take a record of all sections created in a view or a drawing. Therefore, sections' symbols and their designations must not be treated as pure graphic, but rather tools that embrace more information and are linked to each other. The major advantage to this method is in editing the automatically generated drawings interactively. This option eliminates any risk that a section name does not comply with its corresponding designation in the view—a situation which may occur due to human mistake. Moreover, it allows the user to easily add

or remove a section without being concerned about keeping track of all sections' names and changing them accordingly. Therefore, deleting, adding or any type of modification on sections' names in a drawing must automatically refresh all section names to avoid missing or duplicated designations.

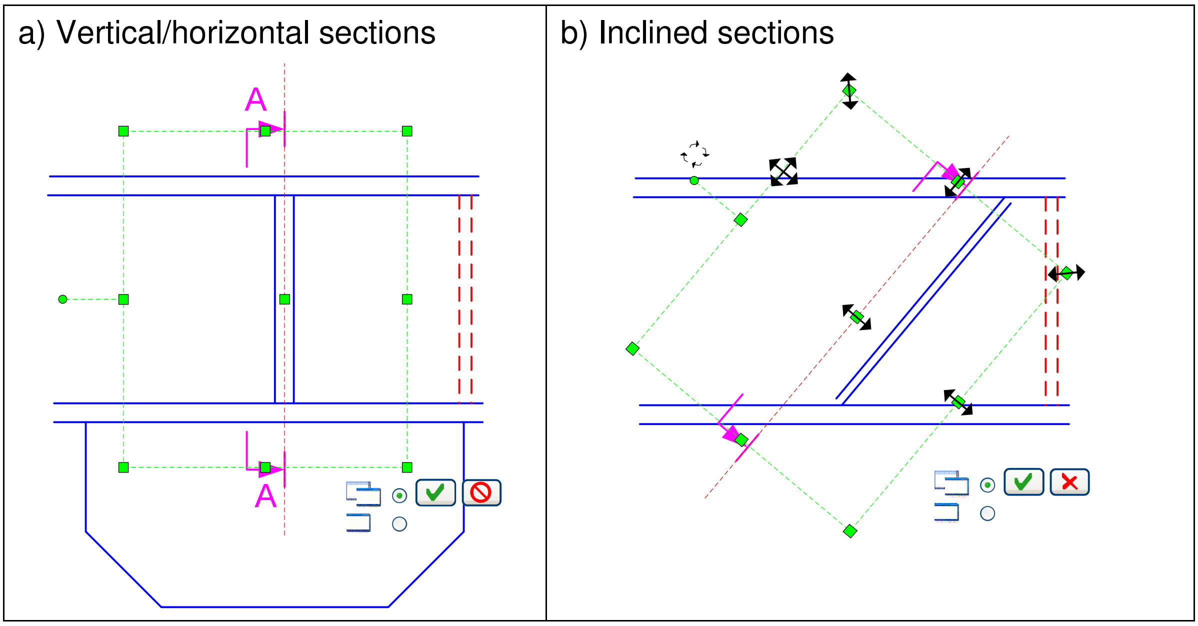


Fig. 5-16: Adjustment handles for section

2. When in modelling mode, the program must take a default name/scale for a section. On the other hand, when in drawing mode, the program must automatically take the next free section name in the drawing. The scale must by default automatically conform to other sections/details that pertain to the member of interest.

3. Hovering the mouse pointer on the section's symbol while at the same time pressing the Ctrl button, must first change the mouse pointer to . Double clicking in this situation must refer to the linked view. If it is in the same view, the program must zoom to it or alternatively open its view. If it is in another window, program must bring its window forward. The reverse must also work; i.e. by a Ctrl-click on a view or its designation, its pertaining section symbol must be displayed.

4. Creating new sections by duplicating/copying section symbols saves the effort of invoking the command each time through menus or by clicking on icons.

5. When giving section extensions, the program must automatically snap to orthogonal directions.

6. Further functions in this command, such as creating sections with certain settings, can be accommodated in a second layer which becomes available through the mouse right-click menu as shown in Fig. 5-17. However, visible

components of a section such as the section symbol or section designation, must allow changes simply by double clicking on them. To guarantee a consistent format of sections in a drawing, such changes must automatically take effect on all instances of sections in the drawing.

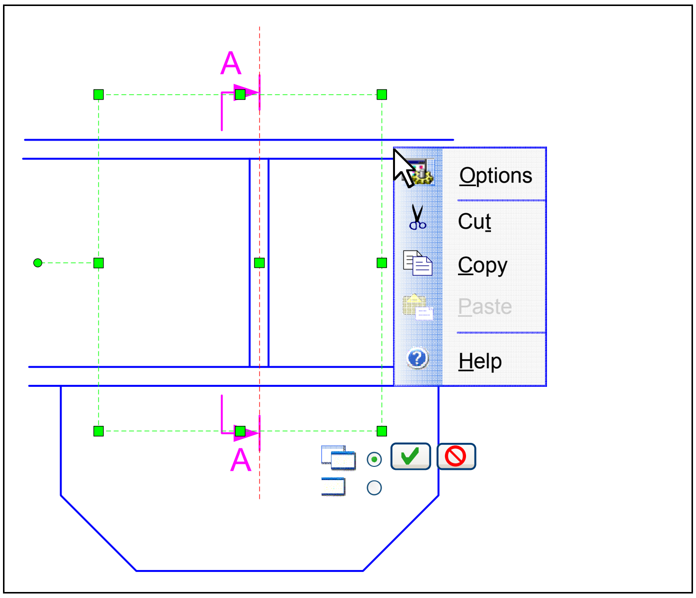


Fig. 5-17: Mouse right-click menu on a section

5.3 Search Function

Search function is a very important element that users often need when working with a high-performance CAD system. Both TeklaStructures and Bocad-3D embed this function albeit with different approaches. Bocad-3D has a comprehensive search function for all elements including welds and bolts. TeklaStructures on the other hand, uses a filter concept which covers building parts as well as welds/bolts. In Bocad-3D, the search results are delivered in a new window, whereas in TeklaStructures they are filtered in the same window or are filtered during the selection. Fig. 5-18 shows this function in both programs; in which language dependency of this function is evident.

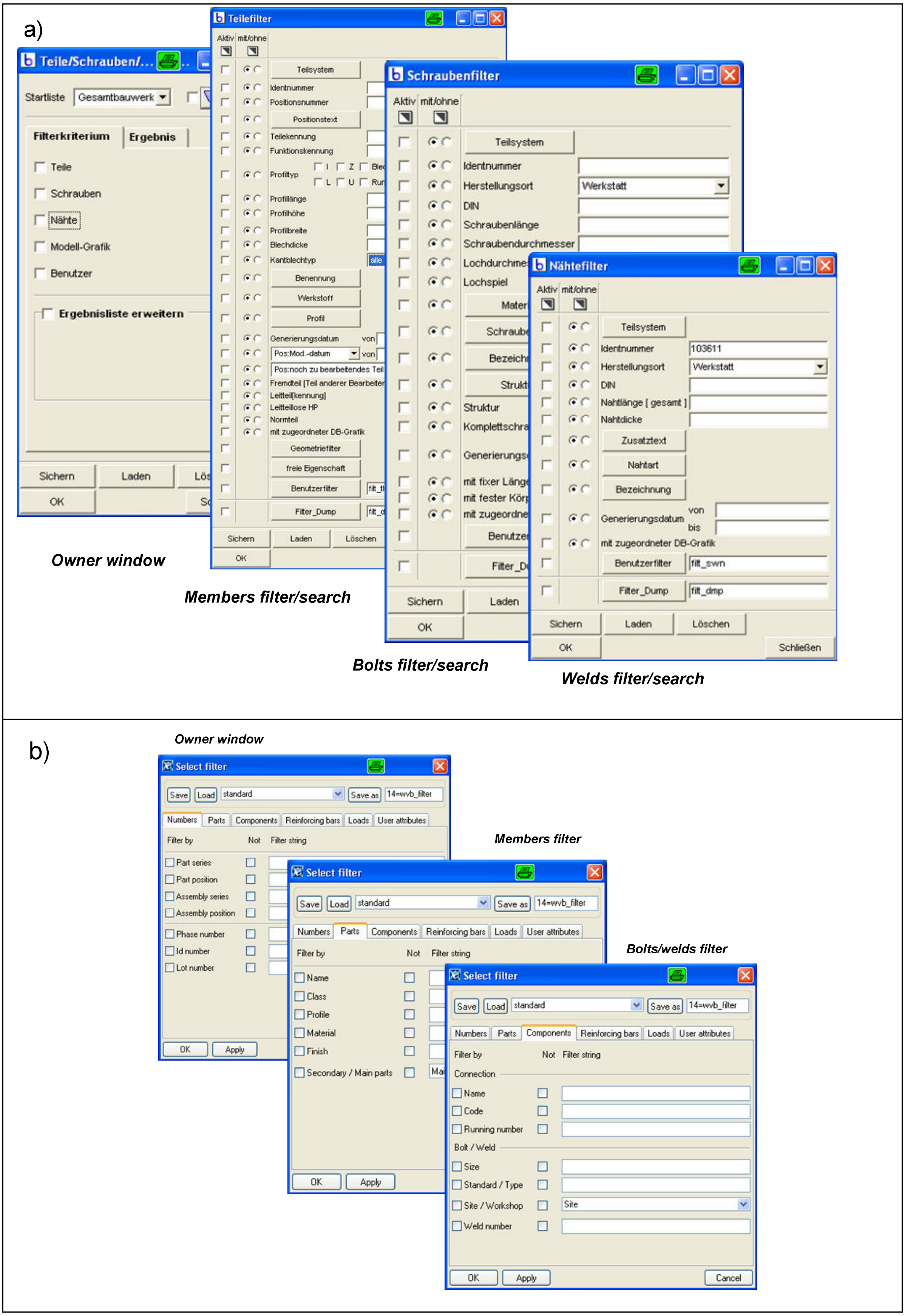


Fig. 5-18: Search/filter function for members, bolts and welds, a) Bocad-3D, b) TeklaStructures

Proposed solution

Besides general considerations regarding the basic controls in this function, following remarks should be taken into account:

- In/excluding the search criterion is done more efficiently in TeklaStructures. Only two options are offered to the user to choose from, as shown in Fig. 5-19.

Bocad-3D TeklaStructures

Fig. 5-19: In/excluding the search criterion in Bocad-3D and TeklaStructures

- This function is sometimes used to find elements that have the same property as the object at hand. Therefore, it is very useful to provide the user with an “Obtain/get” function. This saves the mental effort to store the desired information in “Working Memory” to use as the entry in the search dialog-box. Fig. 5-20 to Fig. 5-23 present the proposed text-free solution for this function. Here, a new button as in TeklaStructures, is introduced which allows quick check/uncheck of the check boxes presented in the command dialog box as follows: 

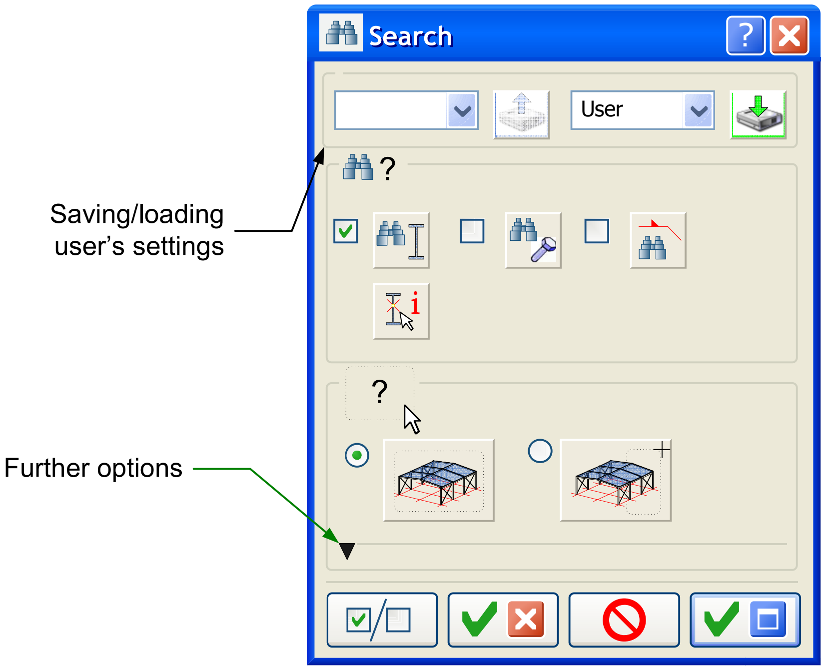


Fig. 5-20: Proposed dialog box for search function

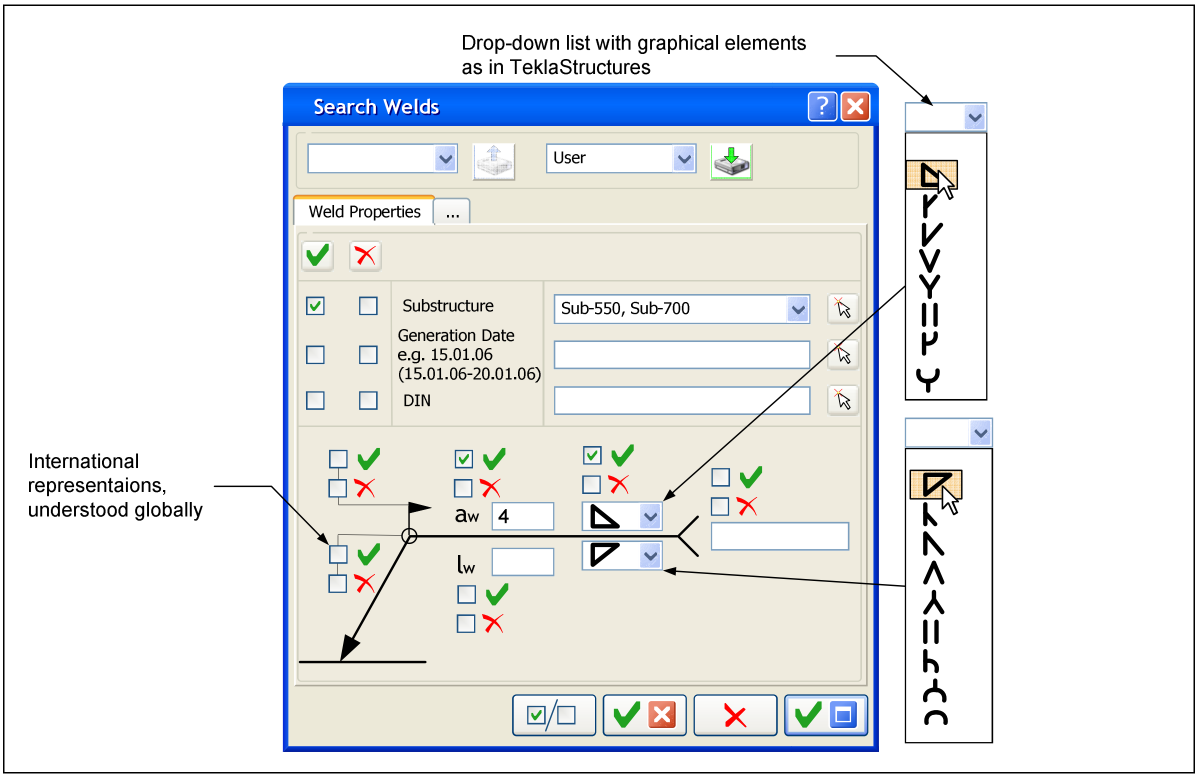


Fig. 5-21: The proposed search/filter dialog box for welds

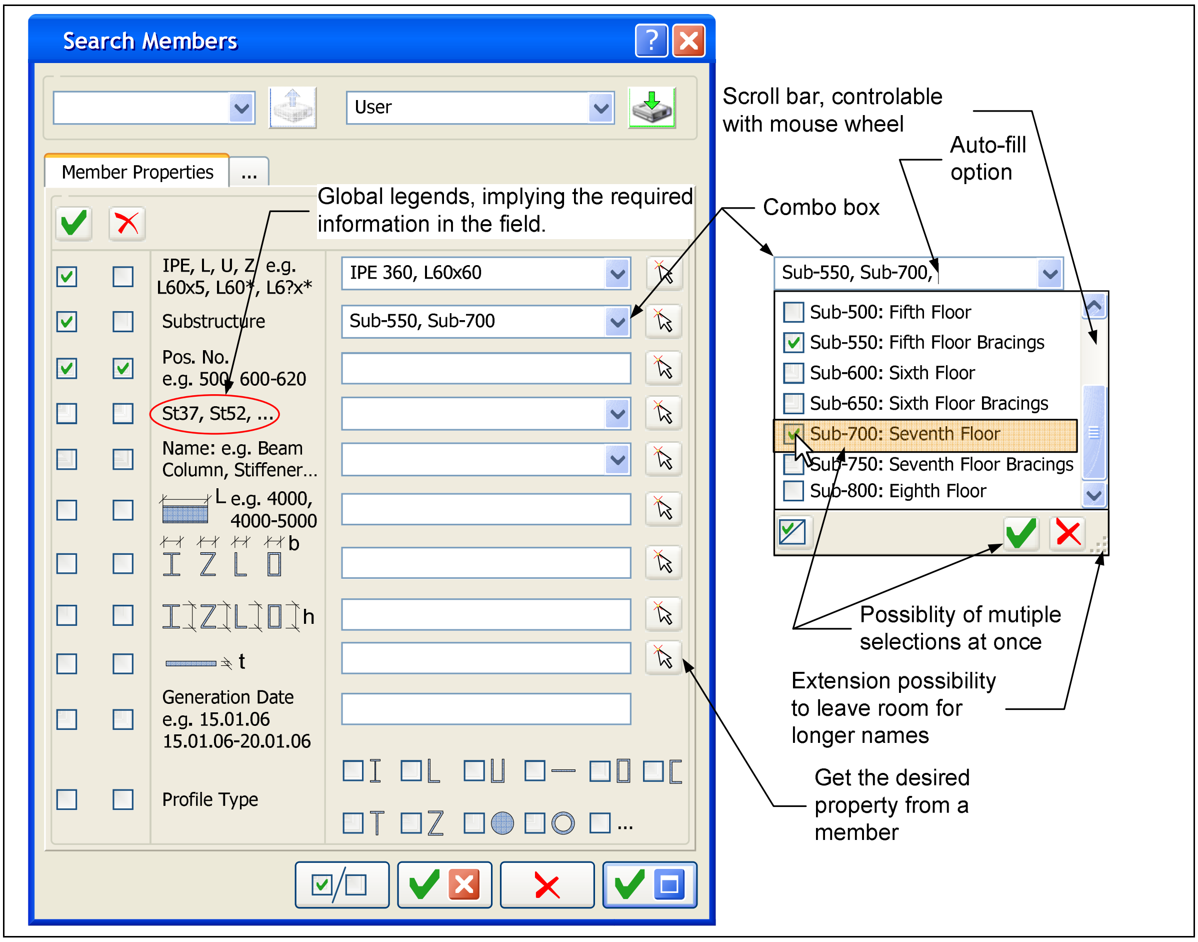


Fig. 5-22: The proposed search/filter dialog box for members

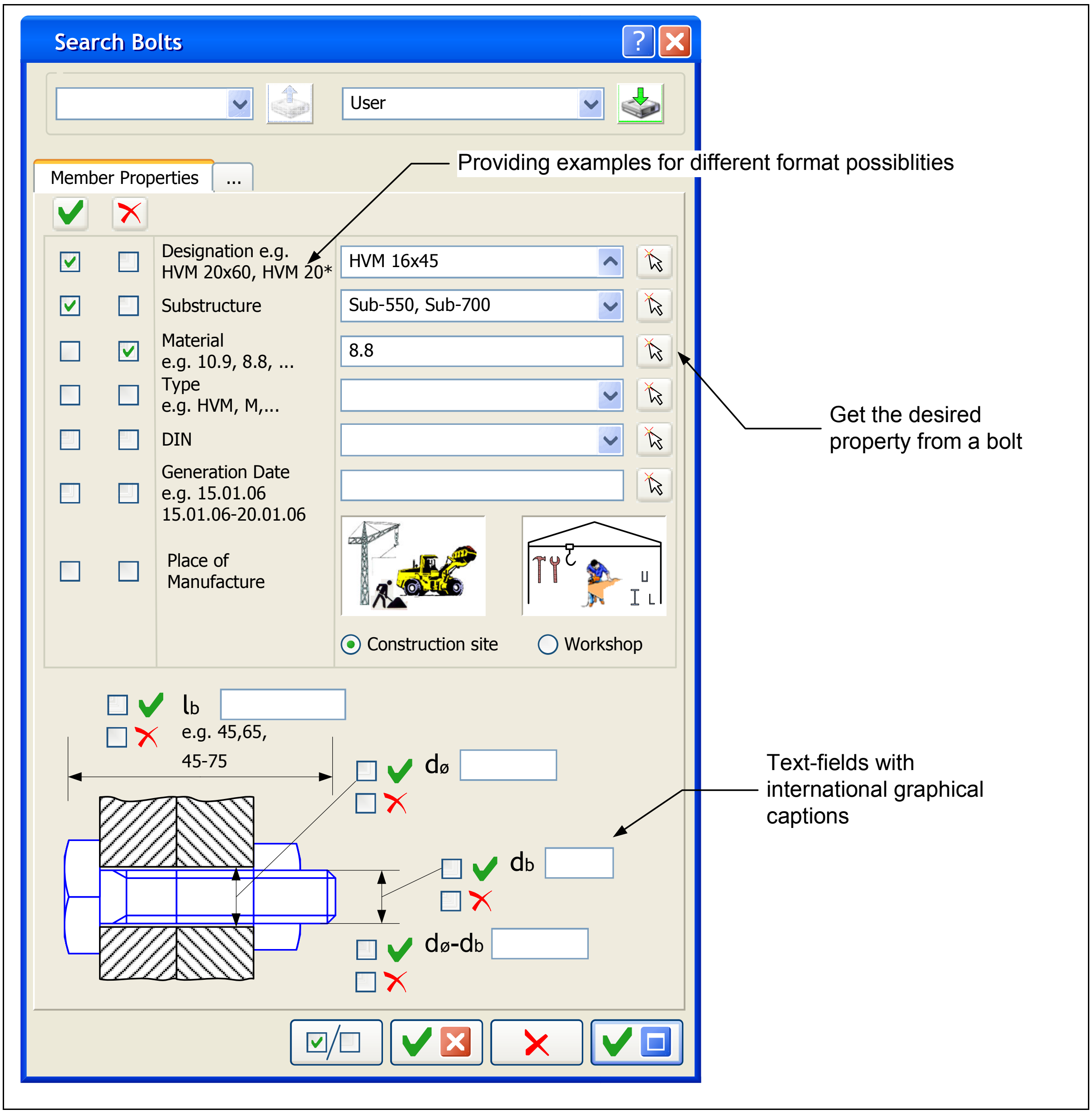


Fig. 5-23: The proposed search/filter dialog box for bolts

5.4 Point-snap

In a CAD program point-snaps are used to indicate or to set a specific geometric point on an entity or an object. They allow precise modelling and are useful within many commands.

There are basically three different ways to deal with points in CAD programs:

1. Defining point-snaps: According to the user's current snap-settings, the program snaps automatically to the points on an entity. Moreover, depending on the point type, the program provides visual cues as the user hovers around the points. This approach is used in AutoCAD, as shown in Fig. 5-24.

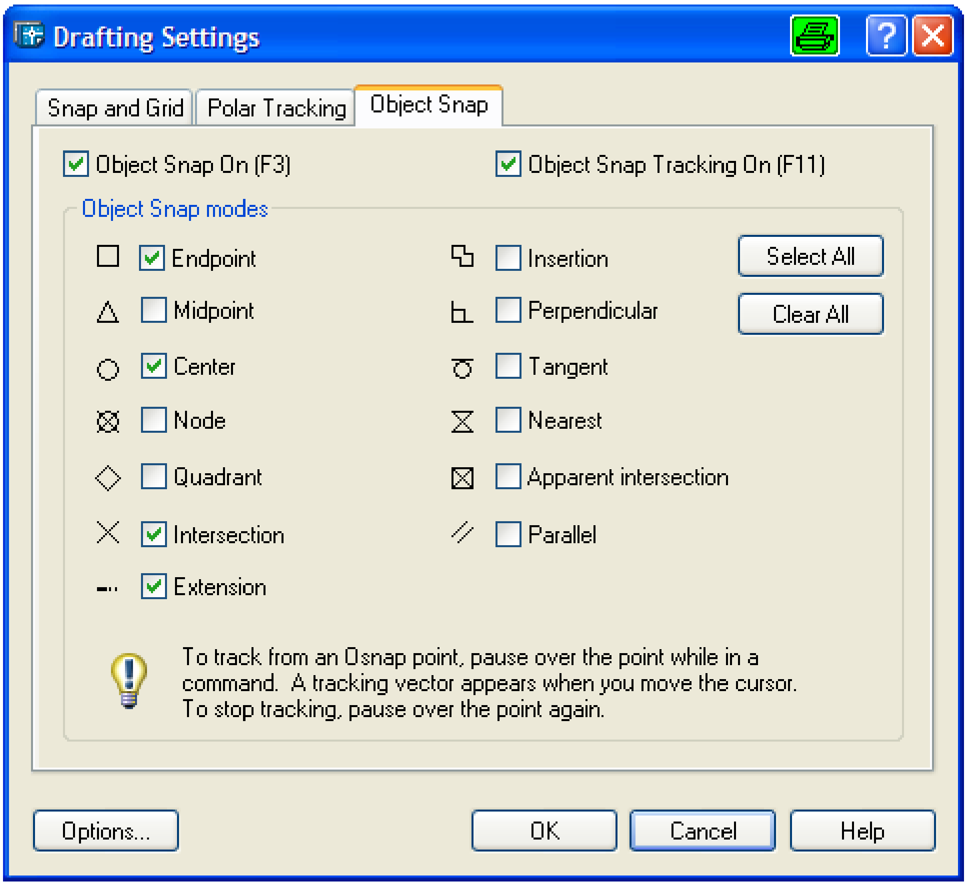


Fig. 5-24: AutoCAD snap-setting

The main advantage of this method is that it is direct and mostly fast. However, there are cases, where this method does not function very effectively; specifying intersection of lines, whose extensions intersect, or specifying an arbitrary point with certain coordinates relative to another point are typical examples. Although AutoCAD provides solutions to overcome these shortcomings such as apparent intersection or tracking, working with them is not yet very efficient. This is shown in Fig. 5-25 where the user intends to draw a new line to the extended intersection of two existing lines. To get the intersection point, the user must hover for some time over the desired lines or the points (or even has to repeat this process several times) until the program recognises the desired action. While this shortcoming may not be very noticeable when working in 2D, in 3D it makes the modelling rather tedious.

2. Providing a point-package: By using a point-package, the user creates the desired points first, i.e. prior to selection, and selects those later when required during the command using the object snap for points. Bocad-3D provides an extensive point-package, see Fig. 5-26.

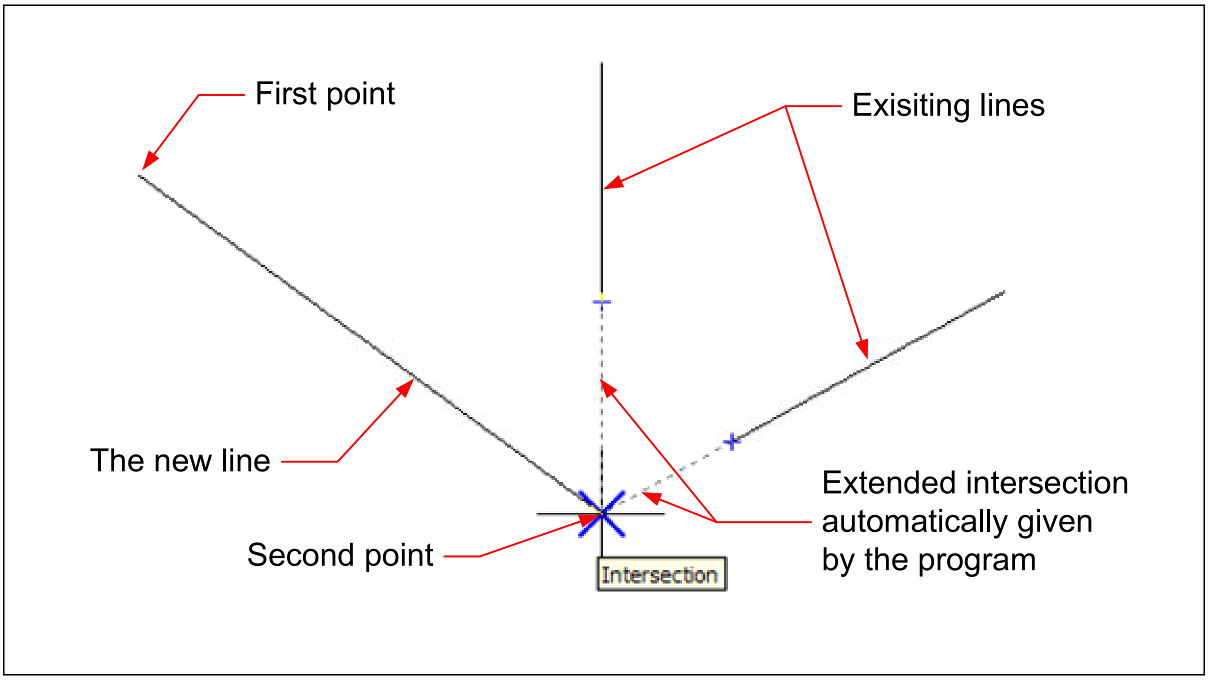


Fig. 5-25: Extended intersection, AutoCAD

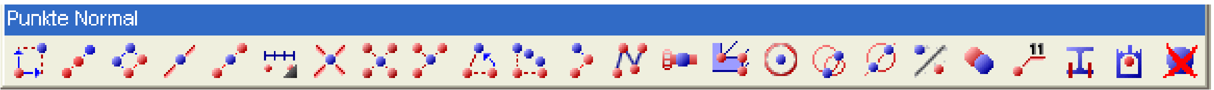


Fig. 5-26: Point-package, Bocad-3D

Although this method offers complete functionality, it has two major disadvantages in comparison to the first method: (1) it lacks an automatic snapping mechanism: user can mostly select only the points that he/she has already created. This is not as efficient for common cases such as end point, intersection, mid point or perpendicular; (2) it lacks an adequate feedback: the user must always select first to be able to see if it was the desired point. This increases the chance of a slip in selecting the right point, which can even become a costly failure. An example shows both of these drawbacks: if the user wants to draw a dimension line, from centre point of an I-profile with rather a thin web, firstly, there is no possibility to select the mid point automatically (as in AutoCAD). Secondly, the user may pick also another point on the web since there is no visual cue to indicate if he/she is about to select the mid point or rather a point on the web, see Fig. 5-27. To overcome this problem in practice, users either have to set the object snap only on points (hoping there is no other point in the vicinity) or zoom in so many times as to distinguish the desired point and zoom out again to the previous working view. This way or another, the method is not efficient.

3. A combination of both methods: An effective solution, as offered by TeklaStructures, is a combination of both methods of having an automatic point-snap together with a point-package, as shown in Fig. 5-28. This method basically presents the maximum efficiency.

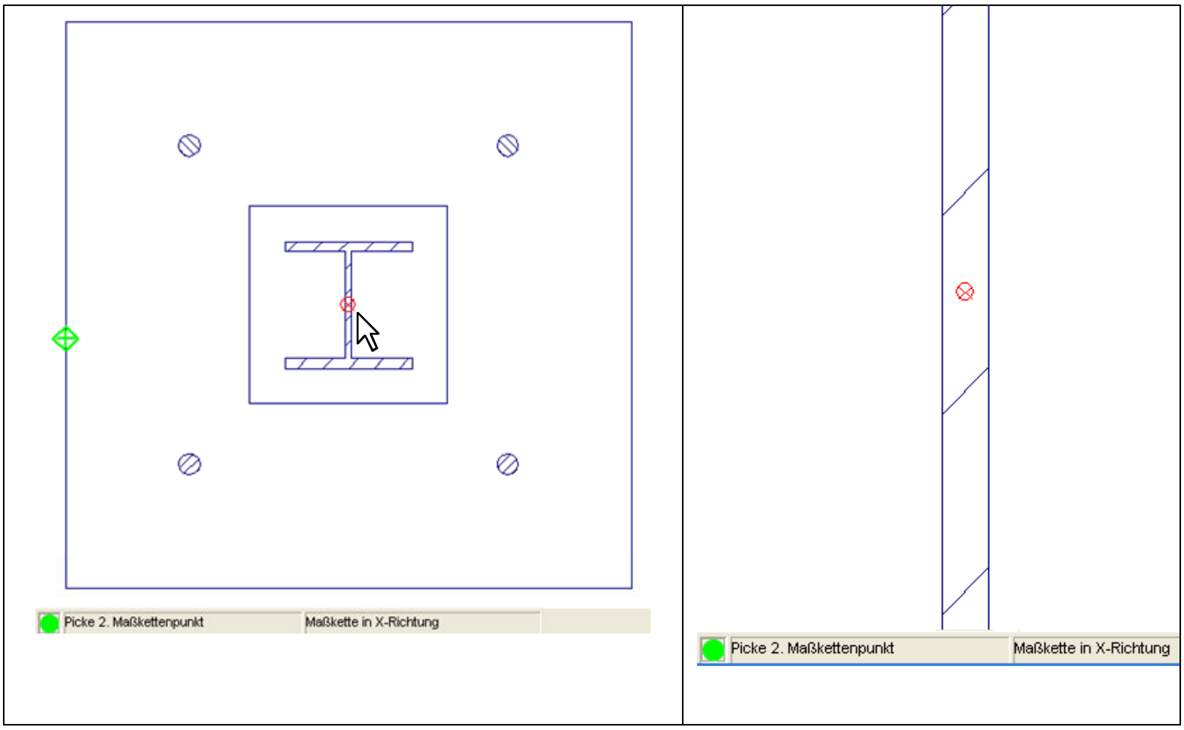


Fig. 5-27: Missing graphical cues for point-snap, Bocad-3D

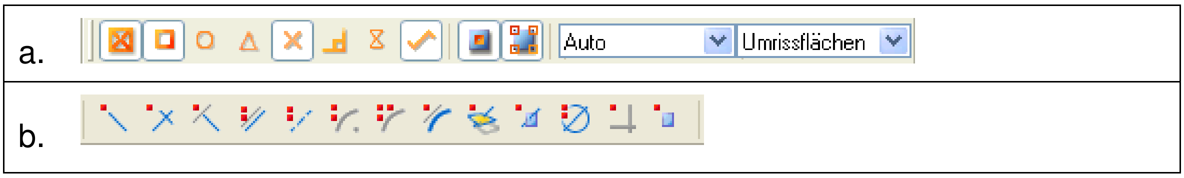


Fig. 5-28: Combination of points' snaps (a), together with a point-package (b) in TeklaStructures

Points include almost 20% of commands, according to the survey results of this thesis. Hence, a due attention should be given to the design of this category of commands. Regarding the point-package in TeklaStructures and Bocad-3D, the following aspects must be considered to achieve a higher level of usability within this category of commands.

Missing Functions

There are three functions which are missing in both TeklaStructures and Bocad-3D and even though existing functions can somehow offer alternatives, they are not easily used. These missing functions are "adding a point along a given direction", "point projection on a plane", and "planes intersection".

The proposed solutions for these functions are given in Fig. 5-29. As an alternative, the first function can also be integrated within the "add point" command, as shown in Fig. 5-30.

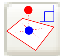
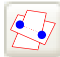
Add point along a given direction	
Point projection on a plane	
Planes intersection	

Fig. 5-29: Proposals for three missing functions in point-package

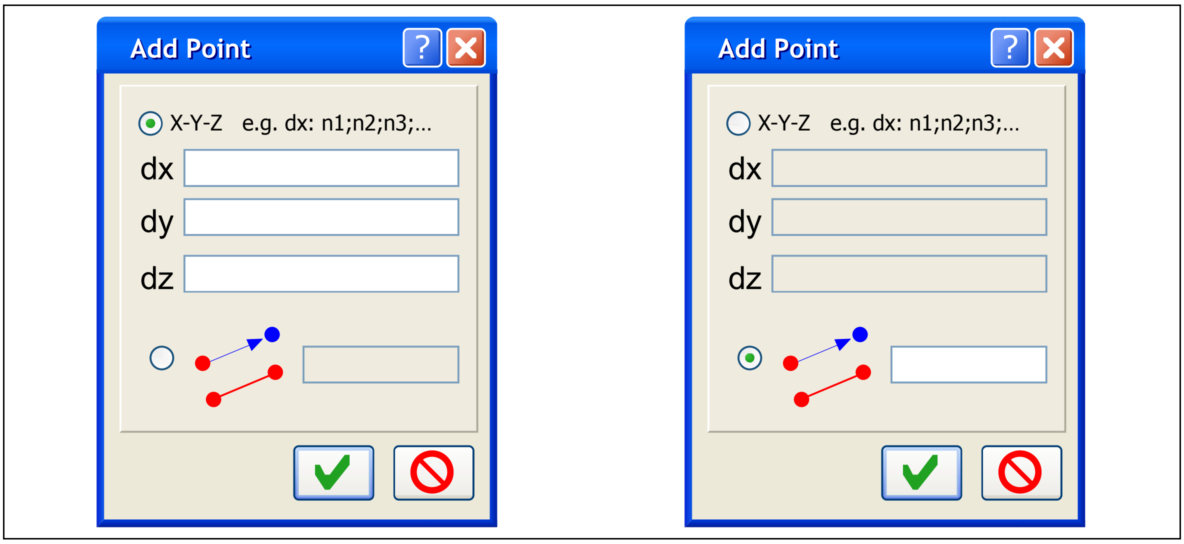


Fig. 5-30: Integrating “Adding a point along a given direction” in “Add point”

Points-Parallel

“Point-Parallel” (or parallel to two points) is in fact a very useful function and is embedded in Bocad-3D as well as TeklaStructures. Nevertheless, both programs offer an awkward procedure for this function: first, the user should give the desired distance, then he/she is required to pick the points bearing in mind that the new points are created based on the right-hand rule having the Z-axis of the view and the X-axis defined based on the order which the points were selected. This is an example of a typical error of “giving users extra problems to solve” in user-interface design. To understand why this procedure is faulty, one should imagine how the users would do the same task, were they drafting on paper: Would they need to decide upon any axis at all? Would they have to decide upon the distance if it should be positive or negative? CAD programs were created to relieve the engineers from the tedious work of drafting; there is no excuse for making simple things complicated for users, even though interacting with a program via mouse and the computer screen can be adequately different from paper and pen.

The same task can be done by giving a positive distance, picking the points in an arbitrary order, and determining the direction according to the mouse movement. A graphical cue, the same as in TeklaStructures (Fig. 5-31), should be used to inform the user about the ongoing process. In 3-D, the graphical cue should snap automatically to one of the coordinate planes/axes at a time.

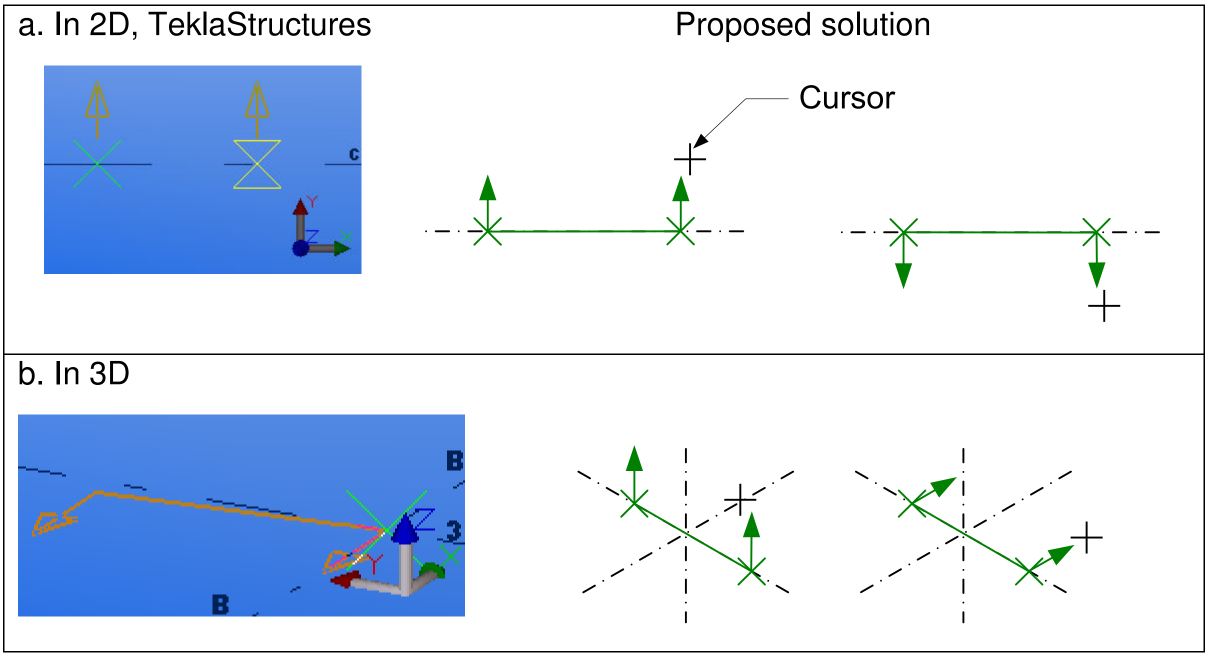


Fig. 5-31: Proposed solution for “Points-Parallel

Members’ Points

Users of high-performance CAD systems usually need to select special points on profiles such as points on middle axis, corners, points on the radius, etc. The context-menu provided for this purpose in Bocad-3D is text-dependent, see Fig. 5-32a. Instead, the solution shown in Fig. 5-33b is proposed.

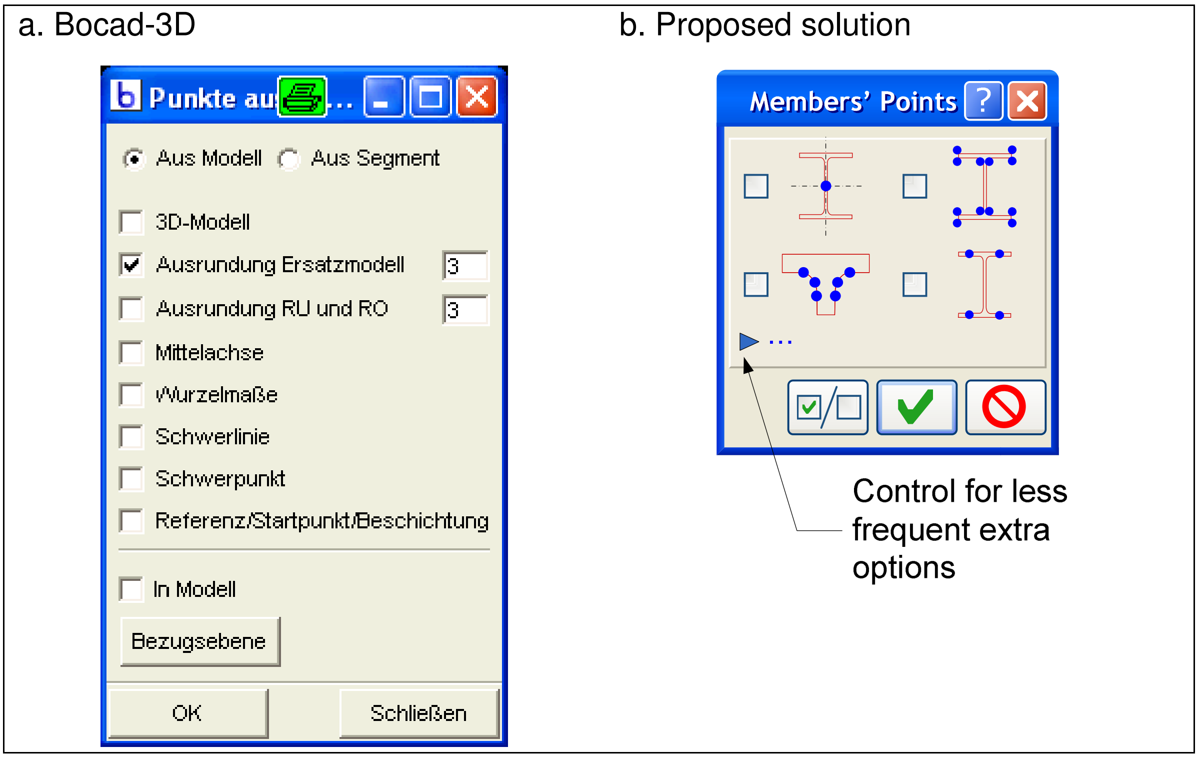


Fig. 5-32: Proposed solution for “Members’ Points” compared with Bocad-3D

Points' Toolbar

Regarding the point toolbars in TeklaStructures and Bocad-3D, the following guidelines are recommended:

- Based on the survey results, the most frequent commands in a point-package are "Intersection", "Mid points", and "Member points". As a design rule, these commands should appear from left to right in horizontal icon-bars, and top to bottom in vertical icon bars. Although a re-arrangement of icons may be cumbersome for users of the previous versions to adapt, for users in new markets such as those in developing countries, it is worth the practice. However, even experienced users may become accustomed to it without much difficulty if new changes comply more effectively with their needs.

- Adding toolbar dropdowns: Toolbar dropdowns are used basically to save up space. Moreover, they can offer an alternative for commands which otherwise appear in context menus. Adding a toolbar dropdown in this case will make the commands more accessible to the users; at the same time it maintains the principle of progressive disclosure. This is a common solution used in AutoCAD and also in Bocad-3D. In design of these controls it should be noted that the active icon appearing on the toolbar is always the one that was last used. This approach is also suggested for a frequent command such as "Member's points"—including its sub-commands. Moreover, it could be used for grouping the less frequent commands such as those pertaining to arc or circle.

- Toolbar separators: Toolbar separators offer a better visibility for the commands which belong to the same category such as intersection or members. Moreover, this boosts the process of forming a muscle memory for users. Using a muscle memory, users will be able to select the desired item on the toolbar without accurately looking at them. Normally, experienced users use muscle memory when interacting with the software. For novice users it takes time and a lot of training/repetition to reach this level. Facilitating such visibility shrinks this process.

Considering all abovementioned points, a solution is proposed for a point-package in Fig. 5-33a, based on the point-package in Bocad-3D albeit with some changes, which are mentioned in Fig. 5-33b.

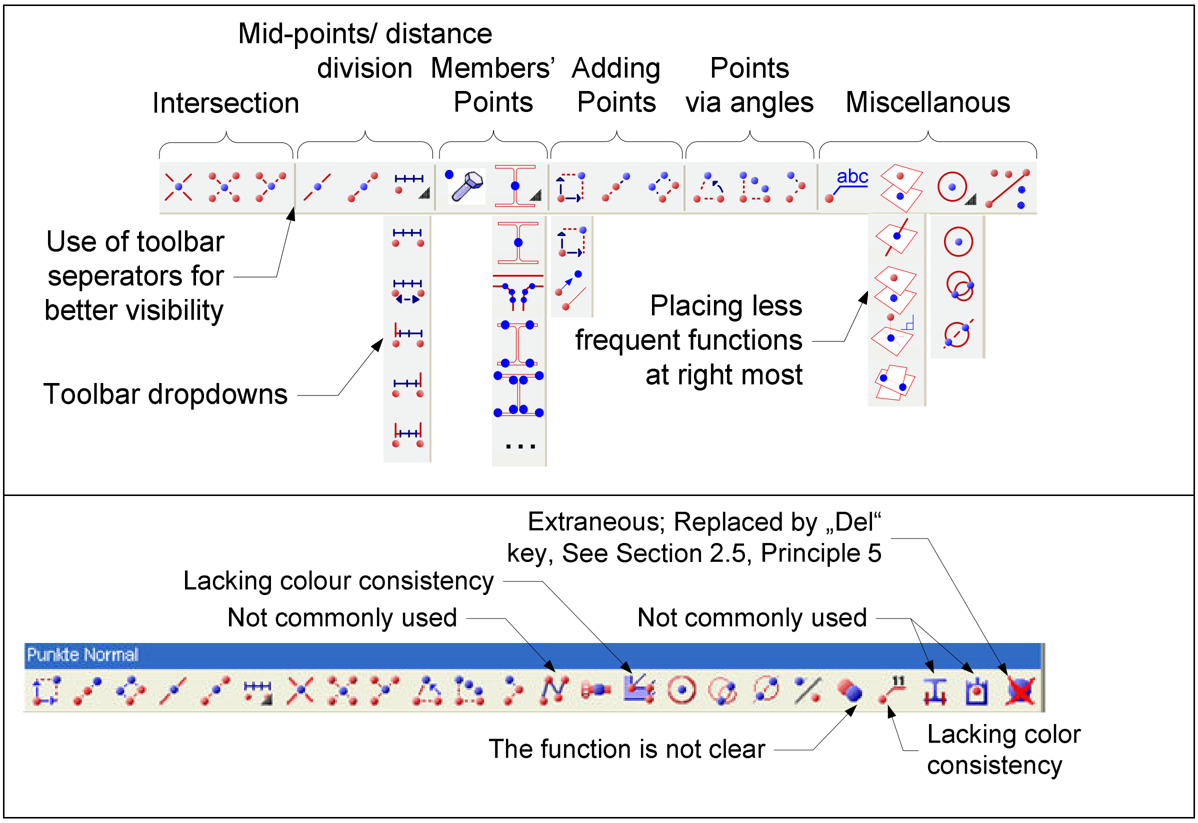


Fig. 5-33: a. The proposed Point-Package; b. Changes made to Bocad-3D

Points in-between

A good user-interface does not demand a fixed mental flow-chart to perform common tasks. Violating this rule lowers the usability of the software by overburdening the user's mental capacity. Based on this rule, the user must be able to use the point-package also during an ongoing command without cancelling it. In TeklaStructures, this is not possible; i.e., invoking a point command cancels the ongoing command. In Bocad-3D such function is provided by defining an extra point-package with a different colour, as shown in Fig. 5-34. This solution, however, takes up screen space and is almost confusing at first encounters. A better solution is to use the same point-package also during a command which does not cancel the command. This means that the program itself should recognise if the point command is being invoked during a command or not and react correspondingly.

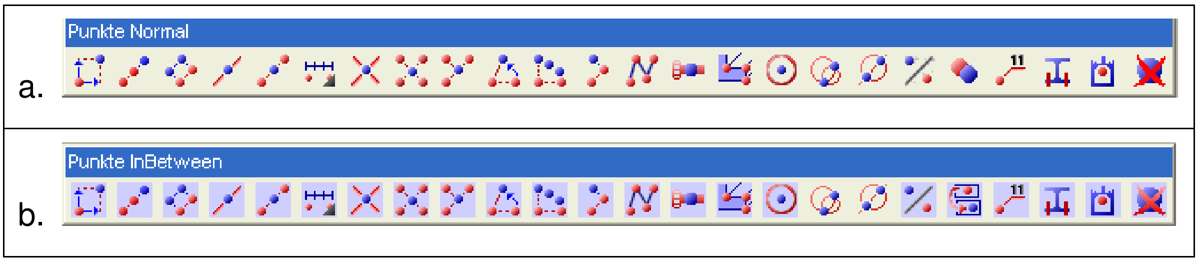


Fig. 5-34: a. Normal point-package; b. Point-package “In-between”, Bocad-3D

5.5 Contour-edit

"Contour-edit" or "plate edit" is an important command in a CAD program for steel detailing. The results of the survey in this research showed that it is the most frequent command in the category of construction/modelling commands with about 15% of frequency (according to Table 3.7). Therefore, it is reasonable to provide a direct and text-free solution for this command.

The methods provided in both Bocad-3D and TeklaStructures are not very intuitive. In TeklaStructures, some basic tasks regarding this command such as moving the corners, take the advantage of normal copy/move function in the program and therefore save the time for learning new commands/function. Bocad-3D offers an extra environment for this command, whereas TeklaStructures allows the user to make contour modification right in the model. The latter solution seems to be more direct as it does not introduce a new layer for opening up an extra environment. However, the way it is implemented in TeklaStructures is sometimes confusing for the user especially when it comes to chamfer shapes as it is shown in Fig. 5-35. Although introduction of a drop-down list is helpful, the fields provided for entries in x and y are rather ambiguous since it is not clear to which directions these are referring. Moreover, it is sometimes hard to select the contour points—especially in complex models or when there are several different contour elements next to each other. In such cases, the user will have to hover on the corners for a longer time until the target points are recognised and highlighted by the program. Therefore, it is in general advisable to have also an extra command environment where not only all corresponding tasks within this command are presented but also the provided view is limited to the target element. The extra environment may offer even more flexibility as it makes it possible to provide more graphical widgets, which otherwise swallow up a lot of memory capacity, should they be implemented in the model view.

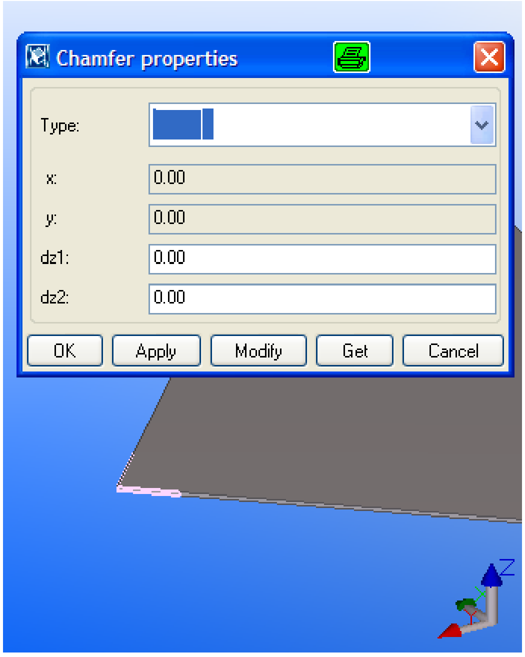


Fig. 5-35: Editing chamfers, TeklaStructures

The most frequent tasks within this command are now discussed.

For corners:

- Moving either to another point manually or via dimension;
- Adding new contour points between two existing ones;
- Deleting corners;
- Defining corner chamfers.

For edges:

- Moving parallel either to a point or via dimension maintaining the original length while changing the original angles;
- Alternatively, moving to a point or via dimension maintaining the original angles but changing the original length.

For these most frequent tasks, it is the best practice to allow the user to interact through direct manipulation by using appropriate adjustment handles. To achieve this, the following guidelines are proposed:

- Between every two existing corners, a third point must be defined by the program and automatically shown on the edge. Does the user hover the mouse cursor over this point, the cursor will automatically change to ( ), indicating intuitively "move to the desired location", Fig. 5-36. This automatically eliminates the need for an extra command of adding contour points.

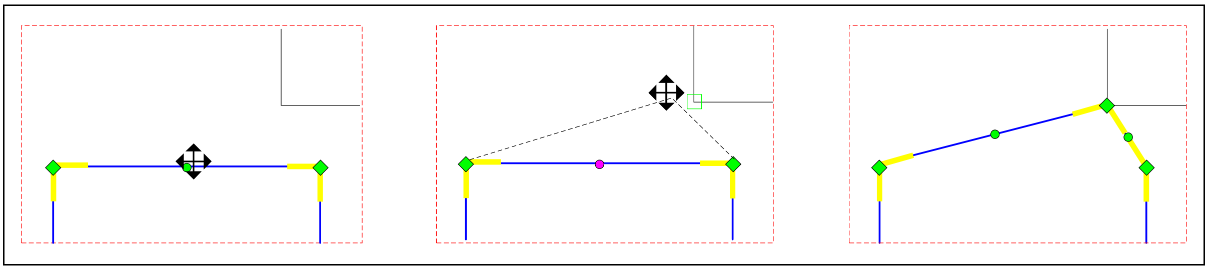


Fig. 5-36: An intuitive solution for defining a new contour point

- For moving corners as well as edges, a combination of common handles and point-snaps is applied, as shown in Fig. 5-37. For moving edges, the default form is in such a way that angles will be maintained; the other alternative can be invoked through the menus. This way or another, the direct manipulation nature of the action is maintained. Moreover, the use of numerical entry via keyboard should also be possible.

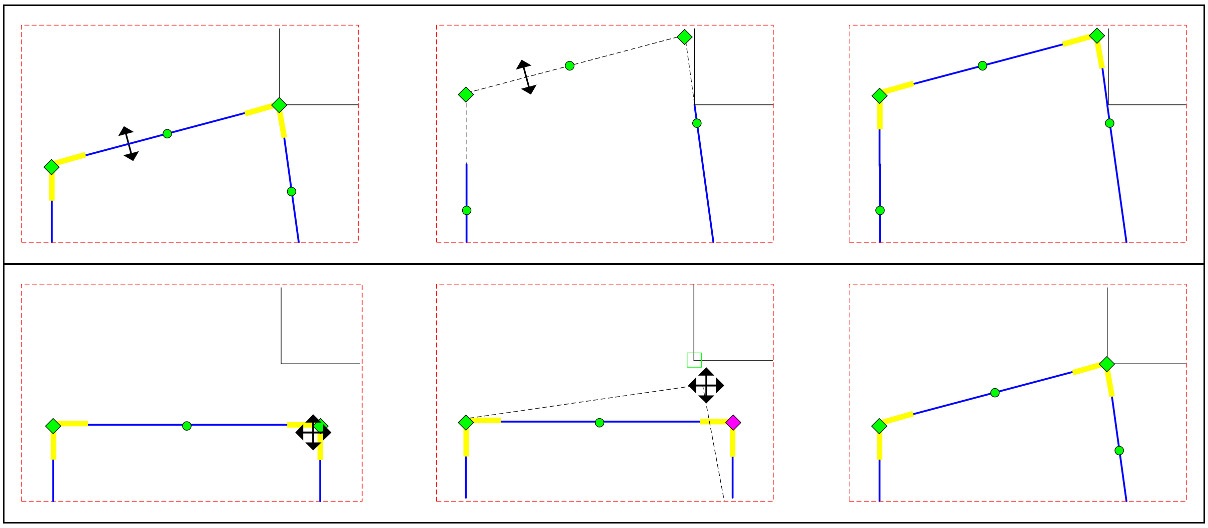


Fig. 5-37: Direct manipulation of contour edges and contour points

- Within the command environment, contour points (or even better, their specific shapes as in TeklaStructures) should be shown as highlighted points. By double-clicking on these points, a dialog box appears with a drop-down list providing different chamfer shapes as well as text fields, which facilitate entry of values. This is shown in Fig. 5-38. To avoid any ambiguity concerning the direction of the desired distance, local axes must be shown on the selected corner. Moreover, to determine the direction of the local axes, it is advisable to follow the right hand rule with Z pointing outwards the working plane.

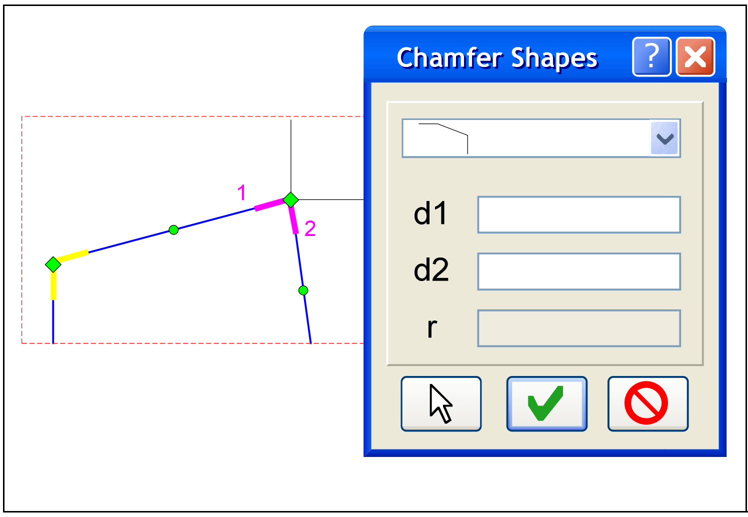


Fig. 5-38: Dialog box for changing contour corners

- The extra environment for this command must have a modeless function, (See Section 4.2).

- In this environment the target should be shown from top view with the default view-depth which is twice as much as the plate thickness.

- The target element/plate must be represented with a different colour than that of the surrounding elements. This not only avoids possible mistaking of this element for similar elements, such as a case of double stiffeners at both sides of the web in an I-beam, but also provides clarity (concerning for example hidden lines) during contour modification.

- The command environment can accommodate an option for immediate collision check of the target element with its surrounding elements. Alternatively, this can be automatically done when the user intends to leave the environment (the user will be notified only if a collision exists).
- Finally, embedding all these text-free functions will not exclude the need for menu items. All remaining (less frequent) functions can be included in menus.

Fig. 5-39 shows a typical solution for the extra environment for contour-edit.

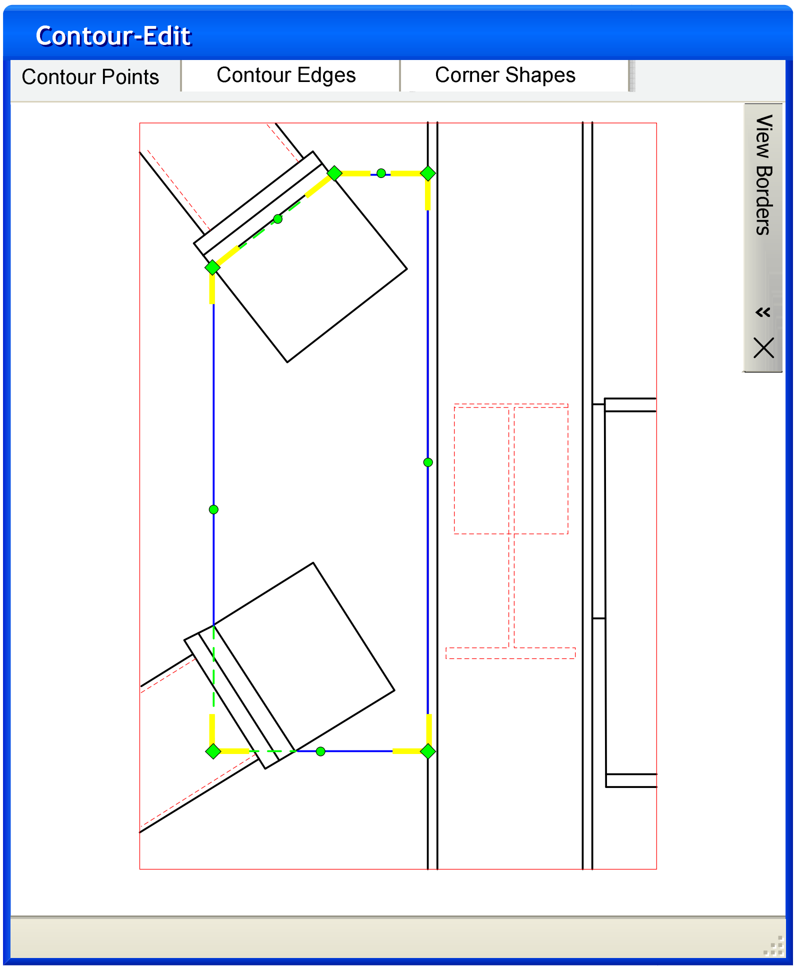


Fig. 5-39: Proposed solution for the command "Contour-Edit"

5.6 Creating Profiles

Creating profiles is one of the most fundamental and at the same time frequent command in a 3D CAD program. The command window for this function is shown in Fig. 5-40 and Fig. 5-41 for Bocad-3D and TeklaStructures, respectively. Apart from general attributes, both programs offer additional settings, so that the user can locate the position of the new member more precisely outright. This is actually a good practice and avoids belated adjustments on the new member such as move, shortening/lengthening. However, the way this idea has been implemented via the user interface in these programs actually does not very much facilitate successful results especially for novice users.

In TeklaStructures (Fig. 5-41), there is no graphical clue whatsoever which can aid users to determine the correct orientation of the element. The available options and their corresponding extensions in drop-down text fields (On plane, Rotation, At depth), apart from their language dependency are merely ambiguous for the novice user who simply intends to create a member between two desired points.

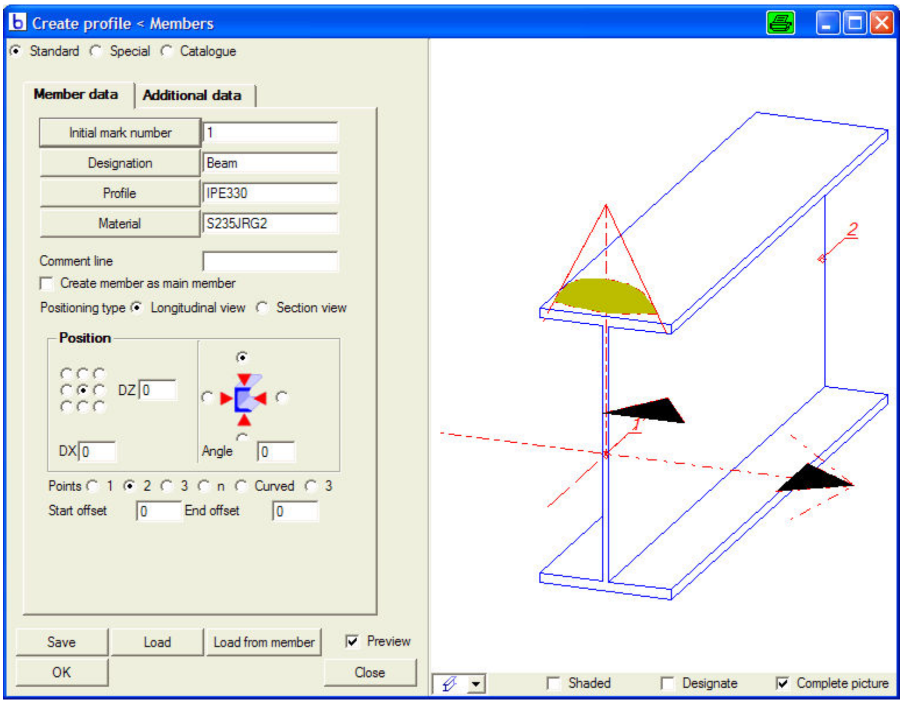


Fig. 5-40: Command window for “Create a Profile”, Bocad-3D

The result is that the user often has to figure out the correct orientation by trial and error—thanks to the effective “modify function” of the program the process will not take long but nevertheless the efficient “modify function” must be an additional feature not a substitute for an adequate preview.

On the contrary, Bocad-3D presents two important improvements: implementation of a real-time preview and a clever use of nine radio buttons, which provide the anchor points. However, it is not yet easy to discover the analogy between the desired points/plane and the graphical interface provided for anchor points and the preview. The local axes of the member, which are shown in the preview, comply with a 2D case where the view and the local Z-axis have the same direction. However, they fail to resemble the right information in 3D where the local axes of the member could be different from those of the view. Furthermore, sign and corresponding effects of the initial offsets are ambiguous. To make it worse, DZ and DX reveal inconsistent behaviour with respect to positive or negative values, as shown in Fig. 5-42.

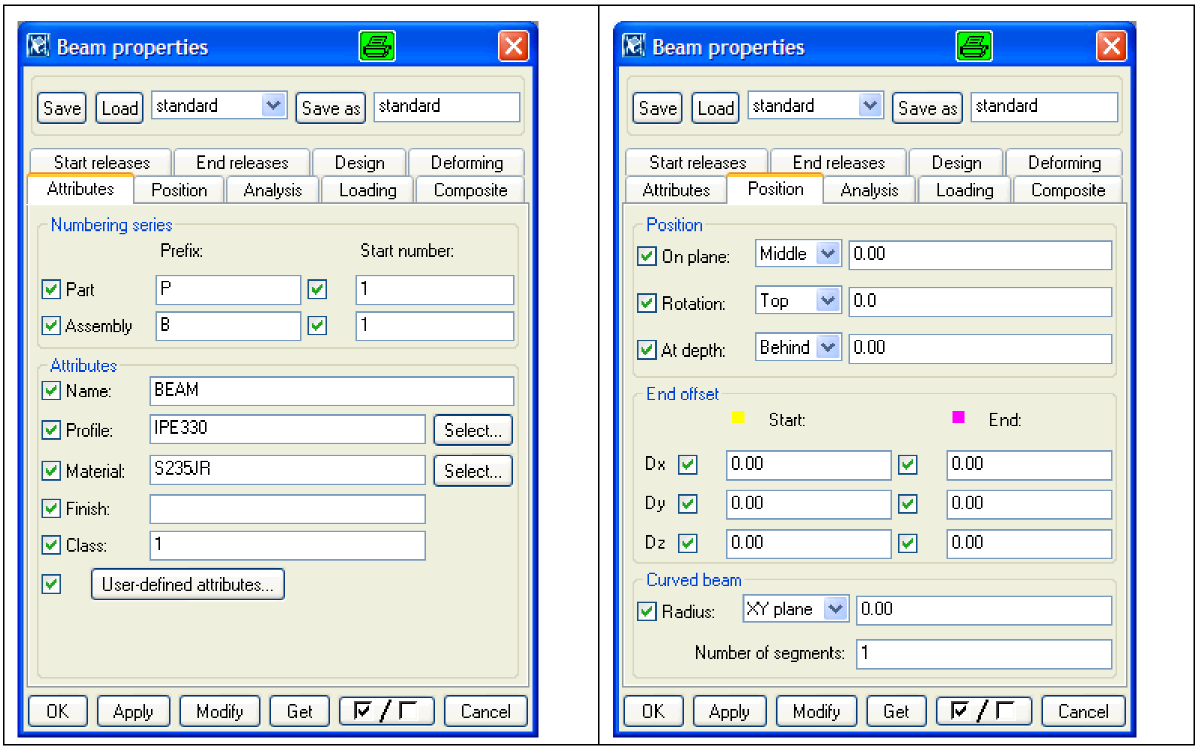


Fig. 5-41: Dialog box for “Create a Profile”, TeklaStructures

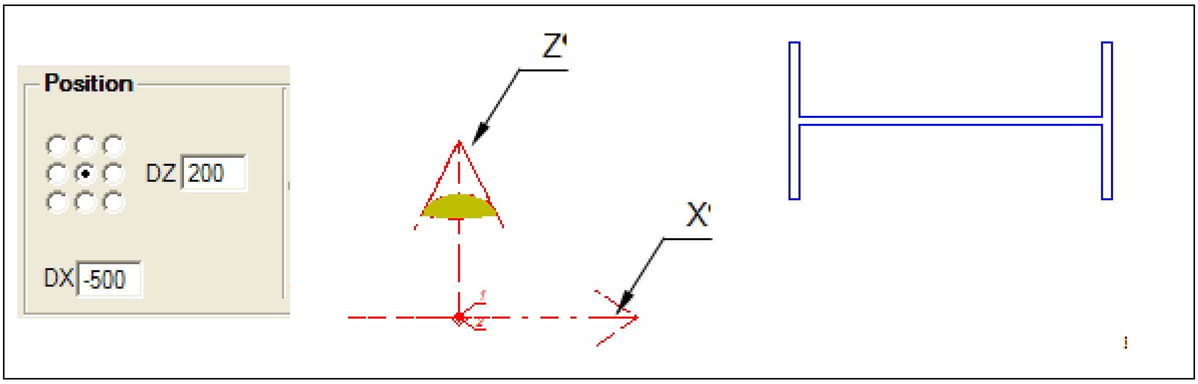


Fig. 5-42: Inconsistent behaviour of distance values for “Creating a Profile”, Bocad-3D

Finally, for most Iranians, using D as in DZ, does not imply a differential value in Z direction for example. This makes it harder for this category of users to guess correctly at first sight; instead, δ should be used.

Schulten (2007) proposes some improvements for this command, as shown in Fig. 5-43 [15]. His proposal presents following advantages:

- Anchor points are obviously clearer;
- There is no ambiguity regarding the initial offsets and their positive/negative direction;
- The command window is completely text-free.

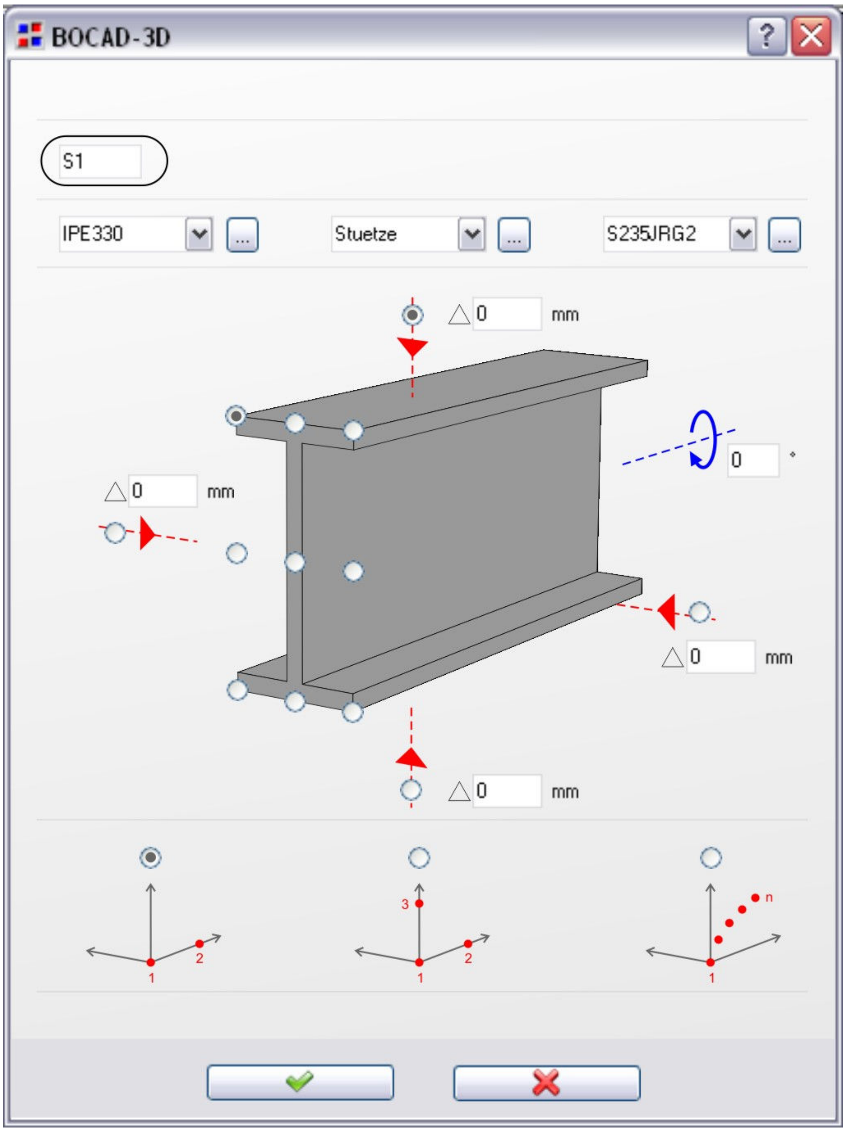


Fig. 5-43: An intuitive dialog box for creating profiles, Schulten [15]

As a result, the user is likely to have a more intuitive interaction with this dialog box than the ones implemented in Bocad-3D or TeklaStructures. The dialog box has however the following drawbacks:

- The profile shown can hardly serve as a real-time preview since the location of anchor points (as radio buttons) have to change correspondingly with respect to the profile's geometry and/or the given offsets. It may be possible to offer different profile shapes (I, L, U, Z) with their corresponding anchor points which will serve as typical previews. Although this solution lowers the burden of UI programming, it cannot offer all the advantages of a real-time preview for the user.

- When working in 3D, the dialog box poses a similar shortcoming as in Bocad-3D; i.e. it is hard to resemble the axes shown in the dialog box with the global axes of the model. In fact, it is not very clear whether the shown axes resemble the (local) axes of working view or local axes of the member.

As an extension of this text-free solution given by Schulten, the dialog box in Fig. 5-44 is proposed.

In comparison to the discussed examples, this solution provides clarity concerning the exact location of the member with respect to the selected points. The shown axes and their designation (1, 2, and 3) imply the local axes of the member more effectively. The method works also for 3D views. The end offsets can be given either via the provided text-field or simply by clicking direct on the dimensions in the real-time preview.

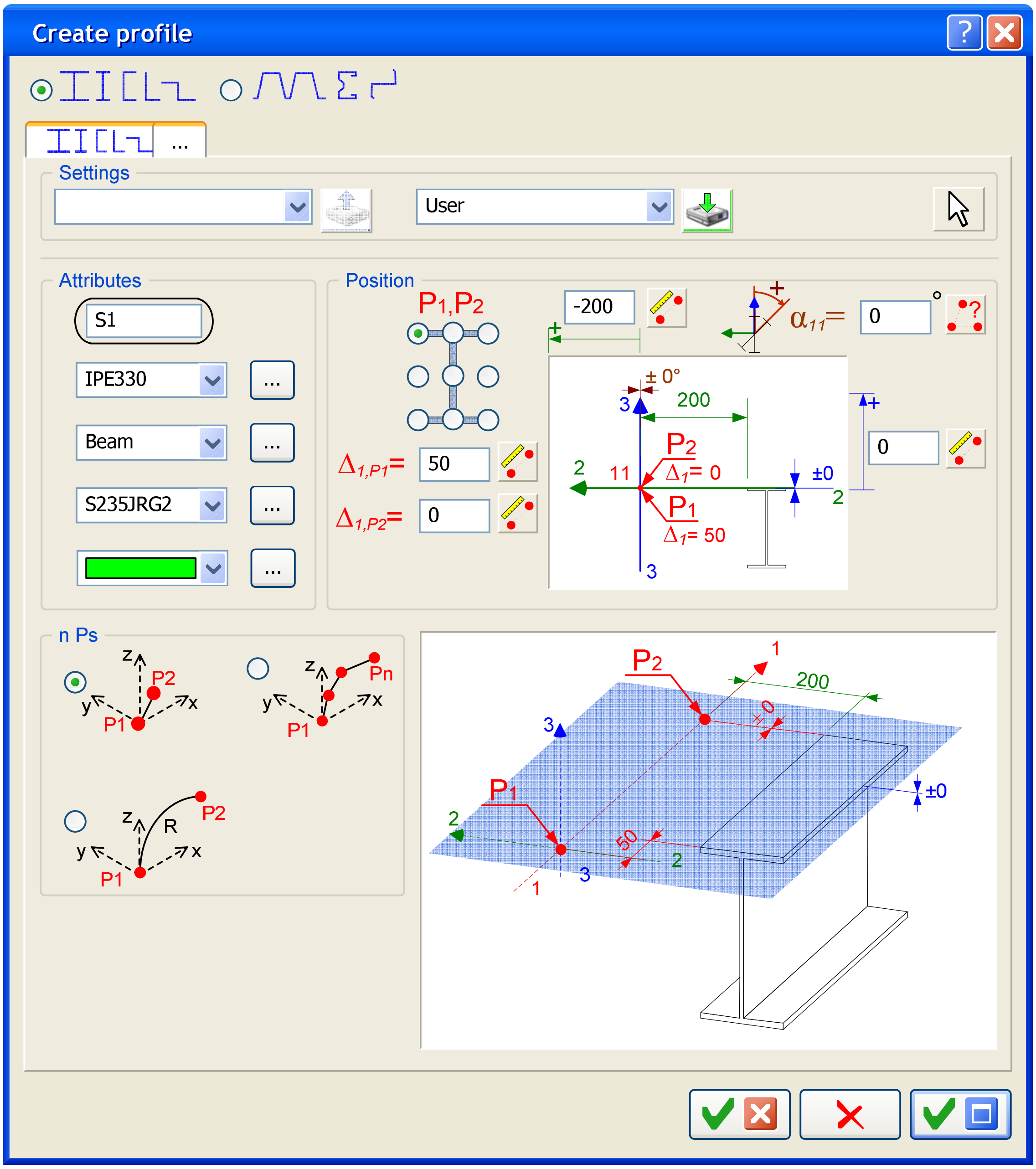


Fig. 5-44: The proposed solution for the command “Create a profile”

When working in 2D, the reference plane implies the working plane. In 3D on the other hand, it is not likely to pose any problem, since it is determined only by the local axes of the profile, not the coordinate axes of the model.

The objects provided in real-time preview facilitate also direct manipulation. This means that the user can insert initial offsets, determine the anchor points or move/shorten the profile by use of common handles.

This dialog box has a drawback that it may look slightly cluttered at first sight and user may have to explore it a little more. However, the accommodation of a real-time preview will ease this starting phase.

5.7 Undo/Redo

Despite its relatively infrequent use (5%), Undo/Redo function has an important functionality within a CAD program. This function offers the basic way for the error recovery and highly contributes to a low-risk environment. To enhance the usability of this function, following rules should be considered:

- User must be able to undo/redo almost all possible actions within the interface be it graphical or modelling. In other words, users must be able to undo/redo all commands, which have an impact either on the 3D model or on the graphical display. Exceptions here are commands pertaining to navigation in the model such as zooming, panning and 3-D rotation. Likewise, commands such as view from a member, section, or internal program settings, which have no direct effect on the model, need no Undo/Redo function.
- In a high-performance CAD system, there are basically two categories of modelling and graphical commands. It is recommended to provide two different undo-functions for each category. This way, user can have full control over this function and would not have to undo many desired graphical commands if he/she wants to undo a modelling command or vice versa. Therefore, independent undo-functions for graphical/modelling commands are recommended.
- The function must provide a feedback indicating which command is to be undone or redone; excluding this option, as in TeklaStructures, will result in user's confusion because he/she may not be able to follow all the corresponding changes in the model, making it hard to be sure that the right steps have been undone. In simpler programs such as Microsoft Word, tracking the changes is usually easy for users, but in a CAD program, user may undo an action, whose result may not be apparent in the open views, leading to undesirable results. This feedback is provided in Bocad-3D in a form of a "further inquiry"—asking the user if the command must be undone/redone. Another way is using the mouse-

tooltip; by hovering the mouse-pointer on the undo/redo function, corresponding command should be shown in the mouse-tooltip. This method is also used in Microsoft Word (Fig. 5-45).

Fig. 5-45: Mouse-tooltip for "Undo", Microsoft Word

- The Undo/Redo function must provide a history of the performed commands. This allows users to undo a series of commands at once instead of having to click on the undo icon for several times. User should be able to determine the number of commands displayed in the history, but as a default, the history may contain the last 25 commands. Furthermore, unlike Bocad-3D, the dialog box for this history must have a modeless function (See Section 4.2). Microsoft offers a very efficient solution for the history, as shown in Fig. 5-46. In the provided dropdown field, the steps will be selected automatically as the user points at them using the cursor. It should be noted that in an advanced CAD program, it will be a good idea to preserve the undo history even after closing the model— as in Bocad-3D. Although this option may be rarely used, if it does not lead to much larger databases, it is worth the practice.

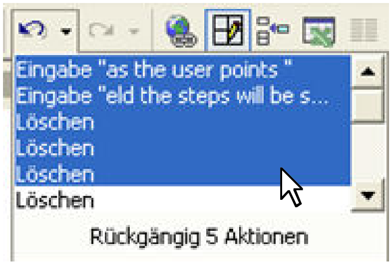


Fig. 5-46: "Undo" history, Microsoft Word

- Assigning an arbitrary step for undo: In all windows applications, users can undo commands exactly in the same order in which they were executed. This means that, if the user wants to undo say the 4th command before last, all four last commands will be automatically undone. In a high-performance CAD system, this can sometimes be very undesirable. Therefore, it will be useful to provide an option, which allows to determine exactly which steps to be undone and which steps should remain unaffected. Obviously, if user's selection leads to a modelling or a logical conflict, the user should be informed with an appropriate feedback by the system, as shown in Fig 5-47.
- The Redo button must have an auto-enabling behaviour; i.e., by default, it must be disabled and appear enabled only after an Undo command.
- Undo in-between: Undo must not cancel any ongoing command.

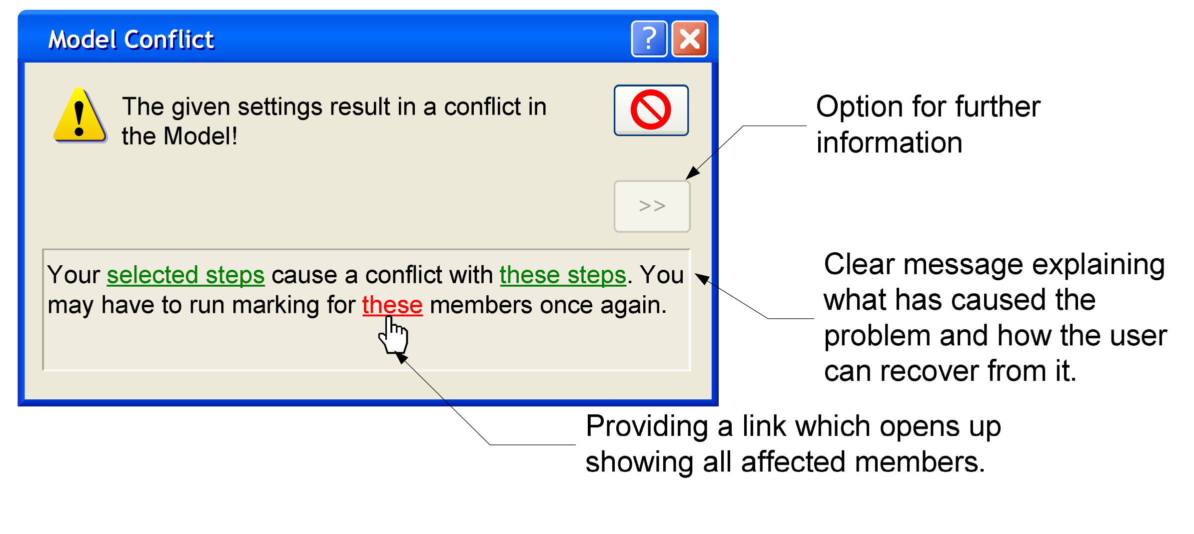


Fig. 5-47: Typical feedback for “Undo” in case of a conflict in the model

- Keyboard accelerators: As in all Microsoft applications, this function must be supported by the keyboard accelerators Ctrl+Z for Undo and Ctrl+Y for Redo. These accelerators must function with both categories of commands (graphical and modelling) as explained in the first feature.

Table 5.1 summarizes these features and their availability in Bocad-3D and TeklaStructures. The comparison shows that Bocad-3D offers relatively a very good Undo-feature for modelling commands. The proposed solution for an effective undo function is shown in Fig. 5-48.

Feature	Bocad-3D	TeklaStructures
Availability of undo/redo:		
Modelling	✓	✓
Graphic	X (inactive by default)	✓
Feedback/Further inquiry	✓	X
Undo/Redo history	✓	X
Arbitrary steps for Undo	X	X
Auto-activating behaviour for Redo	X	X
Undo-in-between	✓	✓
Keyboard accelerators*	✓	✓

Table 5.1: Summary of features for Undo/Redo in Bocad-3D and TeklaStructures

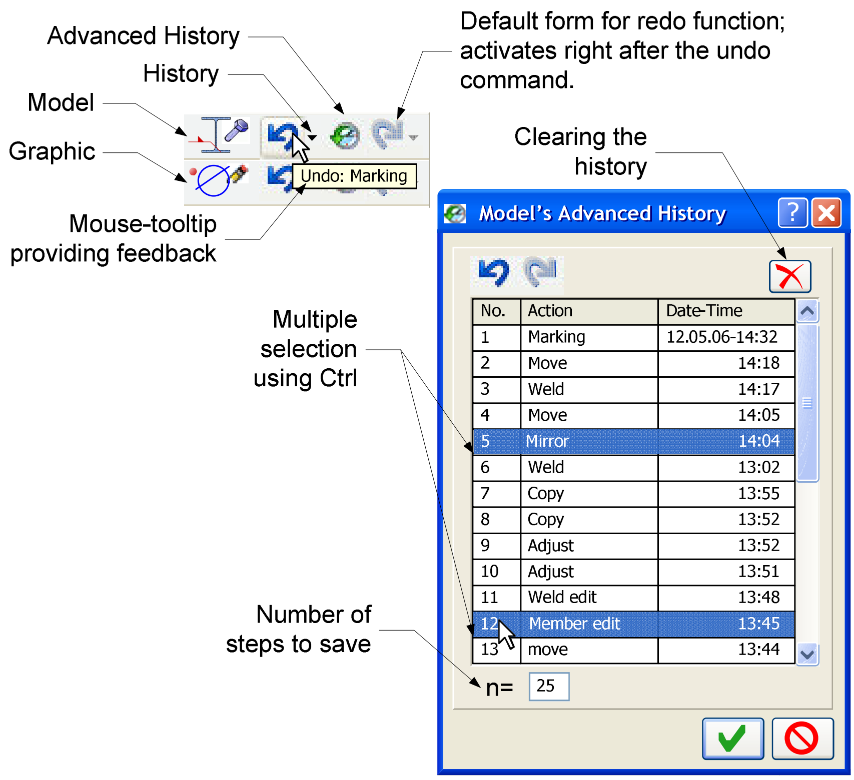


Fig. 5-48: The icon-bar for Undo/Redo presenting the discussed features

6. CAD/CAM Cooperation in Worldwide Networks, International Markets

One of the major benefits of the transfer of advanced CAD/CAM technologies to developing countries, discussed in this thesis, is that engineering teams in these countries can take advantage of the technology in collaborative efforts with European (German in our specific case) engineering enterprises. Such cooperation, also known as offshore outsourcing, could be beneficial on the European side as well, given the numerous advantages of outsourcing (see Section 6.1). In this context, although the technology itself is a crucial prerequisite of beneficial cooperation, it is not the only requirement; there are also other bottlenecks and concerns that must be thoroughly considered and discussed by both sides prior to making any investment decisions. Hence, in this chapter, some of the aspects of such cooperation are discussed that may remain hidden and unexplored but are vital for successful joint projects. This chapter intends to emphasise that even after technology is transferred successfully, there is still a long way to go for developing countries to catch up with their European partners. Furthermore, there is no intention to promote a complete or turnkey solution but rather to discuss basic challenges daunting such cooperation and, thereby, incite awareness on both sides. Indeed, if engineers, on a global scale, are made aware of and prepare for potential problems and the pitfalls of such joint projects, only informed decisions will be made and less friction will occur. Hopefully, the present discussion shall manage to contribute toward this aim.

Here, the discussion of challenges and their solutions is narrowed down to a case study of the cooperation between a German CAD office, Ingenieur-Contor-Weckmann (ICW) in Bochum, and an Iranian engineering office, Saraman Isfahan. ICW is the system leader generally responsible to the customer, including the role of agent looking for orders, clarifying the task, consulting the customer, managing revisions, executing pilot projects, budgeting and the settling of all accounts. Saraman, on the other hand, is a young engineering enterprise with a keen interest in transferring and implementing German high-level CAD/CAM technology. Although, in this specific investigation, neither side may completely represent their peers in Europe and developing countries, they can fairly well demonstrate a typical scenario.

At first, it is necessary to define the problem more accurately by defining the various aspects such as stakeholders and tools as well as specific services and expectations. Table 6.1 lists these.

Stakeholder	Client: A German/European manufacturer supplier	
	(Sub-contractor) Outsourcing company: A German steel detailing office	
	(Outsourcer) Outsourcing service provider: An Iranian engineering office	
Outsourcing service	Connection design, detailing	
Final product	Complete set of drawings for the object of the contract, which includes workshop drawings, single part drawings and assembly drawings as well as NC data and part lists	
Basic tool	Advanced 3D CAD software programs such as Bocad-3D or TeklaStructures	
Quality requirements	Conformity to project specifications; Compliance with specified project codes; The end product must not only accord with European standards for detailing but also facilitate seamless fabrication, manufacturing and assembly.	
Bottom line of expectations	German side	Adherence to deadlines Measuring up to quality Financial profit in €
	Iranian side	Financial profit in €

Table 6.1. Aspects of outsourcing CAD services

6.1 Outsourcing: Advantages and Disadvantages

The most significant advantage of offshore outsourcing is cost savings, which can be substantial—up to 40% [76, 81]. This is possible because foreign personnel in most developing countries require much lower salaries. However, studies suggest that there is in fact more to outsourcing than just cost savings. A recent survey conducted by PricewaterhouseCoopers and comprising 226 senior operating executives of private sector corporations [80] shows that, beyond cost savings, a second major advantage is "access to capabilities—whether human talent, process excellence or sheer physical resources". The study further reveals that "strategic benefit" is the third incentive for companies to embark on an outsourcing effort; this includes “freeing up one’s own resources, improving flexibility, gaining access to capital, geographic expansion, access to new markets, or changing the rule of competition and dealing with load peaks”.

Despite potential benefits, offshore outsourcing has certain drawbacks that are almost independent of the outsourcing product per se. Aside from any political and economical instability in developing countries, which may drastically disrupt the cooperation and cause the investment to be made in vain, there are specific uncertainties about the quality of the end service. Poor service quality, delayed deliveries and lag time for problem solving and clarifications are drawbacks that may negate the main benefits of offshore outsourcing [78, 79]. These downfalls may discourage companies to outsource their services; for example, an online survey in 2004 carried out by Steria Mummert Consulting in Germany [81] shows that only a twelfth of German enterprises are planning to outsource any projects offshore. Another survey in 2007, comprising 175 civil engineering firms, headquartered principally in New York, reveals that only one-third of these firms see outsourcing to be important “in their future growth plan” [77].

Furthermore, the drawbacks may be aggravated in the field of steel detailing mainly due to the following reasons:

- To measure whether outsourcing is working, and how well, Service Level Agreements (SLA) are established which constitute an essential part of such projects. A SLA functions as a performance metric of the service provider and is therefore used to determine if the provider has fulfilled the requirements [82]. There is no specific norm or standard for testing and controlling the quality of steel detailing. Compared to other civil engineering products such as calculations, design sheets and workshop drawings, there is no systematic way to determine quality along the lines of Six Sigma, ISO 17025 or ISO 9000. Within other disciplines, it is possible to carry out sample tests or demand standard certificates from the outsourcing firm. In detailing, however, such possibilities are still nonexistent. Quality checks are carried out purely based on experience. This is why more experienced staff is usually assigned such a task. Likewise, there are no other indicators of the capabilities of a firm aside from prior experience in a field of work. As a result, indicators and standard checks of quality are missing in the field of civil engineering, especially in the area of detailing.
- Some sectors, such as IT services or parts machinery, deliver standardised products in large amounts, e.g. by mass production. This brings the decisive advantage of quality maintenance, meaning that once a product is controlled and checked for quality, there will be an acceptable, statistically-approved guarantee that it can be reproduced on a large scale with quality maintained. In detailing or civil engineering, however, every project is basically different. Therefore, once a

project is finished and constructed, there is no systematic way to guarantee the same level of quality for the next project even though the know-how remains in the firm. As a result, not only quality checks fail to be standardised in civil engineering, quality maintenance is also hard to achieve.

- In other disciplines, poor or lower quality in an end product *may* either be tolerable somehow or remain hidden, even from the client or the end customer. A good example in this case is the outsourcing of software products, which may contain bugs even after extensive testing and Beta releases; end users may find such errors typical and even become used to. In detailing, on the other hand, as the author has experienced personally, errors have immediate financial consequences. These range from reworking and revisions of drawings, which is the least impactful, on to drilling, grinding and/or sawing on site, which often results in waste material. Worst of all are contractual penalties and insurance fees, which may be substantial enough to jeopardise the existence of the company itself. Therefore, lower-end service quality in detailing, as compared to other disciplines, is more likely to lead to liability issues for outsourcing companies.

These basic differences clearly demonstrate how crucial it is for outsourcing firms in the field of steel detailing to find (and educate) the right partner that is not only competent and knowledgeable, but also reliable. This would explain why outsourcing in the field of steel detailing or CAD/CAM is not as common as other engineering or industrial services. Evaluating and relying on a partner based in a developing country is no easy feat. Even with the assumption that a foreign outsourcer is competent locally, there is no guarantee or proof available to indicate that the service they have to offer meets world-class standards.

Nonetheless, there is incentive enough for European or American companies to outsource their services overseas so as to maintain competitiveness. Altogether, successful cooperation is for both sides not only lucrative but also an effective way to develop local and/or regional competitiveness. For outsourcing firms, this is achieved through substantial cost savings whereas outsourcer firms attain knowledge of the latest innovations and the transfer of technology. Furthermore, international cooperation among partners of disparate welfare and development offers the chance to optimise workload sharing so that each side can perform what they do best. Workload sharing ideally leads to “efficiency leadership”, allowing partners to be highly competitive on global markets. This strategy has enabled German enterprises to be the export leaders in the highly-demanding markets of consulting and technology.

6.2 Decision upon an Outsourcing Service

An important decision for European companies interested in outsourcing projects to make is which part of their service should be performed overseas. Is it possible to maintain more or less the same inland business model abroad, or is it necessary to make certain adjustments to it? What adjustments need to be taken and what are the determinants? An investigation of export strategy in the case of GOLDBECK enterprise, one of the most successful construction enterprises in Europe, may shed light on some of these issues. This family-owned German enterprise has had a very successful presence in Europe over the past decade (see Fig. 6-1) to the extent that up to 40% of their annual turnover has been achieved from service exporting [83]. This is most exceptional in a decade when Germany lost more than 800 000 jobs in civil engineering from the 1 600 000 that existed before, and many construction companies went bankrupt.

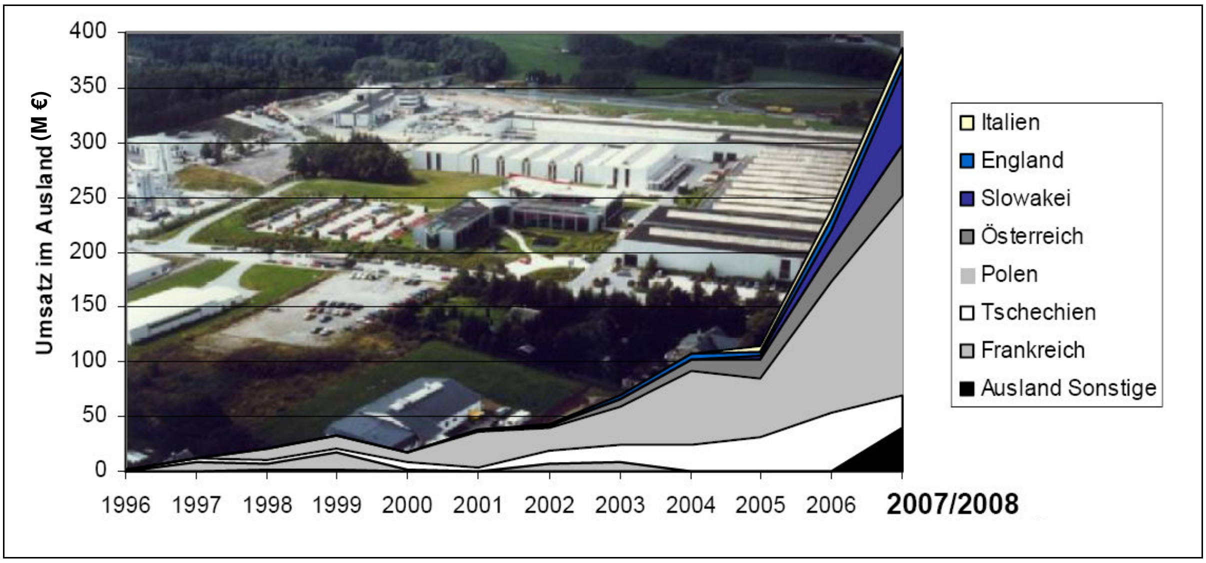


Fig. 6-1: Growth of the engineering company GOLDBECK in Europe during the last decade (Courtesy GOLDBECK)

By systematic and thorough product development, this enterprise became most efficient in all steps of planning, fabrication, and maintenance in their home market (i.e. Germany). This way, market leadership was achieved for promising product lines such as logistic centres, industrial buildings, multi-storey cark parks, and sport halls. However, to enter the new export markets with a large range of these product lines did not prove profitable. Instead, they embarked on a gradual, systematic process to penetrate into the export market. The company started its successful presence in Europe by constructing logistic centres in Poland when this country joined EU—an excellent starting point. This established a reputation for GOLDBECK in Europe as the most efficient supplier of logistic centres, carrying out their work with impeccable quality and in a timely fashion. Amazingly, despite fabrication in Germany, with its high cost of workmanship, the price offered for logistic centres was most competitive and generated higher profits than in their home market. This is a logical consequence of efficiency gained by systematic product development. Based on this well-earned reputation and economic growth in Poland, the subsequent product lines of GOLDBECK, such as multi-storey car parks, have been introduced with profit.

Altogether, their key export strategy may be summarized as follows:

The export business model must be a simplified, limited version of an inland business model in order to overcome the inherent risk factors in overseas business. Reduction of process complexity and maintenance of brand recognition as well as effective identification of local market opportunities are the most significant factors in product selection. The road to success involves focusing on the target market niche as well as core competency.

Although this service export strategy is specific to GOLDBECK and may not therefore be directly applicable to steel detailing, it provides some interesting ideas. Firstly, the complete inland service should not be outsourced; secondly, inland process complexity must be reduced when going offshore; lastly, great effort should be taken towards identification and maintenance of the market niche and core competency.

For German steel detailing companies, their service diversity may be understood by their various clients since every client has different, if not necessarily unique, requirements and specifications. In the scope of a German-Iranian outsourcing project, German companies, as the source of know-how, are capable of adapting themselves to a variety of clients. They may not even have much problem in dealing with new clients, as long as the prior expertise remains applicable to the most part and does not require many adjustments. This flexibility in dealing with different clients forms their inland business model. However, for the Iranian partner, this may not be as feasible since their know-how has been transferred and they have gained it through training. Thus, given the language and distance barrier as well, they might not be able to adapt it to new circumstances or specifications as easily. Hence, the first fundamental issue for a German company to be resolved is what area(s) of service should be outsourced. This is also important when considering the fact that all clients may not agree with the subcontracting of a project to a third party including an Iranian engineering team[1]. Once this is decided and clarified, effort must be made from the early stages of training to systematically concentrate on the specified clients and their standards/specifications. In other words, involvement of the team in projects involving several clients should be avoided. This approach clearly reduces the

[1] Private communication with Mr. T. Weckmann, Chief Manager of ICW GmbH

complexity of dealing with new specifications and circumstances. It therefore paves the way to achieving the second idea, i.e., reduction of complexity. The new team can concentrate more effectively on certain specifications and can start producing earlier. Finally, the so-called “target market niche” and “core competency” can already be identified during training. If the team manages to work productively while adhering to quality requirements, there is a good chance they can deliver at the same level when working remotely later on.

6.3 Training

One of the basic prerequisites of successful outsourcing is that the service provider possesses or obtains the same relevant level of know-how as the outsourcing client; otherwise, the same quality is unlikely to be delivered. Hence, in any outsourcing project, know-how transfer occurs either automatically, simply due to the nature of the collaboration, or more consciously—by conducting training courses, for example. On the other hand, know-how itself is an issue which differs even among domestic firms; in the case of German and Iranian firms, such differences are actually more apparent due to varied cultural and educational backgrounds. Therefore, it is inevitable to consciously accommodate a training period in the framework of cooperation between German and Iranian CAD offices. Both sides must be aware of this issue and free up room in their investments or expectations for it. This further means that they cannot expect an expedient kick-off to an outsourcing project; they have to invest time and capital on training and know-how transfer. Finally, they cannot expect a quick return from their investment.

In this situation, it may sound too demanding, especially for the German side, to financially justify an outsourcing project with Iran. Indeed, it is an absolute challenge and there is no intention here to refute this fact. On the contrary, this issue is emphasised to show that it is necessary to stake out and discuss the challenges for both sides, so that they can adjust their efforts and expectations correspondingly. In short, there is no attempt here to claim “German-Iranian outsourcing made easy” but rather “made possible”.

As next, basic cultural and educational differences are discussed to support the idea of why a training period is necessary. Afterwards, the experience made in this research for a training course is given as an example that can present useful information for future cooperation.

Cultural and educational differences

At first, it is necessary to digress slightly and clarify this term in text, since it might potentially bring about a broad range of interpretations and thereby lead to sweeping generalisations irrelevant to the scope of this research.

- By the term “cultural differences”, it is meant that differences in mentality in the following areas, towards which Iranians tend to have a loose approach in comparison to Germans:

  - Attitude towards time and, especially, deadlines;
  - Accuracy at work;
  - Mental desire for realisation of utmost quality;
  - Customer-oriented business.

- By “educational differences”, it is not meant that differences in theory but rather differences in approaches and procedures towards *applied* theories and knowledge. Indeed, knowledge of theory is one thing, but applying it in reality in a tough market and gaining a competitive advantage is another. The fact is, market success is a product of applied theory together with efficiency, productivity and the push for premium quality—as the customer demands it. Reluctance from the Iranian partner over the latter assets is one of the main obstacles to successful international collaboration. For this reason, this subject is brought up here for further discussion.

At universities of Iran, terms such as “efficiency”, “high-performance”, “cost-effectiveness”, “team work” and “productivity” are relatively unexplored and unfamiliar. This has also been observed by Pulst and Finkbeiner in their cooperation with universities in Tehran [84]. It is especially true with regard to civil engineering departments in Iran. Theoretically, students are familiar with these terms. Teachers sometimes mention them in their lessons, but how these assets are to be achieved and realised in industry, and with which technologies, is not discussed.

The main focus at Iranian universities is dedicated to pure theory. For example, in the Isfahan University of Technology, where the author received his bachelor and master’s degrees, there is great emphasis placed on structural mechanics and finite element analysis, for which students learn to harness the power of computers. Exactly how to realise all the structural concepts on site, which is indeed a more fundamental question making the theory worthwhile, is something that is never discussed in detail. In other words, while students become very familiar with FE analysis programs, they barely get to know the advanced building technologies such as CAD/CAM systems or the state-of-the-art in detailing, product development and systematic design.

In the framework of an outsourcing project, this causes difficulty in the successful transfer of know-how since many engineers or graduates are likely to not fully comprehend the real significance of applying such software systems. The consequence is, although they show interest in learning the new software to deal with basic problems, they are likely to become complacent and lose the ambition to deliver the required quality in time.

Another issue that may pose problems in training as well as staffing is the status of steel detailing as an engineering job from the perspective of a graduate. Due to the prevalent approach at universities in Iran, especially in the last ten to twenty years, structural design seems to be the only prestigious occupation in the entire steel industry. Therefore, top graduates usually choose an office job as a structural designer. In other words, detailing is not considered a well-distinguished job for an engineer since it is normally done by drafters and not engineers. This is also partly due to the image of the old style of detailing methods by 2D drafting instead of 3D modelling using advanced CAD/CAM technologies.

These differences clearly show that the Iranian team has to be set up and adequately trained to meet the requirements of an international cooperation. This is clearly one the most challenging phases in establishing international cooperation between German and Iranian enterprise.

During this research course, several attempts at training workforces failed and resulted in the loss of more than 30 000€ in investment. At first, it was decided to send experienced Iranian engineers to Germany for training on the job. This is actually the most direct way of training that can help trainees develop the essential mentality at work under real-life conditions. Moreover, it was decided to have engineers with several years of practical experience in order to lower the amount of supplemental training. However, problems arose when trainees decided to leave the project, and their Iranian employer, by the end of the training period in order to work freelance with the newly gained experience and, thereby, even became competitors! In fact, having one year of excellent training in Germany is basically adequate for someone to work freelance in Iran although they would need more experience and training to work in Germany. There were also cases where experienced engineers came to Germany for training but could not rid themselves of their prior habits and mentality, making the training costly on the German end. It is noteworthy to mention as well that in two cases the trainees were not civil engineers, but chosen merely because they had experience in drafting and/or detailing in Iran.

Having experienced these failures, it was decided to modify the approach not only by recruiting new engineers but also to train them differently. This formed the third experiment in this research. The modified plan was to employ young engineers with three to four years of experience and first train them in Iran before bringing them to Germany. Moreover, their future commitment to Saraman, as the investor of this training, was secured by financial means. Afterwards, a three-stage training plan was executed for the new group:

- **Initial training in Iran:** Initial training was carried out by the author using the

Bocad-3D program with seven young engineers in Isfahan for three weeks. The objectives of the initial training were not limited to learning the program, but also the basics of European steel detailing. Moreover, regarding the fundamental differences discussed previously, the essential working mentality had to be instilled at this stage early on. To meet the objectives, one of the real-life projects carried out by the author at ICW was selected. All of the necessary steel detailing rules for the case were thoroughly explained. Moreover, a deadline was assigned so that the trainees could practice a typical working environment. It was revealed that the work on trainee mentality had the highest priority, for many accuracy failures were encountered in the first days. Failure to resolve this could lead to disputes as well. However, with young trainees, it is likely to work out after two weeks since they will get used to the new mentality or at least begin to think and work differently by comprehending the prerequisites of cooperation at an international level. They may find some facts very surprising, or even funny. Examples of this are the need to work accurately to the millimetre, consideration of all details to produce at the desired level of quality and the fact that great accuracy with an advanced CAD system is not an option but a must.

By the end of the classes, an evaluation was made as to which members of the work force were ready to proceed to the next stage. Since detailing was quite a new topic to the trainees, it was decided that they should not be evaluated based on technical criteria, but rather on criteria exposing their interest and adaptability to future improvements. In selecting these criteria, the experience of the author working at a German company was clearly helpful. Table 6.2 shows the results of this evaluation on a scale of one (hopeless) to ten (very good).

Trainee	**A**	**B**	**C**	**D**	**E**	**F**	**G**
Sex	F	M	M	M	F	F	F
Regular presence in classes	10	9	6	10	10	7	10
Previous work experience	Yes	Yes	No	Yes	No	Yes	No
English proficiency	5	3	7	6	3	6	5
Trainee's interest	9	7	9	9	7	8	9
Learning ability	7	7	8	10	7	9	8
Punctuality	9	7	8	10	10	7	8
Efficiency	6	6	6	9	6	10	7
Accuracy	6	5	6	9	6	7	7
Self-initiative	2	3	3	9	5	7	5
Accountability	4	4	6	9	5	8	7
Seriousness at job	7	5	8	10	8	8	7
Effort	9	7	8	9	8	6	9
Overall evaluation	**6,7**	**5,7**	**6,8**	**9,1**	**6,8**	**7,5**	**7,5**

Table 6.2: Evaluation based on the first stage of training

- Follow-up training for one month in Iran: The objective here was to pursue long distance supervision/cooperation. In this case, a finished project was assigned to the team again and all of the information needed to successfully complete the project within a month was also provided.

Although simulated projects do not fully model the situation inherent to a real project, they can present useful information about the candidate team, such as their teamwork, responsiveness in distant communications and, especially, reliability when there is no direct supervision.

- Training for one year in Germany: After the follow-up training, the best trainees were chosen to participate for one year in Germany to take on real projects. This way, they could experience the real-life situation; something they definitely need before they return to Iran and cooperate on global efforts with German colleagues.

Due to a variety of reasons, however, only one of the six selected ended up coming to Germany. Three trainees were not present at the first and second training classes and two others received low evaluation marks (Trainees B and E in Table 6.2). With this setup, which almost disregarded the evaluation of the author, the success of the team was left more to luck than thorough planning and preparation. Therefore, it is no wonder that the German team felt the need to halt the project after two months of increasing difficulties. Exactly why this team setup was chosen is deemed trivial to discuss within the present scope. More important are the lessons to be learned which are presented in the next section.

6.4 Management

Management is not an easy job as it is, and even more difficult when it comes to managing an Iranian team that is to cooperate with a German engineering office. However, the effect of management as a tool for success should not be underestimated. Management plays the greatest role in forming effective staffing and retention strategies guaranteeing sustainable team efficiency and productivity, as these cannot be achieved only by providing the appropriate technology and conducting adequate training. To begin with, this section discusses some of the management pitfalls observed on this project that are to be avoided in any future cooperation of this kind. Subsequently, three aspects of human resource management are discussed. Based on observations of the author during this project in both the Iranian and German teams, these aspects are deemed important in the management of an Iranian team.

Lessons learned

Scrutiny of all the unsuccessful attempts to carry out this project, including the final one, reveals that certain attitudes of the managers and decision makers should have been revised and a different approach adopted.

i. Before anything begins, both sides need to communicate effectively by making their demands as well as their capacity as clear as possible. The experience in this research has shown that miscommunication or misunderstanding is very likely to occur even about the simplest points. There are issues that, although either side may take it for granted, turn out to be a point of conflict in the end. Examples of such are whether the trainees are civil engineers, whether they have had “relevant” working experience, how much they can be involved in a project, what they can or cannot do and the level of their education. In addition, issues of a more administrative nature, as well, have proven to be problematic, such as accommodation, insurance, vacation, let alone the matter of salary. Therefore, as the very first step, it is crucial that both sides come to a firm agreement about the terms and conditions of their contract regarding trivial issues as well.

ii. The decision makers from the Iranian side must bear in mind that steel detailing in Germany is a very critical job and by no means comparable to “typical” detailing tasks in Iran. This means:

1. A typical detailer/draftsman in Iran is not necessarily the right person for the job in Germany
2. The job applicant/trainee must be a civil engineer
3. Not everyone who (apparently) agrees with the conditions of employment, including a low salary to begin with and appears to remain loyal to the company can manage to fulfil the requirements when in Germany.

These three outspoken facts again underline the necessity of staged training and carefully-crafted evaluation processes prior to sending workforces to Germany. Unless the decision makers are interested in risking their time and capital, they are advised to avoid sending workforces to Germany that have not been evaluated beforehand. Another point requiring due attention is the capability of the team for stand-alone performance at the German company. In other words, it must be evaluated how well the team can cooperate, as well as communicate, with the German team. Are they fluent in English and/or German? How much supervision/control is needed when assigning a task to the team? How much mentality training is still needed? These are typical questions which may identify the potential need for a supervisor. It should be noted that such supervision should be seen as a part of training and demands specific investment that *both* sides have to meet. One of the reasons that the final experiment failed was the lack of such effective supervision and team leadership.

iii. For the Iranian side, it is very crucial to set fixed clear contracts with the employees from the beginning, i.e., at the time of recruiting. What has proved to be detrimental is “let’s start and we will see…” which means that the employees should first participate in the initial training without having any idea whether or not they are going to be paid for their time investment. It is even worse when an employer cannot keep his word regarding the salary range *for whatever reason* following the initial training, handling it in a nonnegotiable manner and yet complains when workforces leave! This was one of the reasons that, after the second training, some employees left the company. Furthermore, the contract must be clear about the salary the trainees are to receive during the training in Iran as well as in Germany. The training period in Iran, as well as the first two months in Germany, may serve as a probation period. Those who receive low evaluation marks in Iran should be excluded from the training in Germany. Likewise, those who fail to live up to expectations in Germany will have to leave the project as well. Of course, no exact salary figures can be set from the beginning. Nonetheless, certain salary ranges could be agreed upon, even for the period in Germany. This approach allows the company to send a positive message that employees are part of a significant, robust plan, which should be pursued in a thorough fashion. While this plan is supposed to serve the benefits of the company (in a later stage), it does not fail to consider the benefit to employees and addresses, among other things, the salary rate in the early stages. Decision makers should bear in mind that new employees are professionals who are there to make a living. Hence, it is very harmful to imply that managers are still confused about one of the most important topics to employees, i.e., their salary—even though apprentices can barely garner any benefit for the company at least for the first 6 months. Postponing this decision only manages to effectively “disconnect” new employees at the start. Since the company has to invest in training and, on the other hand, the team garners no significant benefit from the first months of training, it is very difficult to warrant a higher salary for the team. This means that a typical job applicant would have to accept an even lower salary to start off with than his/her previous income was. Even though this is logical (at least from the investors’ perspective), it may turn off many job seekers. Nonetheless, those who have the “job fit” and “company fit” may be attracted if the company carries out a decent staffing process as prescribed in [85]. Instead of dealing with this issue professionally by talking openly about it from the early stages of staffing and allowing employees to negotiate, decision makers tend to take on three misperceptions. All of these inhibit proper staffing/retention decisions to be made in hard times in addition to leaving the impression to employees that the management is unapproachable and conceited.

The first, and probably most fundamental, misperception is that decision makers expect new employees to realize the significance of the project on their own and see it as a long-term career investment. This complies with what Bill Silberman calls “intangible benefits of membership”. According to Silberman, intangible versus tangible benefits of membership, i.e. traditional pay and benefit plans, are “pride in organizational brand, pride in being part of something important, career development, and team spirit“[85]. Prior to making such expectations of their employees, managers should ask themselves whether they have created such an atmosphere in their enterprise so as to allow such “intangible benefits” to be understood. In other words, it is a critical task for the founders and upper-level management of an enterprise to develop the core culture of the company and base staffing and retention decisions upon it.

The second misperception is that decision makers are likely to think they are doing new employees a favour by sending them to Germany for training. Therefore, the employees should be willing to invest in the project by demanding less salary. Although they are providing employees the advantage of further training, they are not in fact doing them any favours. Without a doubt, it is a good opportunity, but every other enterprise seeking to develop a competitive edge in the market would provide expert training. It is all about give-and-take—the company provides the chance by investing capital and expects certain demands to be met in return. If the owners look at this professionally, they can also expect a professional outcome. However, if they treat employees as if they were doing them a favour, few if any top-notch job applicants would be attracted to the position.

The third misperception the author has noticed involves decision makers comparing the situation with academia, where a student receives a scholarship. Therefore, they expect that trainees should be content with non-favourable working conditions. However, for working professionals in the real world, “favour”, “gifts as in academia” and “the scholarship situation” are not attractive. After all, the rules of business and professional engineering are not only different from one another, but also miles away from academia. An applicant for a job on such a project and a scholarship holder may both tend to invest effort for less money in the short term to realize better career prospects in long-term. But nevertheless, they have potentially different understandings of the “tangible” and “intangible” benefits of membership for which they are applying. Therefore, decision makers would do well not to compare job-seeking professionals with students, and likewise not make them similar offers since this can damage the professional image of the company.

Assertiveness regarding company values and performance norms

A team of Iranian engineers aiming to cooperate with a German team should be established on the basis of working values and ethics, which may in part be completely different from the prevailing attitude in the region and/or nation. Therefore, it is likely to take time before values such as meeting deadlines, delivering acceptable quality, teamwork and accountability truly become part of the company culture. Here, management can certainly play a significant role in reinforcing these values among the personnel by coaching them for improvement.

There might be a tendency, due to cultural backgrounds, among some employees to continually bring excuses when they fall short of company values. In other words, there might be cases where employees still refuse to adopt the new working mentality and insist on their former working style. Such cases clearly jeopardise the success of the company. Management must thereby react assertively and firmly, but not aggressively, for this can be destructive and culturally unexpected in a country like Iran. Upon repeated instances, a dismissal is the inevitable consequence even though it might be costly for the firm. This assertiveness, on the other hand, sends a positive message to reliable employees assuring of proper managerial decisions in hard times when the existence of the company is on the brink of jeopardy. This was the case in the third experiment where the regular employees on the German side refused to continue the experiment when the extra workload for the teaching, supervising, checking and reworking of tasks became overwhelming.

However, this worst-case strategy must be practiced very cautiously since it backfires on the values of the company and influences the sense of security of employees. Reliable employees need to have the feeling that “they do not fear that their jobs will be in jeopardy if their performance is not perfect and one in which layoffs are considered an extreme last resort, not just another option for dealing with hard times” [86]. Such situations have extremely detrimental effects on employee morale and may limit their motivation, as O’Reilly and Pfeffer describe:

“…Most firms today emphasize, among other things, the employee's responsibility for being career resilient, employment at will and no-fault dismissal, pay for performance, downsizing to cut costs, and maximizing shareholder value above all else. What is the message any sentient employee takes from these practices? Pursue what is best for you, not the firm or the customer, adopt a free-agent mentality, and do not invest any more in the firm than it is willing to invest in you…”

Financial motives

Even if one claims that, for an Iranian engineering team with such good characteristics enabling them to cooperate with a German engineering team, money cannot be the only motive, it is nevertheless one of the decisive ones. It is not to forget that such a team has to develop a much higher sense of responsibility at work which, in turn, actually means “a harder life” when compared to their colleagues in Iran performing routine engineering tasks. Consequently, this “harder life” demands its own compensations and rewards. Otherwise, it would not take long before an Iranian team pulls apart. This is one of the critical decisions that both sides have to make when the team proves its capability and starts providing benefits. Even though the team has reached this state through initial investments of decision makers, the return on investment will not be the greatest concern of employees, especially when they are supposed to receive less salary than those doing routine tasks. It may be argued that, considering the differences in currencies and labour costs, this problem is not very likely to rise. Nevertheless, negotiation over salaries is to be expected.

Recognition and reward

Due to the nature of the job (steel detailing), which is demanding, it is obvious that devoted team members are necessary who have “other concerns rather than just money”. For example, engineers who are ready to work hard for the productivity and reputation of the firm. Moreover, it is obviously desirable to the firm that such members of the workforce stay loyal to the company in the long run. Based on the experience of the author, such individuals do exist in Iran and can be found and recruited for the purpose of international cooperation. Furthermore, they are likely to show extra motivation and enthusiasm at work, especially at the beginning when they are inspired by German engineering and technology. There is good reason for this, as Iranians want to prove they can work just as well and are trustworthy. This enthusiasm, if maintained in the long term, certainly adds significant value to a firm. Nonetheless, it is an absolute challenge for managers to determine how to gain and maintain such enthusiasm. This gives way to a discussion of recognition and reward in management. Sirota, et. al., give an interesting account of this issue [86]:

“Managers should be certain that all employee contributions, both large and small, are recognized. The motto of many managers seems to be, ‘Why would I need to thank someone for doing something he's paid to do?’ Workers repeatedly tell us, and with strong feelings, how much they appreciate a compliment. They also report how distressed they are when managers do not take the time to thank them for a job well done, yet are quick to criticize them for making mistakes. Gaining recognition for achievements is one of the most fundamental human needs. Rather than causing employees to be complacent, recognition reinforces their accomplishments, helping ensure there will be more of them.”

Experts believe that, for an organisation to be successful, recognition and reward are equally as important, if not more so, as financial motives. O’Reilly and Pfeffer put this forward very instructively [87]:

“…Although none of us would work for very long if we believed we were not fairly compensated, money by itself isn't sufficient for motivating really long-term high-performance. As David Russo has noted, a raise is only a raise for thirty days; after that, it's just your salary. Most of us would like to believe that what we are doing makes a difference to others and that our work is important. … Moreover, most of us also want to feel that we are valued as people, not simply as economic agents…”

7. Concluding Remarks and Future Work

User Interfaces (UI) in high-level CAD/CAM software systems for steel detailing are one of the decisive aspects for adaptation and localization of innovative technologies in developing countries. Market-leading CAD/CAM software systems of today are language dependent and relatively inconsiderate to usability concerns of even native-speaking users. These are indeed shortcomings, which pose many problems for novice CAD/CAM users in developing countries.

Fortunately, the interaction designers of steel detailing CAD systems can make an effective use of many metaphors taken from technical repertoire of all civil engineers around the world. This includes many standard graphical representations common in construction drawings. The use of such metaphors does not only lower the amount of textual languages in CAD systems, but also facilitates learning and allows CAD/CAM users to rapidly gain hands-on experience of the system.

In addition to metaphors, direct-manipulation style of interaction is the most effective solution for enhancing directness and intuitiveness in CAD systems. By presentation of common adjustment handles on objects of interest, the user can automatically understand the permissible actions. This style of interaction does not only lower the use of text for many common tasks, but also eliminates many syntactical complexities—resulting in more efficient interaction with the system.

The overview of guidelines and principles of software design in this research revealed that for a software system to be successful, it is crucial to base on the user’s prior knowledge and experience. Designers need to capture and validate the users’ mental models. Therefore, a user profile of CAD/CAM users in developing countries was elaborated which reflects limitations, capabilities and preferences of this user’s class for CAD systems. For users in developing countries, learning and exploration must be facilitated within the system. Moreover, the system must reinforce user’s confidence in use of the system by providing a “low risk environment”.

The survey conducted in this research reveals the most frequent commands in CAD systems for steel detailing. The results are useful for prioritising efforts necessary to develop intuitive text-free user-interfaces for international use. By participation in real life projects, the author was able to develop a correct insight of many tasks done when working with these systems. This was necessary to carry out a task analysis, which was a fundamental step for the development of the proposed solutions.

7. Concluding Remarks and Future Work

This research, furthermore, presented several examples of how user interfaces in existing CAD systems may be modified to become more learnable and direct. The emphasis here was not only on basic components, but also on important commands based on their frequency or functionality.

However, it must be noted that, even though these prototypes reveal many improvements, they must not be taken as a finalized design since no usability testing was carried out. This gives way to future research in a broader framework of usability testing for CAD systems in developing countries. Moreover, further research needs to be carried out to verify the applicability of the proposed solutions from the programming perspective.

In the end, some major challenges facing a German-Iranian cooperation in steel detailing were discussed. It was shown that even after a successful transfer of key CAD/CAM technology features to developing countries, there is still a long way for these countries to join in international projects with their European partners. This is mainly due to cultural and educational differences, for which engineers in developing countries need to be trained in scope of an international cooperation. Moreover, the influence of an effective management must not be underestimated to guarantee a sustainable working cooperation.

Appendix A

Approximate Confidence interval for the Mean of a Bernoulli Random Variable [73]

Consider a population of items each of which independently meets a certain standard with some unknown probability *p*. *n* of these items are tested to determine whether they meet the standards. The resulting data can be used to obtain a confidence interval for *p*.

If we let *X* denote the number of the *n* items that meet the standards, then *X* is a binomial random variable with parameters *n* and *p*. Thus, based on central limit theorem, when *n* is large, it follows by the normal approximation to the binomial that *X* is approximately normally distributed with mean *np* and variance *np(1-p)*.

Hence

$$\frac{X-np}{\sqrt{np(1-p)}} \approx \mathrm{N}(0,1)$$

which $\approx$ means “is approximately distributed as”. Therefore, for any $\alpha \in (0,1)$

$$\mathrm{P}\left\{-z_{1-\alpha/2} < \frac{X-np}{\sqrt{np(1-p)}} < z_{1-\alpha/2}\right\} \approx 1-\alpha$$

and so if *X* is observed to equal *x*, then an approximate $100(1-\alpha)$ percent confidence region for *p* is

$$\mathrm{P}\left\{-z_{1-\alpha/2} < \frac{x-np}{\sqrt{np(1-p)}} < z_{1-\alpha/2}\right\}$$

However the forgoing region is not an interval. To obtain a confidence interval for *p*, we use the fact that *X*/*n*, the fraction of items that meet the standards, is the maximum likelihood estimator of *p*. Hence it follows that $\sqrt{X[1-(X/n)]}$ will approximately equal $\sqrt{np(1-p)}$ and thus from above equation we can see that

$$\frac{X-np}{\sqrt{X(1-\frac{X}{n})}} \approx \mathrm{N}(0,1)$$

Therefore, for any $\alpha \in (0,1)$

$$\mathrm{P}\left\{-z_{1-\alpha/2} < \frac{X-np}{\sqrt{np(1-p)}} < z_{1-\alpha/2}\right\} \approx 1-\alpha$$

or, equivalently

$$P\left\{\frac{X}{n} - z_{1-\alpha/2}\sqrt{\frac{\frac{X}{n}(1-\frac{X}{n})}{n}} < p < \frac{X}{n} + z_{1-\alpha/2}\sqrt{\frac{\frac{X}{n}(1-\frac{X}{n})}{n}}\right\} \approx 1-\alpha$$

by replacing $\hat{p} = \frac{X}{n}$ and setting a confidence interval of 95%, which yields $z_{1-\alpha/2} = 1.96$, the final formula would be

$$CI = (L,U) = \left(\hat{p} - 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + 1.96\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right)$$

in which: CI= Confidence Interval; L= Lower limit of CI; U=Upper limit of CI; $\hat{p}$ = Calculated proportion or probability; n=Sample size.

References

*All Web-sites were last accessed on 28.02.2008

[1] Iran Daily Newspaper, No. 1549, Jan. 22nd, 2000, Page 7.

[2] Azadnia, H. R.: Investigating the Application of Information Technology in Optimum Design, Modelling and Construction of Industrial Steel Structures in Iran, Master of Science Thesis, Isfahan University of Technology, 2004.

[3] Mondialogo Publications:
<www.mondialogo.org/download.php?file=uploads/tx_bmmocmpress/MEA_ListeTop40_de.pdf>

[4] G. Pegels, SHR. Eslimy-Isfahany, H. R. Azadnia: "Proposal of Best Practice in Steel Detailing for Iran Based on Worldwide Experience", *Asian Journal of Civil Engineering,* Vol. 4, No. 4, 2004. pp.87-99.

[5] EN ISO9241 Ergonomic requirements for office work with visual display terminals, 2001.

[6] Karat, J., “User-Centred Software Evaluation Methodologies”, in Helander M. G., Landauer T. K., and Prabhu P. V., *Handbook of Human-Computer Interaction*, Elsevier Science B. V., Netherlands 1997.

[7] Bevan, N., Kirakowski, J., and Maissel, J., “What is usability?“ *Proceedings of the 4th International Conference on HCI, Stuttgart,* 1991.
< http://www.nigelbevan.com/>.

[8] Grudin, J., “Utility and usability” *Research issues and development contexts. Interacting with Computers*, Vol. 4, Issue 2, August 1992, pp. 209-217. Cited in [9].

[9] Nielson, J., *Usability Engineering. Academic Press*, Cambridge,1993.

[10] Microsoft Cooperation, 2000, Usability in Software Design.
< http://msdn2.microsoft.com/en-us/library/ms997577.aspx>.

[11] Johnson, J., “*GUI Bloopers: don’ts and do’s for software developers and web designers”*, Morgan Kaufman Publishers, San Francisco, USA, 2000.

[12] Folmer, E., Van Gurp, J., Bosch, J., “A Framework for Capturing the Relationship between Usability and Software Architecture”, *Software Process: Improvement and Practice*, Volume 8, Issue 2 , Pages 67 – 87, 2004.

[13] Miller, R., Lecture notes of MIT “User-interface design and implementation” taken from MIT online open courseware, 2004.

[14] Microsoft Corporation 2000, “UI Guidelines vs. Usability Testing”.
< http://msdn2.microsoft.com/en-us/library/ms997578.aspx>

[15] Schulten, D., „Programtechnischer Ansatz für Effzienz und Verständlichkeit von Bausoftware in weltweiten Netzen“. *PhD Thesis at Bergische Universität*

Wuppertal, 2007.

[16] McDonald, J. E., Schvaneveldt, R. W., "The application of User Knowledge to Interface Design", in Guindon, R., *Cognitive Science and Its Applications For Human-Computer Interaction*, Lawrence Erlbaum Associates, New Jersey, 1988.

[17] Foraker Desgin 2007, "Introduction to Usability". <www.usabilityfirst.com>

[18] Marcus, A., "Interface Design: Examples and Statistics", Aaron Marcus and Associates, Inc. (AM+A), 2002. www.amanda.com

[19] Theofanos, M. and Laskowski, S. "The Benefits of Usability in the Software Procurement Process", *National Institute of Standards and Technology*, U.S Department of Commerce, 2007 <http://zing.ncsl.nist.gov/iusr/documents/why_should_I_use_the_CIF_files/frame.htm>

[20] Landauer, T. K., *The trouble with computers: Usefulness, usability, and productivity*. MIT Press, 1995 Cited in [18]

[21] Raskin, J.: The Humane Interface: New Directions for Designing Interactive Systems, Addison-Wesley Professional, Boston, 2005.

[22] Basse, L., John., B. E., Kates., J., Achieving usability through Software Architecture, Technical Report, Carnegie Melon University, Software Engineering Institute, 2001.

< http://www.sei.cmu.edu/publications/documents/01.reports/01tr005.html>

[23] Folmer, E., Software architecture analysis of usability. *PhD Thesis, University of Groningen*, 2005 <http://irs.ub.rug.nl/ppn/287730794**>**

[24] Gould, J. D., Boies, S. J., Ukelson, J, "Human-Computer interaction: Background and Issues", in Helander M. G., Landauer T. K., and Prabhu P. V., *(Eds) Handbook of Human-Computer Interaction*, Elsevier Science B. V., Netherlands 1997.

[25] Bertaggia, N., Montagnini, G., Novara, F., and Parlangeli, O., "Product usability", in M. Galer, S. Harker, and J. Ziegler, *(Eds) Methods and tools in user-centred design for information technology*. Elsevier Science B. V., Netherlands 1992.

[26] Maurer, D., "What is Usability?", Step two Designes PTY LTD online publications, 2004 < http://www.steptwo.com.au/index.html>

[27] Mayhew, D. J., "Principles and Guidelines of Software User Interface Design", Prentice Hall, New Jersey, 1992.

[28] Norman, D. A., "Cognitive engineering", in D. A. Norman and S. W. Draper, *User Centred System Design*, New Jersey, Erlbaum, 1986.

[29] Wickens, C. D., Hollands, J. G.: Engineering psychology and human performance, 3rd Ed., Prentice Hall, New Jersey, 2000.

References

[30] Anderson, J. R., “Cognitive Psychology and its Implications”, Fourth Ed. Freeman and Company, NY, 1995.

[31] Wolford, G.: “Function of distinct association for paired-associate performance”. Psychological Review, Vol. 78. Cited in [32].

[32] Hitch, G. J.: “Principles from the Psychology of Memory: Part 1- Working Memory”, in M. M. Gardiner and C. Bruce, *(Eds) Applying Cognitive Psychology to User-Interface Design*, John Wiley & Sons, GB, 1987.

[33] Gardiner, M. M.: “Principles from the Psychology of Memory: Part 2- Episodic and semantic memory”, in M. M. Gardiner and C. Bruce, *(Eds) Applying Cognitive Psychology to User-Interface Design*, John Wiley & Sons, GB, 1987.

[34] Carroll, J. M. and Rosson, M. B., “Paradox of the active user”, *Interface Thought: Cognitive Aspects of Human-Computer Interaction*, ed. Cambridge, MIT Press, 1987, pp. 80-111.

[35] Hampton, J. A., “Principles from the psychology of Language,”, in M. M. Gardiner and C. Bruce, *(Eds) Applying Cognitive Psychology to User-Interface Design*, John Wiley & Sons, GB, 1987.

[36] Nielson, J., “Personalization is Over-rated”, 1998. <http://www.useit.com/alertbox/981004.html>

[37] Sanfor, A., J., “The mind of man: Models of Human Understanding” The Harvester Press, 1987.

[38] Anderson, J. R.: “Is human cognition adaptive?”, Journal of Behavioural and Brain Sciences, Vol. 14, pp. 471-484, 1991, Cited in [29].

[39] O’Hara, K. P., Payne, S. J.: “The effect of operator implementation cost on planfulness of problem solving an learning”, Journal of Cognitive Psychology, Vol. 35, pp. 275-289, 1998, Cited in [29].

[40] Simon, H. A.: “Invariants of human behaviour”, Journal of Annual Review of Psychology, Vol. 41, pp. 1-19, 1990, Cited in [29].

[41] Prabhu, P. V., and Prabhu, G. V., “Human Error and User-Interface Design”, in Helander M. G., Landauer T. K., and Prabhu P. V., *(Eds) Handbook of Human-Computer Interaction*, Elsevier Science B. V., Netherlands 1997.

[42] Norman, D. A., “Design rules based on Analysis of human error” *Communications of the ACM*, Vol. 26, Issue 4, 1983, pp. 254-58.

[43] Davidson, M. J., Dove, L., Weltz, J., “Mental models and usability”, *Depaul University, Cognitive Psychology 404*, November 15, 1999.

[44] Norman, D.A. “Some observation on mental models”, in D. Gentner and A.L Stevens, *(Eds) Mental models*, New Jersey: Erlbaum, 1983.

[45] Craik, K., “The Nature of Exploration Cambridge University Press”, Cited in online encyclopaedia of human-computer interaction, 1943.

< http://www.interaction-design.org/encyclopedia/mental_models.html>

[46] Encyclopaedia of cognitive science, Vol. 3, Editor in Chief Lynn Nadal. Nature Publishing Group, 2003, pp 1-6.

[47] Johnson-Laird, P.N. “Mental Models - Towards a Cognitive Science of Language, Inference and Consciousness”, Cambridge MA: Harvard University Press, 1983.

[48] Gentner, D., and Stevens, A. L., “Mental Models. Hillsdale NJ”, Lawrence Erlbaum Associates, 1983.

[49] Manketlow, K., and Jullian J., “Principles from the psychology of thinking and mental models”, Applying cognitive psychology to user-interface design, John Wiley & Sons, Inc., 1987.

[50] Norman, D., Psychology of everyday things. Basic books, New York, 1988.

[51] Hammond, N., Gardiner, M. M., Christie, B., and Marshall, C., “The role of cognitive psychology in user-interface design”, in M. M. Gardiner and C. Bruce, *(Eds) Applying Cognitive Psychology to User-Interface Design*, John Wiley & Sons, GB, 1987.

[52] Sasse, M. A.: Eliciting and describing users’ models of computer systems, PhD thesis, Department of computer science, University of Birmingham, 1997.

[53] Booth, P. A.: An introduction to Human-Computer Interaction. Lawrence Erlbaum Publishers, East Sussex, UK, 1989.

[54] Chang, Y., „Eine von Landessprachen unabhängige Nutzungsoberfläche mit intelligenten CAD-Objekten des Bauwesen“. *PhD Thesis at Bergische Universität Wuppertal*, 2001.

[55] Shneiderman, B., “Designing the user interface: Strategies for effective Human-Computer Interaction”, 2nd ed., Addison-Wesley, 1993.

[56] Marshal, C., Nelson, C., Gardiner, M. M, „Design Guidelines”, in M. M. Gardiner and C. Bruce, *(Eds) Applying Cognitive Psychology to User-Interface Design*, John Wiley & Sons, GB, 1987.

[57] Neal, D. C., Carroll, J. M., “The role of metaphor in user interface design”, in Helander M. G., Landauer T. K., and Prabhu P. V., *(Eds) Handbook of Human-Computer Interaction*, Elsevier Science B. V., Netherlands 1997.

[58] Shneiderman, B., “The future of Interactive Systems and the Emergence of Direct Manipulation,” *Behavior and Information*, Vol. 1, pp. 237-56 1982, Cited in [27].

[59] Shneiderman, B., “Direct manipulation: A Step beyond Programming Languages,” *IEEE Computer*, Vol.16, pp. 57-69, 1983, Cited in [27].

[60] Hutchins, E. L., Hollan, J. D., and Norman D. A., “Direct Manipulation Interfaces”, in D. A. Norman and S. W. Draper, *User Centred System Design*, New

Jersey, Erlbaum, 1986.

[61] Windows Vista User Experience Guidelines <http://msdn2.microsoft.com/en-us/library/Aa511258.aspx>

[62] Apple Human Interface Guidelines <http://developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/>

[63] Java Look and Feel Design Guidelines <http://java.sun.com/products/jlf/ed2/book/index.html>

[64] GNOME Human Interface Guidelines <http://developer.gnome.org/projects/gup/hig/>

[65] Bickford, P.: Interface design: The art of developing easy-to-use software. Chestnut Hill, MA: Academic Press, 1997, Cited in [11].

[66] Norman, D. A., “Design principles for human-computer interfaces”, in Applications of Cognitive Psychology: Computing and Education, Edited by D. E. Berger, K. Pezdek, & W. Banks. Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1986.

[67] Hardcopy homepage <http://www.info.hardcopy.de/>

[68] Miller, D. P: “The depth-breadth Tradeoff in hierarchical computer menus”, Proceedings of the human factors society 25th annual meeting, 1981, pp. 296-300. Cited in [27].

[69] Raktoe, B. L., Hubert, J. J., *Basic Applied Statistics*, Dekker, INC. New York, 1979.

[70] Fink, A., *How to sample in surveys*, Sage publications, 1995, California, 1995:

[71] Fink, A., *How to analyze survey data*, Sage publications, California, 1995.

[72] Moore, D. S., *Statistics, Concepts and Controversies*, Freeman and Company, San Francisco, 1979, ISBN 0-7167-1022-6.

[73] Ross, M. S.,*Introduction to Probability and Statistics for Engineering and Sceintists*, John Wiley and Sons, New York, 1987.

[74] Microsoft 1995. The Windows interface guidelines for software design: an application design guide. Redmond, WA: Microsoft Press. 1556156790.

[75] Pegels, G., and Weckmann, T., „Allgemeine Benutzungsoberfläche für rechnergestützt koordinierte, vernetzt-kooperative Planungsprozesse“, DFG-Forschungsbericht, 2006.

[76] Katrina, C. A.: “The pros and cons of offshore outsourcing”, ThomasNet Industrial Market Trends, 2002. < http://news.thomasnet.com/>

[77] CENews: “Survey highlights engineering firm trends and challenges”, 2007. < http://www.cenews.com/article.asp?id=2118>

[78] Fainsilber. O., Chadwick-Jones, A., “No Nasty Surprises—Anticipating the hidden risks of outsourcing”, Mercer Management Journal, <www.mmc.com>

[79] Scheiber, S. F.: “Is outsourcing for everyone?” SMT online publications, 2001. < http://smt.pennnet.com/home.cfm>

[80] PricewaterhouseCoopers Publications: “Outsourcing comes of Age: Rise of Collaborative Partnering”, 2007. <http://www.pwc.com/extweb/pwcpublications.nsf/docid/DF993563025DFA0F852572D60067ACA3>

[81] Steria Mummert Consulting Publications: “Erfolgsmodelle im Outsourcing”, 2004. < http://www.steria-mummert.de/>

[82] Hayes, I. S.: “Metrics for outsourcing Service Level Agreements”, Clarity Consulting, Inc. <http://www.clarity-consulting.com/MetricsforITOutsourcing.pdf>

[83] Goldbeck GmbH. Vortrag “Auslandbau: Chancen für den Mittelstand”. Bergische Universität Wuppertal, December 6th, 2007.

[84] Pulst, E., Finkbeiner, T.: Iran im Informationszeitalter. Eul Publications, 2006.

[85] Harris, J., Brannik, J.: Finding and Keeping Great Employees. AMACOM, Broadway, New York, 1999.

[86] Sirota, D., Mischkind, L. A., Meltzer, M. I.: “Stop Demotivating Your Employees!”, Harvard Management Update, Vol. 11, No. 1. Cited in Sirota, et. al., 2006, “Why your employees are losing motivation?” < http://hbswk.hbs.edu/archive/5289.html>.

[87] O’Reily, C. A., Pfeffer, J.: Hidden Value: How Great Companies Achieve Extraordinary Results with Ordinary People. Harvard Business School Press, 2001. Book excerpts were taken from < http://hbswk.hbs.edu/archive/1734.html>.

Lebenslauf

Hamid Reza Azadnia

Persönliche Angaben:	Geburtsort: Teheran, Iran Geburtstag: 21. März 1979 Familienstand: Verheiratet, keine Kinder
Schulbildung:	
Sep. 1985 - Juli 1990	Grundschule in Isfahan
Sep. 1990 - Juli 1997	Shahid Beheshti Gymnasium in Isfahan
Ausbildung:	
Sep. 1997 - Juli. 2001	Bachelor Studium des Bauingenieurwesens an der “Isfahan University of Technology (IUT)“
Sep. 2001 - Juli 2004	Masterstudium: „Konstruktiver Ingenieurbau“ an der IUT Masterarbeit: “Investigating the application of Information Technology in Optimum Modelling and Design of Industrial Steel Structures”
Sep. 2001 - Juli 2004	Wissenschaftliche Hilfskraft an der Bauingenieurfakultät der IUT
Okt. 2004 - März 2005	DAAD-Stipendium für den deutschen Sprachkurs bei „S+W Gesellschaft für Sprachunterricht mbH“, Marburg
Seit April 2005	DAAD-Promotionsstipendiat im Fach Bauinformatik an der Bergischen Universität Wuppertal
Praktika:	
Juli 2001 - Dez. 2001	Praktikum als Statiker bei “Sepahan Beyond Research Consultant Eng. Co.“, Isfahan, Iran
Praxiserfahrung:	
Dez. 2001 - Juli 2003	Statiker bei “Sepahan Beyond Research Consultant Eng. Co.“
Aug. 2005 - Juli 2006	Experimentelle Untersuchungen zur Doktorarbeit als Projektingenieur Stahlbau bei “Ingenieur – Contor Weckmann GmbH”, Bochum