Evaluation of Educational Environments
The TUP model

Roman Bednarik

University of Joensuu
Department of Computer Science
Master's Thesis
11.12.2002

Abstract

The evaluation of educational environments is a field in need of an interdisciplinary approach. As computers become more and more pervasive, educational computing follows this trend in the classrooms. It creates more dependencies between the parts of the learning process and it is more difficult to assess these complex relations. During the last decades the attention on usability evaluation has created many evaluation methods, which unfortunately are mostly of very limited use in the evaluation of educational environments. However, the inspiration they provide is valuable.

This thesis argues that joining the idea of a usability checklist in the form of a questionnaire with the proper pedagogical and technological issues facilitates the usability evaluation process of educational environments. The main requirements of assessing technological, usability and pedagogical factors are fullfilled. The Technology-Usability-Pedagogy model (TUP) is established and the TUP questionnaire is developed.

The ACM Computing Classification System (1998 version): H.5.2, K.3.1

Keywords: *User Interfaces, Evaluation, Computer Uses in Education*

Abstrakt

Vyhodnocování výukových prostředí vyžaduje spolupráci několika vědeckých disciplín. V dnešní době počítače stále více pronikají do našeho prostředí a života a jejich nasezení ve vzdělávání kopíruje tento trend. Vznikají tak vazby a interakce v historii nepředvídané a jejich vyhodnocení se stává obtížnějším. V uplynulých letech vzniklo několik úspěšných metod pro vyhodnocování klasických uživatelských rozhraní. Bohužel, většina z těchto prostředků je pro vzdělávací software použitelná jen za cenu velkého omezení. Co je však na nich z našeho pohledu užitečné, je inspirace kterou představují.

Tato práce argumentuje, že spojením myšlenek seznamů ve formě dotazníků a vhodného pedagogického a technologického teoretického základu lze dosáhnout podstatných vylepšení a výsledků ve vyhodnocování vzdělávacích prostředí. Je vytvořen Technology-Usability-Pedagogy (TUP) model a na jeho základě zkonstruován TUP dotazník.

ACM klasifikace (verze 1998): H.5.2, K.3.1

Klíčová slova: *Uživatelská rozhraní, Vyhodnocování, Computer-assisted Instruction, Použití počítačů ve vzdělávání*

Abstrakti

Oppimisympäristöjen arvioinnissa tarvitaan poikkitieteellistä lähestymistapaa. Tietokoneiden yleistyessä yhä enemmän, tietokoneavusteinen opetus seuraa tätä kehitystä luokkahuoneissa. Tietokoneiden käyttö opetuksessa luo riippuvuuksia oppimisympäristöjen eri osatekijöiden välille, jolloin näitä monimutkaisia suhteita on vaikeampi arvioida. Viimeisten vuosikymmenien aikana käytettävyyden arviointiin on kehitetty monia arviointimenetelmiä, mutta valitettavasti niitä voidaan soveltaa vain rajoitetusti oppimisympäristöjen arvioinnissa. Menetelmät antavat kuitenkin arvokkaita virikkeitä opetusohjelmien käytettävyyden arviointiin.

Tässä tutkielmassa esitetään, että yhdistämällä tarkastuslistoihin perustuva kyselylomake teknologisiin ja pedagogisiin asioihin helpotetaan oppimisympäristöjen käytettävyyden. arvioimisprosessia. Näin teknologisten, pedagogisten ja käytettävyystekijöiden arvioinnin päävaatimukset voidaan täyttää. Tutkielmassa luodaan malli teknologisille, pedagogisille ja käytettävyystekijöille (Technology-Usability-Pedagogy, TUP) ja esitetään kehitetty TUP-kyselylomake.

ACM – luokat (The ACM Computing Classification System. 1998 version): H.5.2, K.3.1

Avainsanat: käyttöliittymät, evaluointi, arviointi, tietokoneen käyttö opetuksessa

Preface

I was always angry with the things which are confusing and not working naturally, always thinking how to design and construct them in the better way. Unfortunatelly, I did not become a designer. Instead of being a designer, I have found myself being a reviewer. It is necessary to point out here that one can target his critiques either with the destructive or constructive contemplations. Let me categorize myself into the latter group and conceive the following chapters as an instruction how to make a review.

I would like to thank to my supervisor, professor Markku Tukiainen, for his inestimable advices, comments, discussion and patience. Special thanks also go to Ramin Miraftabi, not only for revising the language of this work.

I want to warmly thank to my parents and sister Lenka, and to Kaisa, to my friends for the support and power which overcomes the borders. Thank you.

Contents

1 INTRODUCTION

During the last decades, user interfaces have developed from textual command-line driven terminal screens to multimedia environments connecting the whole population. Designers and other persons involved in the development of user interfaces have learned that without the proper evaluation of prototypes, designs and released interfaces they can not accumulate enough knowledge in order to achieve improvements in usability. Computers have pervaded into the various fields of human activity. Nowadays, we can find them nearly everywhere, and education is not an exception. An initial reason of why to develop just another usability evaluation method is that we want to properly assess this intersection of computing and education. We want to enable educators to easily select the learning environments they use.

1.1 The goals of this thesis

This thesis has a few purposes: first to introduce to readers, get them familiar with and establish basic terminology concerning the evaluation of educational environments. The second purpose is to develop a model by which we would be able to carry out a peer review and properly evaluate educational software also with nonprofessional evaluators. Thus, we emphasize the importance of the role of a peer review in the selection of educational software. It is a teacher who is responsible for this selection and the following thesis should establish a framework facilitating an evaluation. There are plenty of usability evaluation methods readily available, but not each of them can be used for our purposes. Also many studies concerning educational environments have been carried out. However, most of them do not fit into the requirement of the evaluation of already deployed environments in order to enable simple but yet comprehensive evaluation and comparison of educational environments.

The core of the work is divided into the five fundamental parts, beside the introduction and conclusion: first we focus on traditional usability issues, then move on to the technological considerations, while the third part is involved with the pedagogical aspects of learning environments. Existing evaluation approaches to educational settings are outlined in the fourth part. The fifth part then integrates the knowledge of these three basic cornerstones together, creating an evaluation method which is intended to fit into the educational setting.

1.1 Background

The following paragraphs introduce and define the basic terminology concerning usability evaluation as an approach of joined efforts of Human-Computer Interaction and software engineering.

Usability and Human-Computer Interaction

Nowadays usability is quite a broad term discussed within various areas. According to the International Organization for Standardization (ISO) definition, usability is: "The effectiveness, efficiency, and satisfaction with which specified users achieve specified goals in particular environments" [1]. From this definition it follows that usability should always be considered in the specific domain or context, taking into account real users accomplishing their tasks under specific circumstances. Currently, dozens of definitions of the usability exist, each of them holding to a particular domain. In general, regardless of the area of interest, they all define usability as the ease with which users are able to use the system, shortly defined as the ease-of-use. The terms effectiveness, efficiency and satisfaction are also well defined in the ISO norms and we will come to the deeper details later in the chapter 2, when introducing usability evaluation.

With ubiquitous computing, we also have to think about usability in the context of everyday things such as microwave-ovens, telephones, personal hand-held diaries, ATMs, and all of the other information artifacts that interact with people. However, the main focus and the field of our interest is in the usability of software applications for the educational purposes.

Within the recent years, system designers and developers have not paid enough attention to the ease-of-use, mostly because other factors have clouded the importance of interaction between humans and computers – HCI (Human-Computer Interaction). The main focus has been on functionality, efficiency and the speed of the applications within cost and performance constraints. Research in the HCI area has been carried out primarily at academic institutions and experimental laboratories of major corporations. From there the discipline of HCI has evolved gradually and today it influences not only the whole area of computer science, but also stretches far beyond. HCI, like usability, has many definitions. The most common, famous and respected is: "Human-computer interaction is a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them" [2]. As clear as this definition is, it poses a wide area of concerns. HCI is an interdisciplinary area and joins many other scientific disciplines like cognitive psychology, linguistics, ergonomics, ethnology, social sciences and other relevant topics from the humanities and computer graphics, artificial intelligence, operating systems and many others from the computer science.

User Interface

Interface, as the wider term, is a boundary across which two independent systems meet and act on or communicate with each other. Giving the actual context of computer science to the previous definition, by the term user interface (UI) we mean the languages and devices in the interaction between a human operator and a computer.

Usability engineering

Having defined both HCI and usability, it is necessary to introduce usability engineering, which is still a very new discipline. Usability engineering (UE) is a design process, which brings together techniques, activities and tools leading to the delivery of a usable (easy-to-use) product. It is clear from this and previous definitions that it is necessary to involve users in the life cycle and this participation has to last during the whole development process of the system in order to create applications which fit for the intended use and are of added value to the intended users.

Usability engineering is based on the iterative process of the user interface development, and like many other engineering fields, this cycle consist of four stages: analysis, design, building and evaluation [7]. The iteration lasts until the evaluation yields the required results. There exist many other divisions of the UE life cycle, with varying levels of granularity, as found in for example [15]. During the analysis stage, users are in the center of the focus, while analysts gather data about them, their environment and processes needed to fulfill intended goals. The analysis produces requirements which are used during the whole system's life cycle. Also, the design stage is concerned with human factors. During the design, the first prototypes of the UI are released, including appearance, control, functionality and behavior. Later, during the development life cycle, the best solution is selected and refined. At the building stage, developers take up the requirements and prototypes and join them while implementing details into the system that enable more refinements and evaluation. This thesis is mainly involved in the evaluation phase of the UE life cycle, from the perspective of its characteristics, desirability and application within a certain context.

Usability evaluation

Usability evaluation as one part of the usability engineering process is an activity of reviewing the product, leading to the identification and assessment of the conformance between the system and usability requirements. In other words, using the previously given definition of usability, evaluation leads to determining how easy the system is to use and learn. This definition is one of the main corner-stones of this thesis.

1.2 Summary

In this chapter, the goals of this thesis have been established. The basic terminology concerning usability evaluation was briefly introduced. Usability (as the central concept of HCI), human-computer interaction, user interfaces and usability engineering are thus coupled in a tight and clear relation.

2 USABILITY EVALUATION

This chapter introduces the main rationales, methods and their classifications, and goals of usability evaluation, explaining thoroughly all of the related terms with the main stress on the feasibility of assessing the usability. Different approaches to usability as an evolving term are presented as prerequisites to usability evaluation. A pragmatic point of view is also given as a complement to the well-established approach.

2.1 Usability, omnipresent 'Alpha and Omega' of it all

Looking back into the past, we can see that attention to usability has not been taking either the primary or equal role in the development process with the other software engineering parts. The reasons for this diminished with extensive research in the HCI field (including a wider spectrum of sciences involved) and insight in computing in the everyday life (leading to a closer relationship between the user and designer). A significant change has also happened in the users population during the past decades. We can observe a gradual increase of using of computers between the wide society. Designers turned towards human factors and the development shifted from the system-oriented to the user-oriented, user-centered design (UCD). HCI then gradually evolved and became important during the last three decades of the twentieth century, tightly connected to the development of personal computers and their software. From that time on we could find the first real attempts towards ”user-friendly” approaches to the interaction with machines, and also the first models appeared (for example the KLM and GOMS models, User-centered design or definition of usability given by Bennett in 1983 [5], adopted in figure 1).

A step forward was the development of the HCI guidelines. These comprised the general design recommendations, display techniques and abilities, human abilities and other knowledge. Guidelines were developed on the basis of observations, laboratory research and experiences of researchers in order to accumulate knowledge. Although the guidelines might have been comprehensive [9], the main drawbacks of them were that they turned out to be difficult to apply, too tied to the technology currently available and they usually did not take into account the actual context. These problems showed that guidelines are not enough and as a side effect they brought more support, speed and effort to the development of other techniques and methods supported by the research in HCI. Subsequently, during the last decade, the technological development has brought an expansion to the use of computers and therefore a need for the evaluation of context specific areas such as multimedia, learning tools or the mobile Internet. It turned out that usability and thus usability engineering have to overcome the borders of experimental laboratories and become an integral part of the software engineering development process with the users as the focal point of interest. Research, technology improvements, experiments and case-studies provide the excellent support for the integration of usability engineering into the development activities.

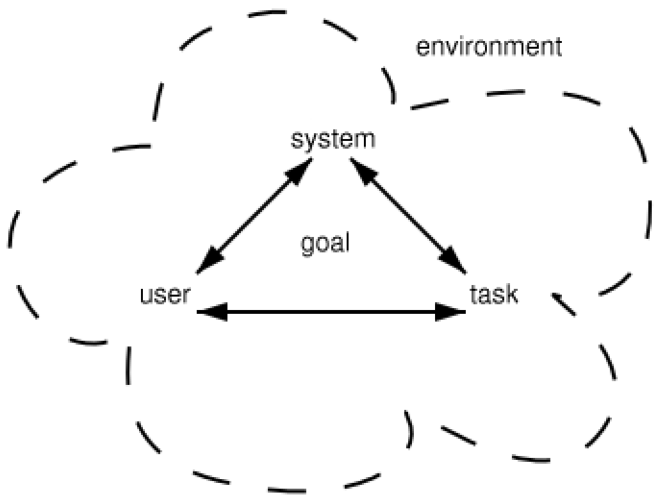

Figure 1 Usability as a goal

Although powerful, the definition of usability from the first chapter is not descriptive enough to conceive the overall concept which is commonly understood by everybody. While facing everyday interaction problems, we hardly think about effectiveness or satisfaction. In the contrary we are likely to be very quickly frustrated if our home appliance is not easy-to-use although we do not suffer any serious harm. A bit more serious loss, in terms of time and costs, arises from problems related to the usability of the working environment. And finally the real dangers impend, if usability problems occur in life-critical computer applications such (as a popularly given example) in a nuclear power plant's operation room.

Even if the system can be designed with respect to effectiveness and efficiency, a usability failure may lead to total dissatisfaction. The main purpose of these examples is to show the difference between a definition (ISO standards here) and the real world setting (contexts) and therefore emphasize that the characterization of usability is not equal to the objectives and specifications of the real-world design and the actual criteria which we could measure by the usability evaluation methods. To find a set of criteria as the input to usability evaluation, we have to unfold the prime definition of usability as a multi-dimensional term and establish its context. The first part of this quest is simplified directly by ISO 9241-11, which explains how usability can be specified and evaluated in terms of user performance and satisfaction. User performance is measured by the extent to which the intended goals of use are achieved (effectiveness) and the resources such as time, money or mental effort that have to be expended to achieve the intended goals (efficiency). Satisfaction is measured by the extent to which the user finds the use of the product acceptable [3]. The same reference then explains the intention of ISO to emphasize the relationship between usability and context of use. This is depicted in the figure 2.

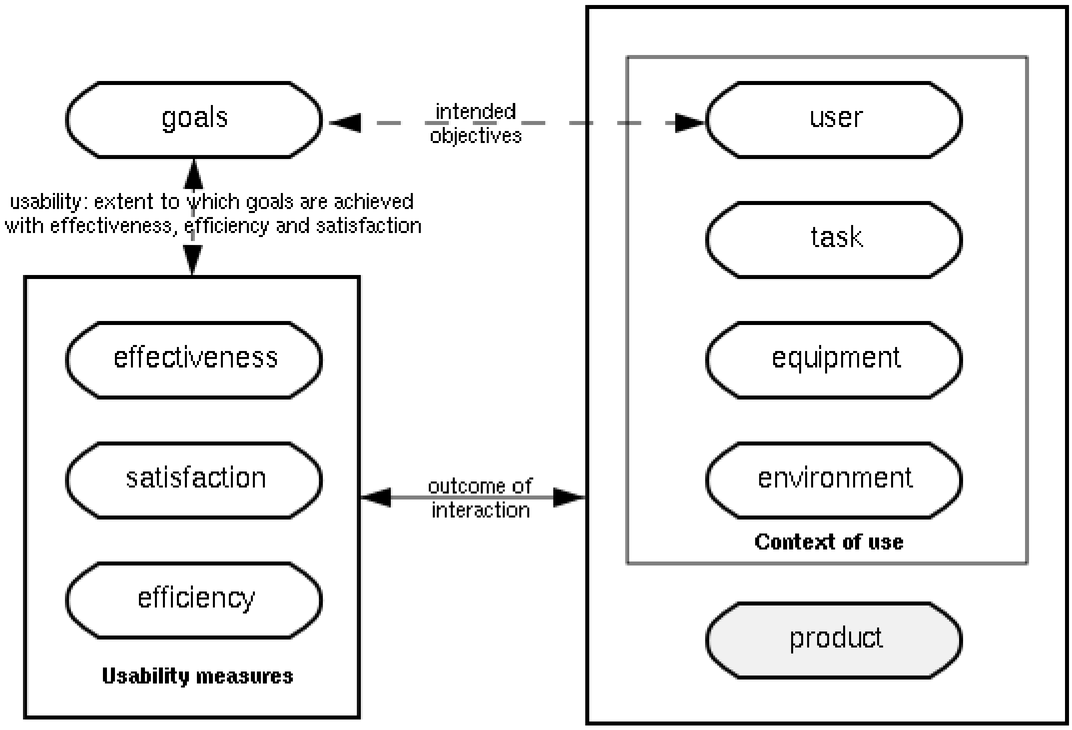

Figure 2 Usability framework, adopted from [3]

The context of use is formed by the user, task, equipment and environment, while each of them influences the usability of a system. All these characterizations, with an increasing dimension of usability, now more concretely defined, are giving a wider area for the derivation of method- and task specific criteria, which brings the possibility of measuring them.

Another definition is given by Nielsen [10], who does not present a strict and descriptive definition of usability, but he considers usability in the context of the overall acceptability of the system. Figure 3 shows Nielsen's view of the usability context. The overall acceptability here consists of two parts, where the first branch reflects socially oriented objectives (e. g. minimum pollution and safety risks, ease of disposal, confidence, ethical issues and others) of the system while the second is part concerned with practical acceptability issues. In the context of practical acceptability, usefulness is identified as the "issue of whether the system can be used

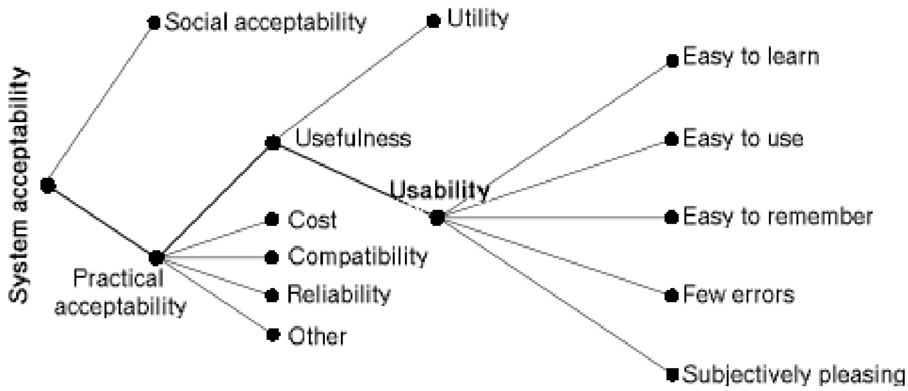

Figure 3 Context of usability, adopted from [10]

to achieve some desired goal", comprising of utility and, finally, usability. Utility is seen as whether the functionality of the system can do what is needed. Usability here is viewed as a key and a multidimensional property of practical acceptability and therefore an inseparable part of the overall system acceptability, the question of how well the users can use the functionality offered by the system. From the model it also follows, that usability is not directly connected to the functionality of the system but to the performance of it. According to Nielsen, usability consists of five attributes: learnability, efficiency, memorability, errors and satisfaction. Learnability or ease-of-learning means that a system should allow the users, who have never worked with it before, to rapidly start their working and accomplishing their tasks. Efficiency of use refers to the speed of accomplishment for users with acquired experience. Memorability as an usability aspect means that a system should be easy to remember, so that the user retains the information about how to work with the system after some period of not having used it and this retention is assisted by the system. Error frequency and severity is the fourth attribute of usability and corresponds with the error rate and ease of recovering from the error states of the system. Satisfaction refers to the pleasantness of the system, i. e. the subjective satisfaction of the user when using it.

To be complete with the definitions of usability, it is valuable to introduce Dumas' [11] perspective, which concentrates mostly on users. According him, “Usability means that the *people who use the product* can do so *quickly and easily* to accomplish *their own tasks*”; the points sustaining this definition are the focusing to the users, productive performance and users' decision about the product's ease-of-use.

2.2 Usability goals

Even when specified concisely, every evaluation of usability should have pragmatically selected goals, usually related to the context of application use. Typical questions posed by users and to which they would like to know the answers are: “Is this product generally easy to learn?” or “Is the system we are going to buy useful for our purposes?” or “Is the offered application better than the one currently used?” Since none of the dimensions previously defined in the context of usability is directly related to these general questions, and none of the usability evaluation methods are designed to directly address and answer these and similar questions, the user still has the right to ask them and obtain the sophisticated answers backed by the comprehensive underlying theories.

The selection of the evaluation goals directly drives the selection of the evaluation method. The more specific and narrow the goals are, the easier it is to select the proper method of evaluation. Therefore, the setting of the evaluation goals should happen at an early stage of the product development life cycle. As Dumas claims, the setting of quantitative usability goals puts usability into concrete terms and forces the design team to consider the product in terms of users' tasks and users' tolerance for time and effort [11]. On the other hand, coming from the previous definitions of usability, there also exist qualitative and pragmatic goals. For instance, a company implementing a product may require the application to educate the development team in the usability issues. Not everyone of methods which will be introduced in this thesis can fulfill such requirement. The high-level usability questions, general terms such as learnability or users' satisfaction, are often considered to be the most fundamental usability goals. Each of them can play differently eminent role considering the actual context of use. Thus, another important consideration around the setting of usability goals is the actual context of deployment. It is also necessary to add that some goals might be common to all techniques, for example to uncover interface problems.

In conclusion, while selecting the evaluation method, one has to take into account the context of use, it means the user tasks, user groups and the system used to run the product, and the purposes of the evaluation. Once the objectives are properly chosen, we can select the method, or a combination of the methods to address them all.

2.3 Usability evaluation methods

Usability evaluation methods (UEM) are used to evaluate the interaction between humans and the system in order to identify problems caused by inadequate design. Although there are many historical ways of identifying of UEMs, this thesis uses the following division: there are three types of usability evaluation methods in general: usability testing, usability inspection and usability inquiry. Beside this main division, modeling and simulation techniques form a separate group complementing the other methods.

Usability testing

Usability testing includes a wide range of methods, where the representative users are involved in the evaluation of the system or prototype. It is important to note, that it is the system which is going to be evaluated, not the user; and that the usability testing method is an artificial situation. The evaluation identifies the areas of a design that need refinements, aiming towards the improvements of the usability of a product. Usability testing methods are in literature sometimes referred to as empirical methods. Most of them are commonly carried out in the laboratories, under the defined circumstances as scientific experiments. However, usability testing has become more informal during the last years as there have been well established methods developed with high reliability and validity; and thus confidence in them has increased rapidly [11]. According to Dumas [11] (p. 26): "Usability testing is appropriate *iteratively* from predesign through early design, and throughout development."

For the purposes of usability testing many different sources of empirical data can be used to evaluate certain interface characteristics. This data is either qualitative or quantitative, the latter category can be roughly divided into relative and absolute data. The typical quantitative data gathered during the evaluation belongs usually to time-domain limits (for example the time-to-learn, the time to achieve first error, the reduction of errors in the time etc.) and error-domain data (the number of errors on typical task, the rate of errors achieved during menu navigation, the number of participants who have the same error). Qualitative data is gathered mostly by questionnaires and verbal protocols. Typical qualitative data might be for example the level of support in a high-risk environment. Each of the data gathered during the course of the evaluation has different importance of contribution, as reported by Ebling and John [8]. Their paper backs the claim that in a limited resources environment (which is very often the case), “collecting performance and questionnaire data should be sufficient” and “researchers assessing the predictive power of an analytic technique might be able to justify collecting only verbal protocol data as evidence for usability problems.” This study shows how important it is to collect multiple forms of data in the course of usability testing.

The main ideas underpinning the usability testing methods are that the test is conducted under well established conditions, variables, goals and procedures in order to ensure the scientific course of the evaluation and later analysis. The inseparable part of every usability testing method should be the debriefing meeting with the participants. Since the term usability testing corresponds to a broad area of usability evaluation methods, here we introduce only a few of the main approaches.

The simple **performance measurement** technique serves to obtain quantitative data about participants' performance when they perform the task during the evaluation test. This data can be used as an input for different kinds of comparative tests. Dumas [11] claims that: ”A typical usability test now includes six to twelve participants in two to three subgroups.” This would ensure both statistically significant data and the coverage of the usability problems revealed.

Think-aloud (or thinking-aloud) techniques comprise of different kinds of methods, encouraging participants to say aloud what they are thinking about while they are performing the actual task. By this, an usability expert can uncover moments of a confusion, preconceptions or errors. Think-aloud methods have many strengths, particularly the wealth of qualitative data collected and comments made by users. On the other hand, think-aloud methods are unnatural to the user because it is difficult to force the users to talk about their cognitive process.

Co-discovery is a technique in which two (or more) participants work together to perform the tasks while talking to each other. The method is similar to the think-aloud technique, but it is supported by the opinion that the dialog is more natural than thinking aloud alone and the observation of the interaction between users brings out more insight to the interaction.

Active intervention is a method in which an evaluator sits in the same room with the participant working on the evaluated product and asks direct questions to probe the participant's understanding of the product. There is a significant difference between this method and interviews, since the latter is conducted after the test, usually with distance to the system. The probing questions should be not biased and should not drive the participant's performance.

As long as the previous methods are affected by the barrier between participants and evaluators and by the overall evaluation setting, there were also developed other, indirect testing methods, wherein the evaluator is separated in the time or (and) space from the participant. These methods are called **remote evaluation**s and often are mediated by a network [12].

Usability inspection

Usability inspection is a category of UEM, consisting of numerous approaches, having evaluators inspecting the interface as an unifying characteristic of usability aspects. The group of evaluators consists mainly of usability specialists but it is also common that software developers, users and other professionals take part in the examination process. With a few exceptions most of the methods of the usability inspection are informal, based on the rules of thumb, general skills, knowledge and the experience of evaluators.

In the situations where it is difficult to recruit and involve the real users into the evaluation (proposed to be a usability test) or the project is running on a tight resources, the usability inspection method is often selected to be carried out. Since many inspection methods aim also to inspect the UI specifications, they can be used during the early stages of the system development life-cycle [6].

In the **Heuristic evaluation**(s) (HE) method, a few evaluators (usually and preferably the usability experts) evaluate the system design by judging its compliance with a small set of established rules (usability principles), called heuristics. HE, probably the most popular type of inspection used nowadays, was originally brought up by Nielsen [10] together with a set of ten basic heuristics (listed in appendix A), which guide the evaluators during the assessment process. This set is not unique and can be individually adopted. Any usability problem found is evaluated for its severity and extent. Nielsen proposes using three to five evaluators for the heuristic evaluation; each evaluator works separately and goes through the system at least three times. Among the biggest advantages of HE we include that it is a low cost method, well documented, easy to learn and intuitive to perform. It can be applied at a arbitrary stage of the

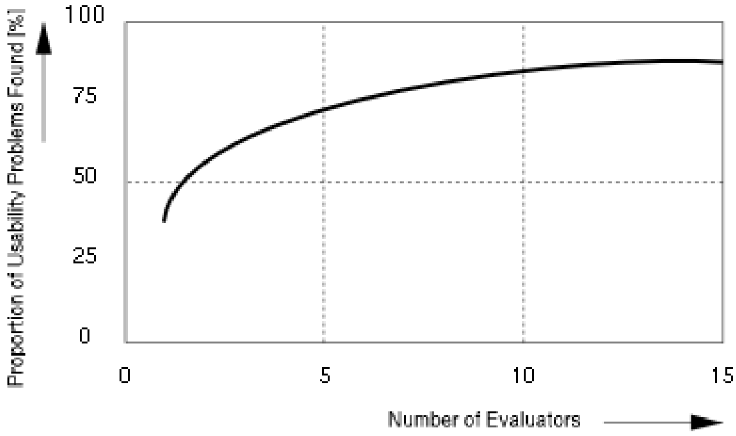

Figure 4 Usability problems found by HE as a function of the number of evaluators. Adopted from [10]

development life-cycle. HE has, though, also few drawbacks, for example it does not offer solutions to the problems identified and the original heuristics delivered by Nielsen are considered to not have enough explanatory power; there is also the need for an usability expert. There should be more evaluators involved in the HE, as seen from the figure 4. From there we see that to cover at least 75 percent of possible problems we need to involve around five evaluators in the evaluation process. Adding more evaluators does not increase an effectiveness.

An important factor here is also the level of experience of the evaluator. According to Nielsen, novice users are poor evaluators, whereas HCI experts are about twice as good, and domain and HCI experts are almost three times better to uncover usability problems than novice users.

Cognitive walkthrough (CW) is a general name for a set of methods, similar to code walkthroughs and based on exploration, where evaluators imagine the users executing a set of real representative tasks using the system under evaluation, step-by-step. The motivation of the cognitive walkthrough comes from the view, that many people prefer to learn some system by goal driven exploration. The walkthroughs are performed by the designer or an expert in cognitive psychology; hence it is the evaluator who tries to act as a user. As an input to these methods we include the detailed description of the user population, interface design, set of the tasks and sequences to complete them. Then for every part of each sequence a credible story is established and evaluators perform the actions and assess whether the user would be able to successfully complete the task with the given conditions and environment. The course of CW is depicted in the figure 5. Since cognitive walkthroughs focus mainly on the attribute of the ease of learning, using only these methods for the evaluation can drive the system design in the

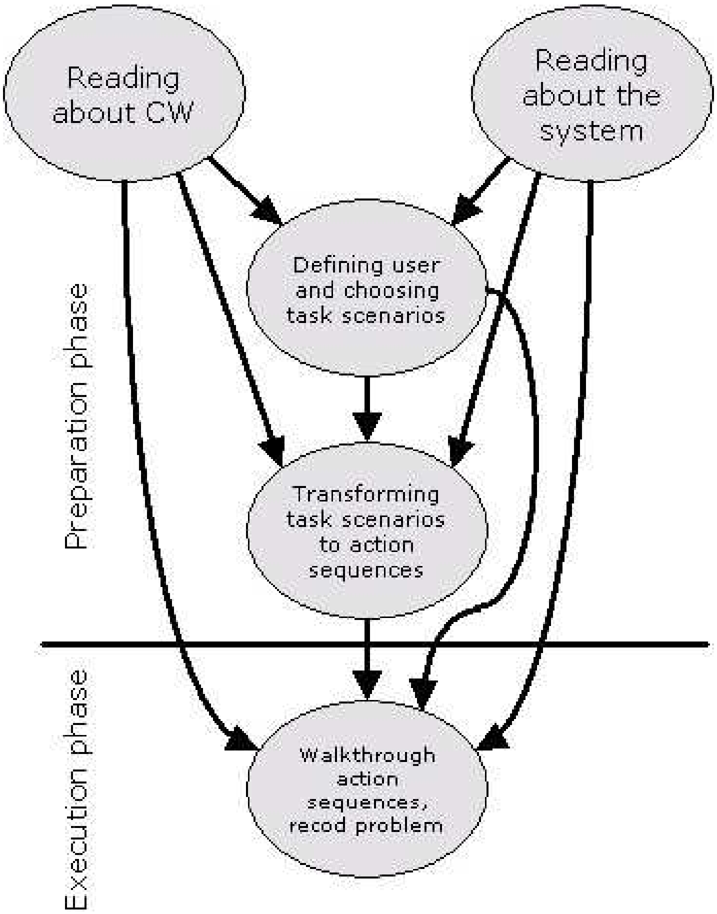

Figure 5 The stages in a CW and dependencies between them. Adopted from [13]

corresponding direction. The CW methods uncover the differences between the designer's and user's conceptualization of the system and considering their capabilities, they are highly applicable and eligible to use during the early stages of the system development but can be used anywhere at any stage of the development cycle. However, CW requires some degree of knowledge of psychological theory and terminology and it can be time consuming too [4, 13].

Since the first version of CW has been presented in a proceeding paper of ACM CHI 90' conference [42], CW is still in the focus of the active research. The actual topics based on CW and the related theory concern the cognitive walkthroughs for the Web (CWW), adopting the original ideas standing behind the CW for a better fit to up-to-date technology [14]. From here we can see that certain efforts exist to transform the methods which have already proved their usefulness into the new, context sensitive methods using the original ideas, in this case the World Wide Web environment. As Blackmon claims in their paper [14] : ”The CWW overcomes a serious limitation of the original Cognitive Walkthrough.”

A pluralistic walkthrough is carried out during the early stages of the system development, when the user, developer and usability expert groups meet together and go through the task scenario and discuss and evaluate the usability of the system represented by the paper prototypes. Because these meetings are attended by the diverse kinds of people with various skills, knowledge, experience and perspectives involved in the project, they tend to gain the wealth of opinions and cover most of the usability problems. Pluralistic walkthroughs are coordinated by the human-factors specialist; all of the other participants are asked to act as the potential users of the developed system. The evaluation is usually carried out on the paper prototypes of the interface as most of the walkthroughs; it means that the pluralistic walkthroughs are appropriate to be used in the early stages of the development life cycle.

The formal usability inspection method was designed to support the designers (without the knowledge of the usability issues) to review a product in order to find a large number of defects. Formal inspection methods are performed through a six step process with strictly defined roles for each participants: the planning, a kickoff meeting, a preparation phase where inspectors review the interface individually, the main inspection review when the inspectors' lists of the usability problems are merged, and a follow-up phase where the effectiveness of the inspection process itself is assessed [17]. The course of the formal usability inspection has also an additional effect; it also educates the designers in the field of the usability evaluation. The inspections can be held in the early stages of the development process, thus enabling more problems to be fixed sooner without the need for reimplementation. Concerning the properties of the formal inspection, it is highly feasible for industrial software development environments.

Feature inspection lists the sequence of features used to accomplish the typical tasks, checks for long sequences, cumbersome steps, steps that would not be natural for users to try and steps that require extensive knowledge or experience in order to assess a proposed feature set [6]. Thus each feature is evaluated with the focus on usefulness [17]. Recollect that according to Nielsen, usefulness is the more general aspect of the system acceptability than usability. Feature inspection is best used when the functions of the system are already established and known so their actual functionality can be compared against the specifications.

Consistency inspection is carried out at a meeting by the designers of the multiple systems, with step-wise assessment of the differences between the actual interface implementation and their own designs. Thus the aim of these inspections is to produce the maximal consistency throughout all of the components of the system [17]. Because of the special focus and attendance of acquainted parties these meetings uncover dozens of inconsistencies. **Standards inspection** is the simple method consisting in the assessing the compliance of features of the actual UI against the predefined standards. Standards inspection is carried out by usability experts.

The Guidelines and checklists approaches are used to support the evaluation experts while practicing a certain usability (inspection) method, giving them the basic scope wherein to perform the actual evaluation of the system. Guidelines address those attributes of a product, which have been shown to improve the usability; following the guidelines should lead the developers in the design of the system to conform to these guidelines and therefore avoid the usability problems. Checklists are meant to be used after development, but before the deployment; a system is inspected to check its compliance with the checklist items. As mentioned before, checklists have originally had many serious drawbacks. While trying to capture the contextual sensitivity, checklists become too extensive, which causes discomfort and less efficiency because they are slow to apply. Other complications come while considering the maturity of evaluators. Experienced experts are able to relate general checklist questions to the specific system and its domain, while novice users or real end-users might have some problems in relating general questions to a specific application.

Perspective-based inspection (sometimes referred to as a scenario-based checklist inspection method) techniques take the results of the inspection methods research, integrate them and focus on the system design from three defined perspectives: two levels of users' experience: the novice and expert user, and finally from the perspective of error-handling [18]. Perspectives are used to focus the inspectors' attention on a specific subset of usability issues. This focus should ensure a higher percentage of detection of the problems related to the certain perspectives used and thus the combination of different perspectives can cover a wider area of problems uncovered than the simple iteration of one (general inspection) method with (usually) one perspective in the focus. Moreover, the task scenarios are used to ensure the relation of the method to the users, the inspection method is well-defined and the criteria are tailored to fit into the user domain. Comparing the perspective-based inspection with the heuristic evaluation in finding usability problems in a web-based application, the first method is about 30 percent more successful [18]. From this it implies that assigning inspectors to a more narrow responsibility (perspective) leads to higher performance on usability evaluations.

Usability inquiry

In usability inquiry methods, usability professionals gather the information about users' opinions, likes, dislikes, needs and understanding of the actual system under evaluation by interviewing them and observing them during their real work, preferably at their working place. Among usability inquiry methods belong field observations, contextual inquiry, interviews, surveys, questionnaires, journaled sessions, logging and screen snapshots. All these methods have in common the stress on observing the users in their working environment and many of them can be done automatically and remotely as well.

Observation techniques are considered to be the main and most effective usability inquiry method. An observation is carried out directly in the field and does not remove the users and the product they work on from the actual context, as is usually done by usability testing methods. It is one of the simplest and cheapest methods of evaluation, with the emphasis in avoiding any interference with the users' work. Still, it is a very useful method, regarding that on the basis of observations that some word processors have been improved to include the templates as a special file category [10]. Observation may be carried out at any stage of the development life cycle, including deployment.

Logging the actual use involves having the computer automatically collect the statistics about the detailed use of the system. From the gathered data we can find useful information how the users actually perform their work. The typical information collected by logging contains for instance the frequency with which each user invokes a certain function of the system or the rate of error situations. If the frequency of using some command which is necessary to fulfill the task is low, it might indicate a significant cause of problems or be the reason for removing feature out of the system. It is also possible to log whole transcripts of user sessions to uncover additional characteristics of the use.

The main strength of the logging method lies in the possible integration of many sources of data to obtain a statistically significant dataset about the use of system. Since we are not limited to one working place and one user, the coverage of revealed problems is possibly quite high. Logging is normally used as a way to collect data from the real working environment of the system after a deployment, but can also be used as a supplementary method during user testing at the early stages of the software development life cycle. From the nature of data collected by logging, it is difficult to measure user satisfaction with the system as the data shows what operations the user did perform but it does not show what operations were not performed at all.

Interviews are very common, informal techniques to obtain data of particular interest from the users. From the nature of the interview follows that the interviewer should be one of the development team members, preferably the usability expert and the issues discussed are related to the users' subjective satisfaction and their opinions. Interviews can be either open or close, wherein the latter form the evaluator keeps more narrow area of interview. Interviews can be carried out during any phase of the development life cycle.

As a similar approach to interviews (in the terms of the data of particular interest) are **questionnaires**, with the differences that they do not involve having an interviewer in the actual evaluation process and there is no interviewer to aid the user with an evaluation. Traditionally we recognize two forms of the questions, regarding the form of answers. These are either open-ended (free-form) or closed, where the answer or a way to answer is predefined. The latter form is usually preferred as long as users do not bother to write in natural language. Even though questionnaires are flexible and easy to use, the biggest drawback of these methods is the low response rate, ranging from about thirty percent if no incentive (compensation or reward) is offered. Another issue to be considered while conducting the questionnaire-based inquiry is the design of a questionnaire itself, being brief, concise, impersonal, valid and reliable.

Focus groups are a kind of informal technique, where multiple users (usually from six to nine [10]) participate in a discussion session in order to address their needs and requirements. This technique can be run both at the early stages (during the design) and also after deployment. A moderator maintains the focus of the discussion using a preplanned script trying to boost up users' spontaneous reactions and ideas without inhibiting the free flow of the session. This requires a certain kind of experience from the moderator. The results of the method are also influenced by the experience level of the users, since user groups differentiate in their needs. Current real approximations of the focus groups approach are on-line conference forums with certain issues in interest. According to the Nielsen's opinion, conference subscribers are often recruited from the above-average involved users, which brings biased results to the survey.

An inseparable part of the usability inquiry methods set is **user feedback**. This method is inexpensive, simple and does not require the time of an usability expert in contact with users. Apart from that, user feedback has more advantages. It is initiated by the users, reflecting their immediate reflections, needs and opinions and it is not limited to a certain period of data collection after deployment, which can also be considered as a drawback. It has to be also supposed that user feedback returns mostly dissatisfied users' answers which does not fully represent the user population. Therefore user feedback should not be used as the only method for usability evaluation but rather as a complementary method.

Modeling and simulation approaches

Modeling techniques accompany HCI research and usability evaluation from the very beginning. Most of them are the results of active research held in the early eighties of the twentieth century. The purpose of these models was to bring engineering models to the HCI field. During the following years, these methods have been improved up to today's maturity, in order to decrease the time and cost expenses and to ease their application, since system developers were not trained in psychological issues. The use of performance modeling should bring quantitative prediction and approximation of the user performance (for example the execution time or the time of error occurrence), therefore these models are proposed to be used in the very early stages of the system development life cycle. However, one can also model the users' knowledge, task environment and user interface as a part of the system [21].

Simulation approaches complement the classical evaluation methods similar to the modeling techniques, in the sense that they are a good source of quantitative data without involving real users in the evaluation. A simulation can be run as many times as a required amount of data is achieved, enabling the evaluator to drive this generation by different parameters.

The GOMS model, dating back to 1983, is one of the most widely known and validated theoretical approaches. The GOMS acronym is created from the components of the model: **G**oals, **O**perators, **M**ethods and **S**election rules. Goals are users' objectives, describing what the users want to achieve and can be further divided into the sets of subgoals. As the example of the high-level goal we can use user's intention t*o write a letter,* while this goal is divided into smaller sequences of subgoals as *open the editor, create a new file, write an address* and so on.

Operators are the basic actions the users must perform in order to fulfill their goals and the actions that the system offers users to perform. In the context of the graphical UI the operator is for instance the menu selection or other direct manipulation action. The definition of operators can be done at different levels of abstraction according to the granularity of goals but many GOMS models define them at a concrete level [4, 20].

Methods are the given or learned sequences of sub-goals and operators used to accomplish the goals. It may happen that there exist more methods to accomplish one goal and the user can then select between them. For this purpose we need to define the selection rules which are personal intentions of selection from the various methods to achieve a given goal. Thus, goals, operators, methods and selection rules represent together one's personal knowledge required to perform a task.

The GOMS as a concept stands behind many other models, for example the NGOMSL (a more natural variant of GOMS), KLM (Keystroke-Level Model, simplified GOMS) or CTT (Cognitive Complexity Theory) models are based on it. GOMS has a few considerable drawbacks: it defines its domain in the expert's performance context (which goes opposite while concentrating the focus on end-users) and with increasing complexity and variety of tasks the design and analysis of models get difficult, with very fine grain.

The Information scent modeling [22] method originally served to generate and capture the hypothetical users' interaction with a web site by traversing the links on the pages. These hypothetical users (here called agents) have some information goals which are compared to the content of the actual page and until the goal is found in the page or after specified effort (in terms of time or links visited) the traversing (browsing) continues on the basis of stochastic decisions. During the simulation, all of the steps of all of the agents are recorded the later analysis.

Petri Nets (PN) and their successor Colored Petri Nets (CPN) are mathematical and graphical modeling tools, which are widely used in the modeling and simulation of various systems, providing the graphical and formal representation that makes the design and analysis easier. The PN approach has been applied for example as a model of an Web browser's interface or in the context of industrial systems [23], mostly to analyze the navigational structure of interfaces. Though there exist many tools that facilitate the modeling and analysis of PN and CPN, using this approach for the usability evaluation poses certain needs to the evaluators' skills. PN based simulation can be used during the whole system development life cycle, while the hierarchical model of interaction is built.

2.4 Comparison of usability evaluation methods

As long as the UEM are assessing the usability of the system, their quality requirements are close to the usability of the system and we can demand effectiveness, efficiency and satisfaction with them. By these we could compare the overall usability of UEM. Although many comparison studies have been carried out, none of them covers the whole area of the usability evaluation methods.

As far as I know, a comprehensive comparative study including the whole set of usability evaluation methods does not exist, only partial comparisons have been done. These studies usually compare the ability of methods with respect to the number of usability problems revealed and cost-effectiveness, which are considered as the most important characteristics of an evaluation method. This is a consequence of the different requirements of methods belonging to different categories, testing, inspection and inquiry. For instance, Doubleday [19] compares Heuristic evaluations (with five HCI experts) against end-user testing (pre-study, observation and debriefing with twenty participants); participants are evaluating the graphical tool used to support the information retrieval, resulting in the conclusion that the HE is less time consuming, assessing more than twice the number of problems (although not including all of the problems found by the user test), but it has some lacks in addressing the end users' task-based problems.

Except for the cost-effectiveness and number of problems identified, Karat in chapter 8 of Usability Inspection Methods [17] includes more issues while comparing usability testing and inspection methods. The comparison is done also with respect to usability objectives, reliability of findings, human factors, facilitation of the organizational acceptance of usability issues, timing issues and problem fixing. However, issues such as learnability or context-sensitivity are missing. Results from this paper (although covering the research mainly from the beginning of the last decade of the twentieth century) state that usability testing covers a wider range of evaluation objectives, identifies more usability problems, focuses more on the human factors, may be used earlier while "it can provide high-level design guidance early in the development cycle" [17, pp. 217], and provides more design improvement recommendations. From the two previous citations we can see a contradiction. Inspection methods do not score in the latter paper very well, which I conclude to be mostly due to the fallout of paper publishing dates and the overall focus. The conclusion drawn from the comparisons is that the best practice of usability evaluation should be based on the combination of usability testing with the usability inspection method, although some areas of overlap may occur.

One important aspect of every usability evaluation is the need for evaluator attendance during the evaluation process; for example during the whole course of heuristic evaluation the evaluator is needed. For the proper analysis of results, it is always necessary to involve an expert after the evaluation. However, some methods do not require the direct attendance of the evaluator at the actual evaluation process. This may be considered as an advantage when choosing evaluation method for systems already in use or running on the low budget.

What is most evident while analyzing the properties of methods listed in this thesis and considering certain working environments, if not adopted to this special context or not adopted well or the usability goals are not established properly, certain usability evaluation methods usually capture only a subset of problems offered in the interaction or even fall short altogether. To overcome this weakness it is recommended to combine more evaluation techniques together, which should yield more comprehensive coverage of possible problems and their symptoms and, even better, reveal the causes of the symptoms. While evaluating already deployed systems in order to enable the comparison of the level of usability between different products, beside the context of use we also have to consider other aspects, like the maturity of evaluators or the cost of evaluation. The following table overviews the methods listed in this thesis.

Method category Method name	Expert/ moderator needed	Appropriate phase[1]					Main advantage / disadvantage
		A	D	B	E	D	
Usability Testing							
Performance measurement	NO		•	•	•	•	Comparable results/Don't find problems
Think-aloud	YES		•	•	•	•	Uncovers users' problems/ Unnatural
Co-discovery	YES		•	•	•		Dialog-based / Unnatural
Active intervention	YES		•	•	•		
Remote evaluation	NO		•	•	•	•	
Usability Inspection							
Heuristic Evaluation	YES		•	•	•	•	Cheap, intuitive/ Provides no solution
Cognitive walkthroughs	YES		•	•	•	•	Theory based,task related/Expert needed
Pluralistic walkthroughs	NO / YES		•				High coverage /
Formal usability inspection	NO / YES		•	•	•		
Feature inspection	YES			•	•	•	
Consistency inspection	NO			•	•		Focused / Low extent
Standards inspection	YES			•	•		
Guidelines & checklists	YES			•	•	•	Low costs / Difficult for novices
Perspective-based evaluation	YES		•	•	•	•	User centered /
Usability Inquiry							
Observation	YES				•	•	Context sensitive /
Logging	NO					•	Integrate vast sources /
Interviews	YES	•	•	•	•	•	Flexible / Subjective, time consuming
Questionnaires	NO				•	•	Subjective / Low return rate
Focus groups	NO / YES	•	•		•	•	
User feedback	NO					•	Unlimited period/ Usually angry users
Modeling and simulation							
GOMS	YES	•	•				Validated / Experts' domain
Information scent	YES	•	•				
Petri Net	YES	•	•				Source of data/ Requires skills

1 With **A**nalysis, **D**esign, **B**uilding, **E**valuation, **D**eployment phases

Table 1 Summary of introduced evaluation methods

2.5 Summary

In this chapter most of the important usability methods have been introduced, together with their main characteristics. One of the main points we learned from this chapter was the importance of context-sensitivity. Usability has been shown as only measurable in specific contexts of use. The main advantages and drawbacks of the various usability evaluation methods have been presented. It has been also shown that not each method is feasible at every development life cycle phase together with the influence of the differences of different evaluators' levels of experience needed which limits the feasibility. Evidence has been given to support the combination of different usability evaluation methods, although usability testing and inspections have some overlapping areas.

3 TECHNOLOGICAL FACTORS OF EVALUATION

This chapter considers the technological aspects of systems that need to be inevitably taken into account while evaluating actual environments. Evaluation goals here are seen from the practical acceptability perspective of the system.

3.1 How a technology influences the tool selection criteria

It is difficult to assess the influence of a technology to usability as long as no scientific discipline exists involved in this problem. Ergonomics, in the classical sense, only addresses the influence of environment-specific issues to the one's work in order to achieve better adaptation to one's characteristics. All natural sciences conclude technological criteria as a part of the requirements, while they try not to depend on the technology state-of-art. However, technology has a clear impact on the theory and the theory then shapes the technology by practical improvements.

Outside this cycle, technology and theory also influence humans and their environments (and other technical systems). This influence has two sides, technology supports humans but also poses certain constraints. These constraints may negatively affect human performance, in certain environments. Therefore, the people responsible in the selection of the tools have to take into account how well the tool will fit into the actual environment. Starting with this general question, one comes down to the more detailed ones in the same environment. Although using the same technologies as single-user software or commercial tools, educational tools differ greatly in their requirements and configurations.

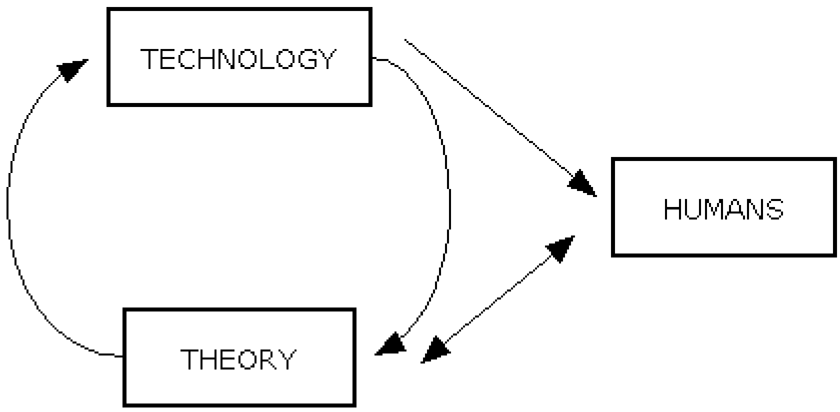

Figure 6 Influences of technology

3.2 Computer as a tool

Computer technology is still rapidly developing. User interfaces have already broken out the boundaries of the desktop computers; ubiquitous computing brings more interaction possibilities with computer-backed interfaces to our lives. On the other hand, it also brings more challenges to assessing this interaction. The library we used to visit twice a week may turn to a digital form, the university can start to refer to itself as a virtual educational institute, we can hardly predict what kind of functionality will be offered from a new vacuum cleaner we are going to buy next year. All of these and similar artifacts and interfaces, although possibly still hypothetical results of pervasive computing, need to be usable and useful; both designers and users want them to fulfill their expectations and goals.

The other issue we face nowadays, brought by the information-society age, is a change in the way we communicate with each other. Personal digital assistants (PDA), mobile phones, messengers and other tools contribute to the enlarging communication possibilities. Currently, computers are also widely used as supportive tools within various, often overlapping disciplines. We were recently overwhelmed with acronyms such as CSCW, CSCL, CASE, CAI, CAD and many others. All these have in common the computer as a mediator and facilitator of the interaction between the task and user or group of users. Most of these systems also followed similar scenarios as the singleware development history; first the functionality has been in the focal point and later the usability problems started to be in the center of interest.

The **World Wide Web** (WWW) has extended to unexpected dimensions, starting from an idea of hypertext supported by the military defense network in 1968, up to today coming to the displays of our handheld devices. It can be said, that the Web as a tool has most users amongst all with the most rapid growth of use; the estimation survey made recently showed 580.78 million users on-line [24] which counts for 9.57% of the global population. Similarly as with the beginning of computer applications development, first the designers focused more on the content than the look (aesthetics) and usability. With increasing gravity of the electronic communication, accessible technology and with commercialization, the importance has been laid also to the graphical layout and ease-of-use of Internet sites; developers of portals learned that users will return to the pages which satisfy usability requirements. Thenceforth many usability evaluation methods have been adopted and developed for the context of the WWW and many standards have been established.

Multimedia should not be regarded as a tool in the narrow sense of the term but as a supporting framework for many computer-based tools. We understand the term multimedia as an integration of multiple output media in a single application. The purposes of multimedia applications differ, which brings difficulties to the usability evaluation. They might be found to contradict if the guidelines direct the designer to implement certain functional elements in a specific way in the entertainment multimedia tool and in the art multimedia database, for instance the graphical appearance of dialog boxes or a way of presenting the information. Thus, in the multimedia applications especially the focus on aesthetics is stressed. Also, navigation in non-textual environment is a novel approach that has not been considered before.

Another important factor concerning multimedia is the common multimodality of interaction. In the past, only vision has been involved in interacting with computers, which has recently changed in the current multimodal interfaces, including hearing and haptic channels (the smell and taste channels are still rarely used). Originally developed usability evaluation methods do not take these issues into the account. However, cognitive issues are considered as a basis for the guidelines approach [28]. The Web and multimedia development and their integration allowed the launching of many of the services facilitating information exchange and sharing, communication and conferencing, message services, distance learning and so forth.

CSCW stands for Computer Supported Cooperative Work, the tools used in collaborative work environments, also know as groupware. Typical applications of CSCW are email, videoconferencing, but also multi-player games, all the applications having the users cooperating in the real time. CSCW goes alongside technology and Internet development, which is also concerned with the evaluation methods of groupware tools. As Steves in their paper [26] claims, there exist basically two approaches to the evaluation of the usability of groupware which were studied to the detail.

The first way is through the studies of real collaborators (users) in their real working environment, while the second approach prefers usability inspection methods. The decision of which of these two approaches to use is difficult, as long as their comparison is not clear and one can hardly interchange each other in the usage. User-centered testing yields detailed information about the work situation and is able to assess the groupware tool in a particular scenario. While trying to contextualize inspection techniques there has to exist a mediator responsible for that duty, in the methods which do not have the ability to adapt to specific contexts. Inspection techniques, on the other hand, are generally less resource consuming.

These findings are already known from singleware evaluation, though the novel asset is that inspection methods can be usefully applied in the groupware tools evaluation. For instance heuristic evaluations [25] and cognitive walkthroughs have been adopted [27]; heuristics were given meaning in the context of teamware requirements, their power being fixed points from where the evaluator can start is exploited. Originally cognitive walkthroughs do not account for multiple users and dynamical group work, though the groupware walkthrough does it all. Pinelle suggests to combine groupware walkthrough and groupware heuristic evaluation in order to test multi-user systems [27]. However, their method cannot address social and organizational issues arising from the real context of multi-user systems.

By the term Computer Assisted Learning (**CAL**) we understand the information and communication technology used to support the learning and knowledge exchange. CAL includes both students and teachers using computers in all study tasks; this means the preparation for a lesson, active interaction mediated by computers, assessing the results, administration and so forth. Kopponen in her dissertation argues, that “built-in educational properties are the essence of CAI (*computer aided instruction)* applications.” [35] Two of the various CAL approaches can be shown : recently, because of the widespreading Internet, the CAL group has been enlarged by the Internet based learning environments (IBL), which is a considerable part of educational software nowadays. The embedding of cognitive technologies into the CAL led to the development of intelligent tutoring systems (ITS). The purpose of ITS is to create a model of the knowledge and guide the learners through the learning process in order to fill the gaps between their knowledge and the model implemented in the system.

CSCL is the acronym for Computer Supported Collaborative Learning, and has a few similarities with CSCW as a concept, though CSCW architectures mainly belong to the commercial sector. CSCL is considered as a part of the CAL tools group. Except for collaborative learning, CSCL tools are also dedicated to facilitate the building, sharing and negotiation of group knowledge in a school environment. The main idea here is that knowledge is built up through group investigation and conversation. The tool should support the construction of arising artifacts within the learning environment, with a possibility to share and access them from every virtual learning place. Once the artifact is built up (for instance the collection of links to Web sites, multimedia objects as pictures, texts or sounds), it becomes the part of the groups' knowledge.

3.3 Practical aspects of system acceptability

While developing usability evaluation methods, one has to necessarily think about the target systems characteristics and capabilities, which also includes the technological aspects of the evaluated products. Given the end-user population, every evaluation method has to take into the account users' goals, tasks, the context of their work, and the technology they are running. But there exist other issues to be considered while thinking about practical usability aspects. From Nielsen's concept of usability depicted in figure 3, we simply include costs, compatibility issues, reliability of the system and other low-level questions concerning the practical acceptability. While concentrating on the extraction of the technical concerns from the the ISO usability definition one finds the term *context of use* as conformed by the environment, equipment, task and users, where all four parts directly influence the practical, hereafter technological aspects of the intended system.

In a real-world setting, usually all of properties of usability are not independent, which also has to be taken into account. In the terms of the task users perform we include the description of the interaction between the user and technological resources as long as the tasks are performed on the actual system. The equipment and environment include the description of physical and social issues influencing usability. Such physical conditions are software, hardware and working conditions (furniture, ambient environment and so on). The environment also consists of the social background such as cultural habits, work practices and policy, privacy and safety, administration and other relevant issues. Users' capabilities influence aspects of the system in two ways. Cognitive issues are covered by the usability part of the evaluation scheme, while physical characteristics influence the accessibility of the system for the certain group of users. It means that in a particular application modes should be offered for different user groups, including disabled people, users of various ages and so forth.

Usability tests are mainly conducted in well-controlled environments, most often in laboratories, which enables more control over the course of evaluation. This control implicitly includes the definition of the environment and equipment conditions. However, with more loosely controlled conditions, e.g. when moving the usability evaluation out from the laboratories and addressing more informal inspection and inquiry methods, the variability of working conditions grows; this poses a challenge on a certain method's adaptability to the changing environment or, if possible, independence to the changing context. This is difficult to satisfy and goes opposite to the requirement for the method to be of general use.

The following list summarizes issues needed to be included while considering practical issues of usability within system acceptability. The grouping of the topics is done on the basis of their individual relativity according to users, tasks, environment and equipment, it means that the context of usability from the perspective of ISO [1] is used. However, some topics are overlapping and it is not clear which category they belong to, for instance in the term of reliability we include both effects of the system to the users and also to the environment. As with every usability checklist, one has to think about it as subject to a continual change, because the rationales underpinning the issues are changing in time with the development of technology and research.

Availability and Compatibility

Into the availability and compatibility group of a practical acceptability we include such factors which represent possible influences of equipment to the users, their performance and therefore they play an important role in the selection of a tool.

Software Compatibility

An institution intending to purchase any kind of new software applications has to be aware of the current system's software equipment and consider this in the terms of compatibility between each other. It would be wasting of resources if the new system is not compatible with the other current tools, for example in the sense of sharing files between different applications. It is common that modern tools enable exporting and importing files to and from the different versions of the application or even to and from the applications from different vendors. Although obvious, the dependence between the applications and operating systems they can be run on is another important factor.

Hardware Compatibility and availability

Similarly as with software compatibility, there exist implicit requirements for the hardware. The new system has to be compatible with the current hardware equipment. In addition, the hardware requirements, in terms of resources needed to run the product, have to be fulfilled.

The cases of incomplete equipment clearly influence the possibility of using the tool. Also, if the tool has demanding requirements for the environment which are not easy to fulfill, it limits the usage.

Accessibility

Users will always drive usability issues. Every usability evaluation method has to consider the user population, which means different user groups from the perspective of users' characteristics and limitations. Accessibility means that every *user can directly use* the system without any further modification to the system and the system has to *offer various customizations* to meet a specific groups' needs. Cognitive aspects of the interaction are mainly covered by usability evaluation methods introduced before. Practical acceptability aspects of the system influenced by the user population include mainly the accessibility questions, as for instance different age groups. A considerable part of the population also has some kind of impairment.

Support for disabled

Possibly a wide group of the users may have different disabilities, including various sense disorders or the other physical constraints. There might also be legal pressure made by governments to require accessibility for people with disabilities, which has to be satisfied regardless of the usability of the software. Finally, there may be users with temporary disabilities, e.g. pregnancy, fractures of limbs and so forth.

Support for the age groups

Certainly there exist different requirements following the age structure of the user population, varying from children to the elderly. With the decreasing birthrate the population also gets older, which poses certain requirements for the systems being currently developed. However, this aspect of the acceptability of the system is related also to its usability.

Multilanguage support and localization

Almost every software application delivered nowadays offers the possibility to select the language of the user, which is often considered as an important feature in certain applications. On the other hand, topics of internationalization raise this issue from the other side. Possibly, the application should be run by different nationalities without any restrictions.

Organizational aspects

Integration

One question which decision makers should consider is whether the system they are going to purchase will fit into the curricula of the organization. In other words, how well and how easy the integration of the new system will be, how big an effort the organization has to call forth towards the integration of the system into the current structure from the technological perspective.

Training

A typical question concerning the training may look like: “Will the organization be forced to train the staff in order to allow it to work with the new product and utilize all of the offered features?” Training might also be needed in the future, in the terms of how big an effort has to be expended while upgrading the system to a newer version. The basic question which every learning institution has to raise concerns the lack of familiarity of working with computers, both for the staff and students. Prior attitudes to the use of computers may also influence performance.

Maintenance and administration

Maintenance within the context of practical aspects of the system is a term coupled with the extent of efforts put toward keeping the system running and fulfilling user needs. Part of the maintenance consideration concerns humans, in the sense that human resources are usually needed to maintain the system.

Finance

There is never enough money. The previous sentence will a person responsible for the budget in the company be constantly saying. Costs have to be considered in two ways. First, in investments needed to purchase and install the system and secondly, in the terms of further investments required to maintain the system, for instance to upgrade the product or to pay the support personnel.

Reliability

Every system is expected to be reliable. Reliability means that the system will do and behave exactly the way it is expected to. Moreover, the system has to offer ways how to ensure reliability.

Privacy, Security and Safety

Technology is evolving in a high pace so that the law often does not reflect it. Beside that, the environment should always be concerned with and protect users' privacy, especially under conditions where it is necessary to share and publish information. For instance in networking environments, both operating systems and actual tools have to support access rights. Maximal safety also has to be guaranteed by the system, which should threaten neither users nor their data nor the platform on which it is going to be used.

Fault tolerance and prevention

Fault tolerance is related to the capability of software to maintain a specified level of performance if a certain operation fails or other unpredictable problems occur [29]. There exist many systems that are widely used in which the mean-time between breakdowns is much shorter that users would expect. The best way to avoid the errors is to prevent them; in order to decrease fault occurrence, the system has to actively prevent errors. Error-free performance is also required by Nielsen's definition of usability attributes, but it is partly related to the technological factors too. For instance the defects leading to the malfunction of some parts of the tools should not occur.

3.4 Summary

This chapter introduced the approach of computers as a tool with stress on influence of technological issues to human performance and usability. The most important examples of tools that influence learning have been shown including computer supported learning environments. The match between the context consisting of users, tasks, environment and equipment and practical usability issues has been shown, leading to the definition of the framework for basic technology-related usability issues.

4 PEDAGOGICAL ISSUES OF EDUCATIONAL SOFTWARE

Until now this thesis has described both usability and the technological frameworks concerning systems. When looking at learning environments, we have to establish the basic framework that consists of human learning theories and find the most important implications and influences to computer assisted learning.

4.1 Human learning principles, theories and models

Many human learning theories and models have been introduced and it is peculiar that each of them influenced a certain period of our history as research in education progressed and in the same time, the theories were influenced by outside factors, for instance by technology. The common idea behind almost all of them is that, they stress the importance of the need to learn in order to help learners live; to make decisions, learn, work and adopt into the current (understood relative to the time of each theory) environment. However, not all of them are relevant to the domain of computer or technology supported education, and even less of them have been adopted. This chapter introduces a set of human-learning theories and models which have been found to be important and relevant to computer mediated learning environments.

How do people actually learn? This question has been under heavy research for many years. And yet another topic related to the computer science is, how computer tools should be designed in order to facilitate the learning process? And finally, how to assess the usability problems in the educational tools? Naturally, both of these issues have been investigated at first separately, after which the conclusion has arisen that only cooperation and an interdisciplinary approach can yield tangible results. For educators with lack of computer science related knowledge and usability experience it is very difficult to consider usability and technological factors while implementing educational environments; for computer scientists without proper pedagogical background is a hard quest to properly evaluate the usability of educational environments. Thinking of learning and usability as independent issues in the educational software context is simply impossible. Design of tools for educational purposes has to be driven by educational theory [33].

According to Coles [31], people learn in three ways, which constitutes the three groups of human-learning theories: through modeling and imitation (Social learning theory), through reinforcing the rules and through rewards and consequences (Behavioral learning theory) and through active dialog and thinking (Cognitive-developmental theory). All of these, in my opinion, can be mediated and facilitated by computer-based technology. Social learning theories stress the importance of social surroundings in the learning process over the actual records. Building and maintaining the proper relationships is possible with the aid of computers. Behavioral learning theories advance the construction of reinforcements such as rules, structures, rewards, punishments and consequences. This endeavor is fairly easy to be done by the computers in my opinion. Cognitive-developmental theories are based on the interaction, dialogue, their perception and processing.

The following part briefly introduces the main human learning theories, they are alphabetically ordered, since the division into groups is done on the basis of the previously mentioned three ways of human learning. The attempt to gather human-learning theories relevant to computer aided instruction has been made by Kopponen [35].

Behavioral learning theories

Behaviorism

Behaviorism is more a concept or attitude than a theory of learning. It simply states that learning is the acquisition of new behavior, which is only observable, discounting any mental or other internal learners' activities. Although obsolete, behaviorism still influences the teaching process, where teachers like to punish and reward students in order to support their learning. Many later learning theories based on behaviorism have been applied in the CAL, though behaviorism has been found to be insufficient to fully explain human learning and therefore methods based on it should be regarded accordingly.

Information pickup theory

According to Kearsley [30], information pickup theory “suggests that perception depends entirely upon information in the "stimulus array" rather than sensations that are influenced by cognition.” More further, the perception is seen as a direct consequence of the environmental properties and it does not involve any form of the sensory processing. The environment should be realistic and unconstrained in order to facilitate perception.

Operant Conditioning

The operant conditioning learning theory is based on the concept of stimuli and response which comes from the behavioristic framework. It states that learning is a result of an individual's response to the stimuli in the environment without mentioning any of the internal (mental) states. Responses then produce consequences (associations), the wealth of which depends on the nature and frequency of the stimulus-response occurrence. Moreover, reinforcement is a key element of this theory, serving to strengthen the desired response. One of the basic principles of operant conditioning says that a behavior that is positively reinforced will reoccur and the intermittent reinforcement is particularly effective [30].

Mathematical Learning Theory

This theory attempts to describe and explain the learning process in mere quantitative terms and as also Kopponen notes it plays an important role in the history of computer assisted instruction [34] as it was applied during the beginnings of first real CAI. Quantitative terms are such as the mean time to learn some particular subject, variance of performance of the whole class and so forth. There are two main principles that govern the mathematical learning theory: first, it is possible to develop an optimal learning strategy for the particular learner while detailed model of learning process is available and the second principle states that if the learners have had enough time to learn then the optimal learning performance can be achieved [30].

Cognitive-developmental theories and models

Constructivism

By definition, constructivism is a philosophy of learning based on the premise that our own construction of knowledge is the result of the reflection on our current or past experiences. It means that each of us construct and refine our own models based on our later or current experiences and during time we adjust these models in order to fit them to our newer experiences. In other words, learning and understanding should be “active, constructive, generative processes such as assimilation, augmentation, and self-reorganization” [33].

GOMS

The GOMS model has been introduced already once in the usability part of this thesis and belongs to the cognitive-developmental group of learning theories. GOMS is closely related to human-computer interaction, describing information processing as a sequence of goals, operators, methods and selections between the methods, all serving to search the problem space. Although originally intended for text editing tasks, GOMS influenced many later psychologists and gave the framework for models such as the minimalistic model or the Soar learning theory which have been applied in later research and computer-based services.

GPS

The General problem solver (GPS) was a highly ambitious attempt to simulate human problem solving by a computer simulation program. The solving of the problem has been broken down into sets of the subproblems and after solving the content of the subproblem set, a general problem could be solved [30]. The GPS theoretical framework introduced the use of productions, meant to specify cognitive models. The whole behavior was explained as a function of memory operations, control processes and rules. The description of problems as an input to GPS turned to be critical and therefore GPS could be used only in well defined problems. Although unsuccessful, GPS has had a great impact on later theories such as GOMS or Soar.

Soar

The Soar theory has been based on the previous works of GOMS models and GPS in the sense that it has built up on the idea of searching a problem space. From GPS the Soar theory inherited the use of productions for expressing human cognition [35]. Chunking is the primary concept of learning in the Soar architecture. Currently, Soar is used in a variety of tasks, mainly by artificial intelligence researches to implement a seemingly intelligent task solver, an agent [36].

Social learning theories

Socio-constructivism

In the Soloway's article *Learning theory in practice* author, beside the definition of constructivism, also points out that we have to include a social context of the constructivistic learning, what means that the learning process is an enculturation, in the sense of the collaboration within the certain group defined by its characteristics including common habits, language, practices, beliefs and others. By this contribution, mere constructivism is enriched and becomes socio-constructivism. Socio-constructivism considers both individual and group cognition as the whole idea of learning. For most educators constructivistic approaches play the key role in the learner-centered environments and allow adopting the learning process into the computer supported situations.

Minimalism

The minimalist theory identifies itself as having its roots in the constructivism and gives a framework for designing the instruction; it is especially intended for the development of training materials for (adult) computer users [30]. Similarly to constructivism, minimalist theory suggests the learning tasks to be meaning-full and self-contained activities, the instructions should allow self-directed acquisition of knowledge and models. Moreover, minimalism also stresses the real-world context of learning and learning materials, using tasks that provide error recognition and recovery and it supports the active forms of training. The minimalist theory has been widely applied to the design of computer documentation [30].

Social development theory

The social development theory claims that social interactions play the main role in the development of full cognition within a limited time span, the age of a human. This mean that humans are able to fully possess certain knowledge only during a certain period of their development. The social development theory as a general framework influenced later works, for instance the situated learning theory.

Situated Learning

The situated learning theory puts in the forefront the importances of activity, context and culture in which learning occurs, in other words the situation of learning. This theory forms a contrast with traditional classes in the past, where the information given to students was out of context and usually too abstract. Knowledge needs to be presented in an authentic context according to this theory. It also stresses the social aspects, considering the position of the learner as a part of the community; collaboration is considered as a basic prerequisite leading to successful learning process. Situated learning is related to the information pickup theory and the social development theory, which were its predecessors; it mostly influences computer-based training services [35].

Unclassified learning theories and models

Multiple Intelligences

The theory of multiple intelligences suggests that there are more distinct forms of intelligences (as opposite to one general intelligence which was presumed earlier) and each learner individually possess' every one of these forms in different degrees. The term intelligence means the way how humans perceive and understand the world. Currently we recognize at least seven forms of intelligence (alphabetically): body-kinesthetic (e. g. ability to control the body movements), interpersonal (person-to-person communication and relationships, e. g. social skills), intrapersonal (e.g. insight, self-reflection), linguistic (verbal, ability to use the language), mathematical (e. g. logic, reasoning, inductive and deductive thinking, pattern recognition), musical (rhythmic) and spatial (e. g. visualization). The multiple intelligences theory gives the framework for individual customization of the whole learning process (i. e. also assessment) following the abilities of each learner.

The total human development model

The total human development (THD) model is a framework that integrates psychology, philosophy and technology in teaching. It stresses the equal importance of the previously mentioned sciences to the learning process, which means that the philosophical foundations, such as including development of the whole person within the whole community, the context of relationships and respect and responsibility; psychological foundations, such as social, behavioral and cognitive-developmental learning theories; and technological foundations, such as exploration through reading, understanding through problem solving and action through program development [31]. Since THD is more general and goes beyond the classical human learning theories it can be used as a foundation for the whole learning organization rather than be applied to a specific subject of learning.

4.2 Educational systems and the HCI perspective

When comparing traditional tools with educational systems, the latter are clearly specific in the way that the user has more to learn in an educational context. However, another difference comes up when considering the goals of users [32]. Accepting the idea of a user having multiple goals, one has to establish the comparison between the different levels of focus upon the different goals. Beside the traditional perception, as users' focusing on interaction goals (for instance saving their actual database view), in an educational setting we want to focus on other types of goals.

The relationship between traditional and educational goals also varies. There might sometimes even be a contradiction between them (for example the interface metrics should drive the design of an interface to avoid user confusion, overloading or fluency in navigation, which can in some learning situations be beneficial); also goals may be found to be subservient [32], supporting each other. As Gilmore also claims, the focus on interface goals might cloud the more important goals, which in an educational setting are regarded to be the learning goals. Further more, in his three case studies of learning systems, Gilmore also disaffirms a few basic HCI assumptions by revealing how performance and learning are related. This results in the conclusion of the requirement of having separate learning and performance (usability) goals; shows the importance of learners' mental-models which allows the transfer of learning between different interfaces; and finally Gilmore suggests that “the benefits of direct manipulation (*as known from HCI*) for office users of technology may be a disadvantage to educational users”. These conclusions change the HCI groovy guidelines in view of educational systems.

4.3 Components of learning environments

According to Soloway [33], every educational system has to address three unique needs of learners: *growth*, *diversity* and *motivation*. Growth is the main goal of education towards the learner, the goal of education; it means the evolving learners' knowledge by promoting new ideas. Diversity considers differences amongst learners, their gender, environments, physical conditions, cultures and so forth, in order to accommodate their various needs. The third part of learners' needs is motivation. While evaluating educational environments, pedagogical issues are extremely important and have to be equally taken into account with the usability factors.

Soloway further considers constructivism and socioculturism as two theoretical frameworks currently underlying the reform in education. According to him, educational environments comprise of the *context*, *tasks*, *tools* and *interfaces* [33]. Context is the environment in which the software will be embedded, how it will be used and who will be the users. The second component of learning environments are tasks supported by the software. Tools and interfaces have been covered previously in this thesis, however there are also pedagogy-related issues in them which have to be addressed in the educational environment. Learners have to be given the possibility to address each of their unique needs from each component of the learning environment. For instance, proper tools should be selected and tasks designed to respect learners' growth and diversity and to increase their motivation. At the same time, the tool has to satisfiable fit into the current learning environment and complement other teaching activities at the learning organization.

4.4 Pedagogical aspects of system acceptability

The Soloway's view to the structure of learning environments is a strictly learner-centered orientation. In the real learning processes there are also at least two other parties involved. Namely, the teachers and the management. Therefore, while identifying the essential issues of learning environments, we have to include also these two roles into the consideration. Our initial framework of learning environments adopts Soloway's ideas and consists of a learner, a teacher, and manager and identifies how well are their roles in the learning process supported by an environment from the perspectives of the components of the tool. Using the previously given theories, we match the roles against the components in order to identify the essentials of this intersection, the needs of each participant in the learning process.

In the following paragraphs of this chapter, the most vivid points of the intersection are identified, categorized by the context, tasks, tools and interfaces.

Context

As has been shown earlier about the classical usability evaluation methods, most of them are not context sensitive, which means that they do not take specific environments into consideration. Furthermore, in an educational setting, context sensitivity means that educational goals change with the distinction of educational domains. There are definite differences between software supporting e.g. geography course and software meant to aid math students and similarly, the information offered for university students of geography differs to that offered to pupils in comprehensive schools. We clearly see that the context is the component where the learner's unique needs considerably intersect with the aspect of a tool.

Context

Considering the learners, knowledge needs to be presented in the most authentic context as possible and aligned with the learning goals. This clause is a direct implication of constructivism and the situated learning theory.

Roles

In the learning process it is very important to allow every participant to keep his or her role in the teaching process and when appropriate to facilitate the creation of new profiles, editing stored profiles and changing roles according to the needs.

Personalization and customization

A learning environment has to offer maximal personalization in order to adapt to the learners' needs such as growth. From the theories of learning and developmental psychology we learned that each stage of learners' growth has to be aligned with different learning strategy, context and tasks. In its best, a learning environment should be adaptable to the learners.

A learning environment has to provide many perspectives to perceive the subject of learning and enable their customization. This can be done either by the tool itself, or by a teacher or, if considering self-studies, by a learner.

Cultural diversity

Currently learning also encompasses cultural issues which have to be taken into account. The education nowadays becomes more multicultural than ever and our society becomes a multilingual one. The tools should effectively build on the cultural diversity and, of course, it has to represent knowledge in inoffensive way towards the cultural ethnics.

Credibility and trust

One of the primary aspects of every learning system is the credibility of authors and sources. This becomes more crucial especially for the Internet resources. Therefore, the requirement on all the resources being clearly and rigorously referenced has to be satisfied.

Educational environments should also support learners' trust in them. This support has to be concerned throughout the whole learning process and throughout all the components of a learning environment.

Tasks

Into the tasks group as a component of learning environment, we categorize the activities, means, setting, conditions and the surrounding leading to the successful attaining the learning goals.

Motivation

All of the tasks that lead to fulfilling the learning objectives have to sustain and even increase learners' motivation when needed. The environment has to gain users' engagement and interest, learners often need an additional encouragement to sustain the motivation and interest.

Goals

The traditional perception of the interaction goals falls short when considering learning environments. *Learners* should be given specific and clear learning goals under instructional situations, but while they discover new knowledge the goals should be less focused.

It is also the *teacher* who establishes the goals and wants them to be fulfilled. The use of certain learning environment has to help to satisfy the teacher's goals.

Task sequence and level of abstraction

According to the minimalistic and constructivistic approaches, all learning activities should be self-contained and independent of the sequence. On the other hand, there might be situations, where it is necessary to follow a certain task sequence to properly construct the knowledge.

The level of task abstraction closely corresponds with the learning goals. The learning environment should properly offer a combination of low- and high-level tasks to completely support learners' comprehension.

Real world match, authenticity

One of the most stressed characterizations of an optimal learning process is the match with the real world. The learning is of no value if it refers to artificial environments or tasks. The environment has to allow learners actively acquire the knowledge and the knowledge itself has to correspond with the needs of a modern society.

Knowledge representation match

There has to be offered a proper representation of the learning domain objects for the learners, for instance by the means of an adjustable level of abstraction according to the experience and growth of the learner or by integrating enough means of representation.

Tools

As stated before, all learning environments consist also of the tools being used during the learning process. Technology-related issues of the tools are addressed by a separate chapter of this thesis. Nevertheless, pedagogy-specific or learning-specific tools' factors still need to be covered.

A complex tool should possibly comprehend the whole learning process; in terms of time, the period starting with teachers' material preparation and finishing with the course assessment; and in the terms of persons, including all involved parties, learners, teachers and process managers, classes and surrounding persons, for instance learners' families.

Learning styles

As Collazos points out: “Success in collaborative learning subject matter means both leaning the subject matter (collaboration to learn), and learning how to effectively manage the interaction (learning to collaborate)” [37]. Taking the conclusions from previously mentioned theories, every learning tool should also offer address different learning styles of the learners.

Learning materials management

If applicable, the tool should help teachers prepare, edit and share learning materials. This can be done also by an external application, while the learning environment uses it for the material management.

Learning process management

This part of learning environment offers the perspectives of learning process to the managers. Beside that, the tool should enable different views to learners' performance during time to facilitate assessment. If the nature of learning is based on the collaboration, the tool should facilitate groupwork, e.g. by providing the means for communication, sharing and monitoring of the learning process.

Interfaces

The usability part of this thesis considers the effects of the interface to the performance in a traditional HCI way. However, new consequences arise by taking the learning environment into account, which are not explicitly mentioned as being part of the interface's usability. In addition and as has been shown above, some principles of HCI usability might be in opposition to the fundamental requirements of learning environments. E.g. in the learning situations, error free performance is not always desirable. Even more attention should be put towards feedback, which has to support students by suggesting and encouraging their next steps.

The interface of the environment should distinguish between the learner, tutor and manager. It would not be appropriate if the layout of the interface would lead to misunderstandings of the main conceptualization, for instance it should be able to accept different formats of inputs. Taken altogether, interfaces have to be tailored to the tasks performed and learners' needs.

4.5 Summary

The most important human learning theories have been introduced in this chapter. Behavioral theories are concluded as being on the opposite side of the spectra to the theories based on the constructivism. Constructivistic theories brought the shifts in education, from the teacher-centered to the learner-centered, from the school to life-lasting education, from focused learning to holistic approaches. The modern educational tools should follow these shifts. However, behaviorism and other learning theories still have to be taken into account.

As a conclusion drawn from the theoretical frameworks, the essential pedagogical aspects of educational environments have been introduced. Three unique needs of learners', growth, diversity and motivation have been identified; three main roles involved in the learning process, the learner, the teacher and the process manager have been identified; and learning environments split into the context, tasks, tools and interfaces which yields the categories of pedagogy-related factors. Pedagogical issues of learning environments have been established as the criteria of fulfilling each roles' needs in the distinct parts of the learning environment.

5 USABILITY EVALUATION OF EDUCATIONAL ENVIRONMENTS

5.1 Existing evaluation approaches of educational environments

The lessons learned from the development of traditional usability evaluation methods say that without appropriate evaluation techniques, the developers, designers and other responsible staff do not accumulate enough skills, knowledge and experience to build better systems. This claim is valid throughout all the systems including learning environments and is independent of any application area.

Kopponen in her dissertation argues that the design and evaluation of CAI environments are based on four demands: domain-based demands, instructional demands, user interface demands, and pragmatic demands; her point of view is close to Soloway's division of learning environments presented in the previous chapter. Based on these perspectives of CAI, Kopponen developed a set of criteria used for the evaluation of the CAI courses.

Similarly to the traditional HCI usability evaluation, the dimensions of methods for educational environments vary from summative and formative, through quantitative and qualitative, controlled and formal experiments and observations and informal methods. Considering groupware as an approach to learning environment development, plenty of evaluation methods have been developed for the collaborative domain. For instance heuristic evaluations have been adapted to collaborative environments. Baker shows many of the problems that appear when trying to apply known HCI evaluation methods without proper adaptation [25]. Generally, heuristic evaluation does not focus on mechanisms of interaction, therefore HE here is based on mechanics of collaboration, the basic activities of shared work; this expanded approach is deemed to also uncover problems which cannot be covered by the original method.

Heuristic evaluations have been also used as a model in the work of Squires and Preece [40], directly targeting the prediction of usability in learning environments. They enriched Nielsen's original idea by the socio-constructivist view of learning, yielding to “learning with software heuristics”, which brings better contextual sensitivity to the inspection method. Another example of an adaptation of the classical HCI method is the groupware walkthrough [27], based on the cognitive walkthrough method. Similarly as in the HE adaptation, the groupware walkthrough is based on the mechanics of collaboration.

As mentioned before, many learning environments are based on the multimedia framework. The evaluation of multimedia software does not differ from single-user systems very much, though there are special issues that need to be considered, for instance more attention has to be paid on cognition related factors. One of the evaluation approaches used in multimedia is the guidelines review [28].

Finally, one of the most used approaches to the evaluation of educational systems is the checklists and questionnaires method. In the following section a few of currently used evaluation checklists are introduced.

5.2 Existing evaluation questionnaires

Checklists are one of the most often used usability evaluation techniques and belong to the inspection evaluation methods group. The main purpose of using checklists is to assess a systems' conformance to the established principles of evaluation. Recently, plenty of checklists have been developed, mostly aiming at Web environments. In the next sections I will briefly introduce some current checklists meant to be used by the end-users (in our case by teachers) in the evaluation of educational environments, the Delta checklist and Ravden and Johnson checklist. More complex evaluation based on the checklist approach is introduced in the handbook by the Learning Technology Dissemination Initiative (LTDI) [39].

Ravden and Johnson

The Ravden and Johnson checklist [41] has about 120 questions separated into categories addressing certain aspects of usability, for instance visual clarity, information feedback or consistency, which keeps and leads the evaluators' attention to these aspects. The original Ravden and Johnson checklist influenced many later checklist-based approaches. However, it can be declared to be obsolete as it does not fulfill the requirements laid on contextual sensitivity.

Ravden and Johnson's checklist is divided into eleven sections:

1. Visual clarity
2. Consistency
3. Compatibility
4. Informative feedback
5. Explicitness
6. Appropriate functionality

7. Flexibility and control
8. Error prevention and correction
9. User guidance and support
10. System usability problems
11. General questions on system usability

As argued in [16], the method is thorough in the usability matters. While modified to implicitly expect numerical answers, the use of the Ravden and Johnson checklist has been found artificial by teachers who are used to giving numerical grades; the Ravden and Johnson checklist also does not address pedagogical issues and since it includes more than one hundred questions, it is long to apply. I see the good points of the Ravden and Johnson checklist in leaving out the 'Comments' fields to evaluators and its structured approach which guides the evaluator.

The Delta checklist

As claimed by itself, the Delta checklist “... can be used in order to gauge the quality of a computer-based interactive learning facility with respect to its basic aesthetics, the nature of the learning environment which it provides and the types of pedagogy involved.” The Delta checklist provides 13 categories and two open-ended questions; and originally been developed for the evaluating of educational CD-ROMs.

As I mentioned above, the Delta checklist introduces 13+2 categories:

1. Engagement
2. Interactivity
3. Tailorability
4. Appropriateness of multimedia mix
5. Mode and style of interaction
6. Quality of interaction
7. Quality of end-user interfaces
8. Learning Styles
9. Monitoring and assessment techniques
10. Built-in intelligence
11. Adequacy of ancillary learning support tools
12. Suitability for single user/group/distributed use

13. Availability in terms of cost and delivery platforms
14. Outstanding strengths and attractive features
15. Outstanding limitations and weaknesses

Certainly, the Delta checklist pays attention to the cognitive and pedagogical issues. However, I feel the Delta checklist is not comprehensive comparing to the other methods of usability inspection evaluation and it does not supports a peer review based evaluations.

LTDI checklist

The LTDI checklist is a multipart questionnaire, attempting to cover a wider area of learning environments. This evaluation instrument consists of seven parts meant to be used during the various states of evaluation and implementation of the learning technology. LTDI's '*first step evaluation checklist*' includes about 100 questions and covers aspects of learning effectiveness, usability, presentation and the content of software. The author leaves the decision of what is important and relevant up to the users of checklist, teachers; the checklist is meant to support teachers' decision of which software to use in the learning process, in other words it should guide them throughout the review of the software under selection. The opening part considers issues prior to the selection of learning software, while the following parts guide the evaluators step-by-step and address the first impressions, interaction perception and presentation of information, closing with pedagogical issues of matching strategies with objectives or assessment. After this guided walkthrough, the summarized rating of usability, layout, and academic content attainment of learning objectives is offered for evaluation.

Another part of the LTDI checklist is the open-ended questionnaire for pre- and post-intervention. The tool also gathers information about students' opinions, attitudes and confidence in the software.

The LTDI first step checklist contains several parts; the second part, called the “step by step guide” introduces the checklist consisting of the following categories:

1. First impressions
2. Level of user control and interaction
3. Package design and layout
4. Prioritization and presentation of information
5. Provision of student support
6. Matching strategies with objectives

7. Feedback support for users
8. Assessment
9. Moving between sections
10. Overall evaluation – usability, layout, academic content, attainment of learning objectives

The LTDI checklist seems to be more comprehensive in attempt to the evaluation of learning environments between the checklists introduced previously and to be the nearest to the facilitation of the peer evaluation. However, in my opinion it has also a few considerable drawbacks; LTDI does not address technological factors and it includes misleading questions in the checklist (e.g. “How many icons appear regularly on the screen? Can you describe each of their function?”) without further explanation, which does not give any help to the evaluator and therefore makes the evaluation inaccurate. It may be difficult for inexperienced evaluators to find the relation of questions similar to the one mentioned above to usability issues. While the selection of the parts to be included in the evaluation is up to the teacher, it is thus difficult to compare the results of evaluations of the certain environment.

5.3 Summary

The usability evaluation methods of learning environments currently used have been introduced along with the deeper insight to the three representatives of the checklist-based evaluation. The main characteristics of the Delta checklist, Ravden and Johnson checklist and LTDI checklist have been highlighted. While comparing the capabilities of the introduced checklists, one has to consider the different purposes of each of them and the purposes of the method we are going to develop.

Ravden and Johnson have developed their checklist in the late eighties of the twentieth century when the impact of an innovative technology has been just about to start. This checklist is generally criticized for its extensiveness and difficulties with relating questions to the actual interface. However, the Ravden and Johnson questionnaire is thorough and comprehensive in the terms of traditional usability issues.

The Delta checklist has been primarily developed for the evaluation of multimedia learning CD-ROMs, which had a clear influence on its structure. This is apparent in sections like 'Quality of interaction', 'Learning styles' or 'Monitoring and assessment techniques'. The Delta checklist has been used in the study which aimed to find the preferred evaluation scheme among the teachers and it received a better reception than the Ravden and Johnson checklist [16].

The LTDI checklist is a result of focused study and research at implementing learning technologies in Britain and addresses the most of educational issues from group. It is a comprehensive and multipart questionnaire aimed to be the first aid to the teachers.

6 THE TUP METHOD

The change of context of evaluation to educational environments raises many interesting and important issues. In the previous chapters I have pointed out the most important factors, viewed in the perspective of usability evaluation, underlying design, development and the evaluation of educational systems. Usability methods have been introduced as a traditional approach of evaluations. Technological aspects of systems have been proposed as being integral parts of every evaluation. And finally, pedagogy-related topics have been quoted and shown to be an integral part of educational environments. Thus, the whole model of learning systems evaluation comprises of technological, usability and pedagogical issues taken into equal concentration. This framework ensures the sensitivity of the evaluation to learning environments, while the usability and technology-related issues are still covered. The following part of this thesis establishes the TUP (technology, usability, pedagogy) model.

6.1 TUP approach

It is up to the teachers which software they will select for their courses. The number of learning environments grows rapidly and most likely this ascent will continue in the future. To help educators with this decision and to facilitate their reviews of the environments, an usable usability evaluation method should be developed, with equal attention to usability, pedagogy and technology related issues. Practicing this method should yield easier and accessible comparisons of educational systems and therefore ease the selection of educational software. While the classroom teachers are not usability experts, the proposed method has to offer a way, how to ease the evaluation for them.

Using the checklists in form of questionnaires fulfills these primary requirements and gives the possibility to practice a peer evaluation. Questionnaires have also other considerable advantages. They can be easily maintained, data can be gathered easily and independently from each evaluator so the time span is possibly unlimited. If the design of questionnaires is made with respect to later processing, the retrieval of results can be effectively tailored. Further, while practicing a checklist based evaluation, there is no need for an expert attending to the actual evaluation process.

However, special attention has to be kept on the actual questions of the checklists. The formulation of particular question can influence the evaluators and consequently the results of the evaluation; the language of checklist items should be neutral, the use of negatives should be avoided in order to not mislead the evaluators. The evaluators should be helped to recognize the different importance (weight) of the questions. Similarly important consideration has to be carried towards the form of answers. For instance, the Delta checklist uses both numerical fields for 13 questions and two open-ended (free form) questions. LTDI uses a combination, however mostly boolean YES/NO types of answers are used. Questionnaires themselves can be evaluated for their validity and reliability after a statistically significant amount of data has been collected.

A checklist for evaluating learning environments should be constructed in order to enable wide application and to be easily adaptable for the possible innovative environments. As shown in this thesis, learning systems may differ, starting from the paper-form lecture notes to including collaborative systems, multimedia and Internet based services.

The purpose of the TUP model is to facilitate teachers in their reviewing the learning environments and sharing these reviews [16]; because of the afore mentioned advantages and for the main purposes of evaluation, the checklist approach to the evaluation has been selected. The TUP model equally concentrates on the *technological*, *usability* and *pedagogical* issues, it is aimed to *provide a peer-review* and a*ccumulate the knowledge*.

The general idea of TUP also consists of the actual evaluation tool which uses the proposed checklist. By gathering the organizational information, the data about the environments being evaluated, the evaluators, and the evaluations, the TUP tool becomes an integral part of the TUP model. The previously mentioned additional information needed to be gathered prior to the evaluation is outlined in appendix E.

The TUP method can hardly be categorized according to the division of methods used by this thesis. As based on the use of checklist, it is clearly an usability inspection method. However, from the viewpoint of its usage, it can be categorized also as an inquiry method.

Questionnaires also have a few drawbacks. As mentioned in the usability chapter 2.3, the questionnaire-based studies suffer from a considerably low response rate. In addition, if the checklist starts to be long the application of it is tedious. Evaluators also tend to answer according to more memorable events instead of unbiased experiences. In my opinion, most of the problems related to the use of questionnaires can be anticipated by the careful design of the questionnaire and its supporting system.

There exist a few critiques to the use of checklists as educational evaluation methods, for instance Squires shows seven drawbacks of the checklist approach in their article [40].

Squires claims that “it is difficult to indicate relative weightings for questions”, “selection amongst educational software of the same type emphasises similarities rather than differences”, “the focus is on technical rather than educational issues”, “it is not possible to cope with the evaluation of innovative software”, “it is not possible to allow for different teaching strategies”, “off-computer, teacher generated uses are not considered” and “evaluation in different subject areas requires different sets of selection criteria” [40, p. 471]. The proposed TUP method, the TUP questionnaire and the TUP system based on this model considerably remove most of these drawbacks.

6.2 TUP checklist

The following checklist is based on the previously visited theoretical backgrounds. My purpose is to develop a wide checklist, whose content will be validated during the next period of the development of TUP; it means that the supporting environment will be developed and the questionnaire will be actively used for the evaluation of selected environments. Further experiments will be performed in order to improve the coverage of the method. The checklist is divided into three parts, technological, usability and pedagogical aspects. Further division of each part is based on the previously established schemes. Technology-related aspects are categorized into the groups established in the chapter concerning the technological factors in this thesis. The division of usability factors partly follows the ISO definition of usability and its relation to the context of use and Nielsen's attributes of usability. Pedagogical issues of the TUP checklist are divided into the subgroups according to Soloway's view of educational environments [33]; most of the pedagogical issues arise from the constructivistic theories of learning, although other approaches integration implies the rest.

Types of answers

The checklist itself has to fulfill usability requirements to the extent that it has to be easy to use easy to learn, and it has to be understandable. In addition, evaluators should not be confused by the offered way to answer the questions. The following table overviews the possible types of answers. These will be later used to specify the type of an answer, as long as many different possible ways exist. The possibility of leaving a question unanswered has to be given to the evaluator, for instance by answering 'Not applicable' or 'N/A'.

Questions which belong to the 'short numerically answered questions' and questions expressing attitude offer 'middle' or 'neutral' values; that means there is an odd number of possible values. In addition to the numerical representation, a textual cue is offered for the numerical questions.

Type of answer	Values	Denotation
Open ended	Free form text	A
Boolean	YES / NO	B
Boolean tick	Box ✔/ Cross ✕	B
Short numerical[2]	5..1	C1
Short numerical + zero[2]	5..0	C2
Long numerical[2]	10..1	D1
Attitude	Strong confirmation .. Strong disagreement	E

Table 2 Denotation of answers

Technological aspects

The following table summarizes the checklist questions addressing the technological aspects of learning environments. The division follows the categories established in chapter 3.3. If not stated otherwise, the default type of an answer is a boolean type B.

Technological aspects Subgroup	Questions	Type of answer
Availability and Compatibility		
Software Compatibility	Does the system support the import/export of external resources?	
	Is it possible to use the tool on different operating systems?	
	Does the tool need additional software or components to be installed?	
	Does the tool satisfy the requirements on an free open source software?	
Hardware Compatibility	Does the system need some additional equipment to function properly?	
	Do you need to use unusual external interfaces, devices or media?	
Availability	Is the required equipment readily available?	

2 Accompanied by a textual aid appropriate to the question, e.g. :'Very satisfacted..Very unsatisfied' or 'Very low..Very high'

Technological aspects Subgroup	Questions	Type of answer
	It is easy to set up the environments necessary for the tool.	E
	It is easy to install the tool (e.g. by an installation wizard).	E
Accessibility		
Support for disabled	Is the system adaptable to the needs of disabled people?	
Support for the age groups	Does the system distinguish between the different age groups of users?	
Localization	Does the system follow the regional setting, e.g. format of numbers, time format, keyboard?	
Multilanguage support	Is it possible to change the language of the environment?	B
Organizational aspects		
Maintenance	Is it easy to maintain the system?	
	The system requires much maintenance.	E
Administration	The system is easy to administrate.	E
Training	Is it necessary to train personnel in order to use the tool?	
Finance	Is it expensive to purchase and run the system?	
Integration	Does the system fit into the organization's technological structure?	
Reliability		
Privacy	Is the privacy for users guaranteed?	
Security	Are the security measures adequate for the system?	
Safety	Did you experience any health risks while using the tool?	
Fault tolerance	If any fault occurs, does it cause the breakdown of the system?	
Fault prevention	Does the system actively prevent the faults?	
Defects	The system is free of technological defects leading to malfunctions.	
	The system offers diagnostics tools in order to find possible hardware or software defects.	

Technological aspects Subgroup	Questions	Type of answer
General technological notes		
	Did you notice any technological problems causing your learning objectives not to be attained?	A

Table 3 Technological aspects

Usability aspects

Beside technology-related aspects, the TUP model also equally concentrates on the usability aspects of learning environments. The following list is obtained by taking the Nielsen's views of usability, the ISO definition and the categories of established usability checklists. However, considering the influence of usability to the learning process has to take its part here along with the implications arising from the core usability principles. As long as the pedagogical factors are essential in the TUP model, in cases of an inconsistency pedagogy is superior to usability. Therefore, the aspects usually covered by usability evaluation (for instance the match between the real world and the system corresponding with the constructivistic view to learning as well) were moved or partly moved into the pedagogical section. While there are contradictions between usability and pedagogical requirements, traditional usability aspects such as the low error rate are not included.

Usability aspects Subgroup	Question	Type of answer
Learnability		
	Users can rapidly start working with the tool without a long period of training.	E
	Does the usage of the tool impose heavy cognitive load to the users causing problems?	
	Is it possible to select advanced modes of control according to the users' experience?	
Interaction		
Modes of interaction	Does the tool offer various interaction modes (e.g. sound, haptic channel) ?	

Usability aspects Subgroup	Question	Type of answer
	The level of interaction with the tool corresponds to the users' characteristics.	E
	Are the means of interaction properly selected?	
Reversibility	Do the users have the ability to 'undo' and 'redo' their actions?	
Response time	Is the feedback offered by the tool within a reasonable time?	
Shortcuts	Does the system offer shortcuts for the most often invoked commands or sequences?	
Help	Does the system offer any help facility?	
	Is the help easily accessible?	
Navigation	Are the users always properly informed about their position in the system structure?	
	Navigation within the environment is easy and natural for users (e.g. the structure of menu).	E
	Are users facilitated to search within the environment?	
Memorability	Does the tool require re-training of usage after breaks?	
	Does the environment require users to memorize facts unrelated to the learning objectives?	
Aesthetics	The design of the interface is aesthetic.	E
Visual aspects		
Text	Is text in the system clearly legible?	
Graphics	The tool uses graphics appropriately.	E
Organization	The displayed information is properly organized.	E
Audio	The tool uses sound appropriately.	E
Overall usability		
Consistency	Is the environment consistent in terms of design, navigation and terminology?	

Usability aspects Subgroup	Question	Type of answer
	The designer's model of system corresponds with the users' view, expectations and perception of it.	E
Localization	The system complies with the local setting and habits.	E
	Is it possible to use the system by multiple nationalities without restrictions?	
Support for age groups	Is it possible to use the system by various age groups?	
General comments on usability	Which problems have you noticed while working with the tool?	A

Table 4 Usability aspects

Pedagogical aspects

The pedagogical issues form the core of this checklist and thus they are superior to the other issues of the system evaluation. The grouping of issues follows the categories established in 4.4. Referred “learners' needs” in the following table comply with the unique needs defined in the chapter 4.3.

Pedagogical aspects Subgroup	Question	Type of answer
Context		
Context	Is the subject of learning presented in the authentic context?	
	The usage of the tool fits into the learning setting (e.g. class versus individual learning)	E
	The tool conveniently complements other class activities.	E
	The content fits into the institutional curriculum.	E
Roles	Does the system distinguish between the roles of the users in the learning process?	
Personalization	Is the environment self-adaptable in order to reflect learners' growth?	
Customization	Does the system enable the customization of the perspectives of learning objects according to the learners' needs?	

Pedagogical aspects Subgroup	Question	Type of answer
Cultural diversity	Cultural issues are addressed and properly handled by the environment.	E
Credibility	Are the resources and references used in the tool credible?	
	Is the information presented by the tool up to date?	
Trust	Do you feel the tool maintains trust in it?	
	Learners' ownership of learning is maintained by the environment.	E
Task		
Motivation	Are the tasks designed to maintain (or increase) learners' motivation?	
Goals	Is the selection of the learning goals appropriate?	
	Is it possible to modify the learning goals?	
Task sequence	Does tool enable multiple paths to achieve the learning goals?	
	Are the task sequences adaptable according to the learners' growth?	
Task abstraction	The level of abstraction corresponds with the learners' growth.	E
Real world match	Learning tasks correspond with the real world tasks?	E
Knowledge representation match	Does the system provide enough means for the proper knowledge representation?	
	Does the environment provide multiple representation of knowledge?	
Tools		
Materials management	Can the teacher prepare and modify learning materials using the tool?	
	Can the teachers share learning materials using the tool?	
	Is it possible for users to prepare a presentation using this environment?	
Process management	Does the tool facilitate the evaluation of the learning process?	
	Is it possible to create tests, examinations or assignments using the tool?	

Pedagogical aspects Subgroup	Question	Type of answer
	Does the tool provide process management to be secure for all the parties?	
	The process management follows organizational requirements.	E
Learning styles	Does the tool offer self-directed learning?	
	Does the tool enable different learning styles (e.g learning by exploration, learning by doing etc.)?	
	Does the tool facilitate groupwork?	
	Is it possible to create notes or bookmarks within the tool?	
	The tool supports the creation and sharing of learning artifacts?	E
	Does the tool support off-computer learning activities? (e.g. motivates discussion)	
Interfaces		
Layout	The layout of the interface supports attaining the learning objectives.	E
	The interface layout contains all of the information necessary to achieve learning goals.	E
Tailorization	The interface of the tool is designed according to the target learners' needs.	E
	Is it possible to tailor the interface according to the individual learners' needs?	
	While interacting with learners, does the tool accept also alternative responses?	
Overall pedagogy	Do you attain your learning objectives with the tool?	
	Would you recommend the tool for learning purposes?	A
	What do you think about the educational capabilities of the tool?	A

Table 5 Pedagogical aspects

6.3 Additional notes about the TUP questionnaire

In the present, the TUP questionnaire represent a wide spectrum of issues to be considered during the evaluation process. The content of questionnaire and therefore the questions and the actual evaluation process have to be studied and analyzed into the detail and the final version of the TUP model will be derived from that. Currently, the TUP model consists of 96 questions, where the technological part contains 27 items, the usability part contains 27 items and pedagogical part addresses 42 issues at the moment.

After the TUP questionnaire's questions have been established, it is necessary to add a few comments on the use of the questionnaire. Every question should be accompanied by a thorough explanation in order to help the evaluator in relating the question to the actual environment. What did not fit into the tables above is the field offering the evaluator to add a comment to every question. The whole evaluation process is structured into the three basic parts: technology, usability and pedagogy. The actual proceeding of the evaluation should be considered.

6.4 Summary

The TUP model constituting of technology, pedagogy and usability has been introduced. Rationales of using the checklist approach to the evaluation have been presented. The TUP model has been established with the TUP checklist in the form of questionnaire.

7 CONCLUSIONS AND FUTURE DIRECTIONS

The usability of educational environments cannot be assessed by traditional HCI methods. This thesis has shown a few HCI paradigms which are in the contrary to the requirements of the proper evaluation of educational environments. Traditional HCI evaluation methods of usability mostly fall short in addressing the essential parts of learning environments and the unique learners' needs as long as learning cannot be approached in the pure usability sense. However, every system interacting with humans will always include some kind of interface and that is the point where traditional HCI methods take part. This thesis argues that it is possible to overcome the main drawbacks of using questionnaires for the purposes of evaluation of educational environments.

The development of the TUP questionnaire has brought plenty of questions which should be answered during the next stages. Namely: it is clear that the set of the pure usability principles has been reduced while other parts of the questionnaire are strengthened. Does it indicate that usability as such is not that important in educational environments? Or does it pose a question of rating the importance of usability within educational environments? Another issue interlacing throughout the thesis is the context of education. What should be included in the context of educational environments and what issues have such an influence on learning that we have to include them in the evaluation? The development of the questionnaire has also shown that we have to gather information describing the applications and also the background of the evaluators, their institutions, environments and courses supported by the evaluated tools. The previously mentioned information are valuable sources for the potential users of the tool. Considering the maturity of evaluators and use of the checklist, the system facilitating the evaluation has to be developed. The important issue which must be considered is that the TUP method heavily relies on the abilities of the evaluators, the teachers. Their actual performance during the evaluation clearly influences the results of the study. Finally, the pedagogical part of the TUP questionnaire is mostly supported by socio-constructivism. This theoretical approach is not completely developed nor implemented into the educational environments which may cause dissatisfaction with the results of a TUP based evaluation.

While thinking about the future stages of the development of TUP, first the evaluation tool has to be developed to facilitate the use of the TUP questionnaire. Then a few studies have to be launched in order to collect data and evaluate the questionnaire itself. My idea of the tool supporting the TUP based evaluation includes build-in intelligence enabling the automatic adaptation of the checklist to the actual application in order to facilitate the evaluation. This feature should ease the evaluation for the nonprofessional evaluators, which are the main group of users of the TUP.

In conclusion, I feel that this thesis has installed more questions than it has answered so far. It is the result of the unexplored interdisciplinary topic in focus and it brings more challenges for the future. This thesis, however, provides the fundamental framework for further research and development.

8 REFERENCES

[1]International Standardization Organization – ISO homepage: ISO 9241. Available at http://www.iso.ch/iso/en/ISOOnline.frontpage, August 2002

[2]ACM Special Interest Group on Computer-Human Interaction Curriculum Development Group: ACM SIGCHI Curricula for Human-Computer Interaction. Technical Report, ACM, New York, 1992,1996

[3]Bevan, N.: Human-Computer Interaction Standards. Proceedings of the 6th International Conference of HCI, Yokohama, July 1995

[4]Dix, A., Finlay J., Abowd G., Beale R.: Human-Computer Interaction. Prentice Hall, 1993

[5]Bennett, J.: Analysis and Design of the User Interface for Decision Support Systems. In J. Bennett, ed., Building Decision Support Systems. Addison-Wesley, Reading. 41-64.1983

[6]Nielsen, J.: Usability Inspection Methods.Conference companion on Human factors in computing systems, p.377-378, Denver, Colorado, United States, 1995. Available at http://portal.acm.org/citation.cfm?doid=223355.223730, August 2002

[7]Butler K. A.: Usability Engineering Turns 10. Interactions, p. 58-75, ACM Press, January 1996

[8]Ebling M. R., John B. E.: On the Contributions of Different Empirical Data in Usability Testing. Symposium on Designing Interactive Systems 2000, p. 289-296, Brooklyn, New York, 2000.

[9]Smith S. L., Mosier J. N.: Guidelines for designing user interface software. The MITRE Corporation, Bedford, Massachusetts, USA 1986. Available at : ftp://archive.cis.ohio-state.edu/pub/hci/Guidelines/, August 2002

[10]Nielsen J.: Usability Engineering. Academic Press, 1993

[11]Dumas J. S., Redish J. C.: A Practical Guide to Usability Testing, revised edition. Intellect, 1999.

[12]Hartson H. R., Castillo J. C., Kelso J., Neale W. C.: Remote Evaluation: The Network as an Extension of the Usability Laboratory. CHI96 Conference Proceedings, vol. 1, p. 228-235. 1996

[13]Jacobsen N. E., John B. E. : Two Case Studies in Using Cognitive Walkthrough for Interface Evaluation. Computer Science Technical Report CMU-CS-00-132, Carnegie Mellon University, USA, 2000

[14]Blackmon M. H., Polson P. G., Katajima M., Lewis C.: Cognitive Walkthrough for the Web. Proceedings of the SIGCHI 2002 conference on Human factors in computing systems. ACM Press. April 2002

[15]Mayhew D. J.: The Usability Engineering Lifecycle. CHI'98 Tutorials. ACM Press, April 1998

[16]Gert P., Miraftabi R., Tukiainen M.: Evaluating Educational Software Environments. ICCE 2002.

[17]Nielsen J., Mack R. L.: Usability Inspection Methods. Wiley, 1994

[18]Zhang Z., Basili V., Shneiderman B.: Perspective-based Usability Inspection: An Empirical Validation of Efficacy. Empirical Software Engineering: An International Journal, Volume 4, No. 1, p. 43-69, March 1999.

[19]Doubleday A., Ryan M., Springett M., Sutcliffe A.: A comparison of usability techniques for evaluating design. Proceedings of the conference on Designing interactive systems : processes, practices, methods, and techniques: processes, practices, methods, and techniques, p.101-110, August 18-20, 1997, Amsterdam, The Netherlands

[20]John B. E., Kieras D. E.: Using GOMS for User Interface Design and Evaluation: Which Technique? ACM Transactions on Computer-Human Interaction, vol. 3, p. 287-319. ACM Press, 1996

[21]de Haan G. et al.: Formal Modelling Techniques in Human-Computer Interaction. In G. C. van der Veer, S. Bagnara, and G. A. M. Kempen, Eds., Cognitive Ergonomics: Contributions from Experimental Psychology, Theoretical Issues, p. 27--67. Amsterdam, The Netherlands: Elsevier Science Publishers, 1992. Available at: http://www.mmi.unimaas.nl/userswww/dehaan/articles/formal-models-review.html, September 2002

[22]Chi ed H., Pirolli P., Pitkow J.: The scent of a site: a system for analyzing and predicting information scent, usage, and usability of a Web site. Proceedings of the CHI 2000 conference on Human factors in computing systems. ACM Press, April 2000

[23]de F.Q.V. Turnell M., Scaico A., de Sousa M. R. F., Perkusich A.: Industrial User Interface Evaluation Based on Coloured Petri Nets Modelling and Analysis. In LNCS 2220: Interactive Systems: Design, Specification, and Verification ed. by C. Johnson. Springer, 2001

[24]Nua: How Many Online? Available at: http://www.nua.ie/surveys/how_many_online/, September 2002

[25]Baker K., Greenberg S., Gutwin C.: Heuristic Evaluation of Groupware Based on the Mechanics of Collaboration. Proceedings of the 8th IFIP Working Conference on Engineering for Human-Computer Interaction (EHCI'01), May 11-13, 2001, Toronto, Canada.

[26]Steves M. P., Morse E., Gutwin C., Greenberg S.: A Comparison of Usage Evaluation and Inspection Methods for Assessing Groupware Usability. Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work - Volume 2001 September 2001

[27]Pinelle D., Gutwin C.: Groupware Walkthrough: Adding Context to Groupware Usability Evaluation. Proceedings of the SIGCHI conference on Human factors in computing systems. Minneapolis, Minnesota, USA. April 2002

[28]Bearne M., Jones S., Sapsford-Francis J.: Towards usability guidelines for multimedia systems. Proceedings of the second ACM international conference on Multimedia. October 1994

[29]DTR 9126-2:ISO/IEC 9126-2: Software Engineering - Product Quality Part 2 -External Metrics. ISO, January 2001

[30]Kearsley G.: Theory Into Practice – TIP database. Available at: http://www-hcs.derby.ac.uk/tip/ or: http://tip.psychology.org/, September 2002

[31]Coles R.: Disparity Between Intellect and Character : THD – total human development model. Available at: http://www.newhampton.org/thd/, September 2002

[32]Gilmore D. J.: Interfaces for Supporting Learning. Proceedings of the Interdisciplinary Workshop on Complex Learning in Computer Environments, p. 91-96. Joensuu, Finland. May 1994

[33]Soloway E., Scala N., Jackson S. L. et al. : Learning theory in practice: case studies of learner-centered design. Conference proceedings on Human factors in computing systems, p. 189-196. ACM. NY, USA, April 1996

[34]Kolehmainen P. : Understanding Software Designers' and Teachers' Perpectives: Towards New Kind of Evaluation of Computer-Based Learning Environments. Proceedings of the Interdisciplinary Workshop on Complex Learning in Computer Environments. Joensuu, Finland. May 1994

[35]Kopponen M. : CAI in CS. Academic dissertation, Joensuu, Finland, 1997

[36]The Soar Group: Soar Home Page. Available at http://ai.eecs.umich.edu/soar, September 2002

[37]Collazos C. A., Guerrero L. A., Pino J. A., Ochoa S. F.: Evaluating Collaborative Learning Processes. Proceedings of the 8th International Workshop on Groupware (CRIWG' 2002), La Serena, Chile. Springer, September 2002.

[38]Delta checklist. Available at: http://www-interact.eng.cam.ac.uk/CAL95/Eval-Checklist1.html, September 2002

[39]Learning Technology Dissemination Initiative : Implementing Learning Technology. Available at: http://www.ltc.hw.ac.uk/ltdi/implementing-it/, September 2002

[40]Squires D., Preece J.: Predicting quality in educational software : Evaluating for learning, usability and the synergy between them. Interacting with Computers, vol. 11, p. 467-483. Elsevier, 1999

[41]Ravden S., Johnson G. : Evaluating Usability of Human-Computer Interfaces, a practical method. Ellis Horwood Limited, Chichester, 1989

[42]Lewis C., Polson P., Wharton C., Rieman J.: Testing a Walkthrough Methodology for Theory-Based Design of Walk-Up-and-Use Interfaces. Proceedings of ACM CHI'90 Conference on Human Factors in Computing Systems, p.235-242, 1990

9 ABBREVIATIONS

Abbreviation	Description	Chapter of the first occurrence or definition
ACM	Association for Computing Machinery	
CAL	Computer Aided Learning	3.2
CSCL	Computer Supported Collaborative Learning	3.2
CW	Cognitive Walkthrough	2.3
CWW	Cognitive Walkthrough for the Web	2.3
GOMS	Goals, Operators, Methods, Selection rules	2.3
GPS	General Problem Solver	4.1
HCI	Human-Computer Interaction	1.1
HE	Heuristic Evaluation	2.3
ISO	International Standardization Organization	1.1
KLM	Keystroke-Level Model	2.1
TUP	Technology Pedagogy Usability model	
UCD	User Centered Design	2.1
UE	Usability Engineering	1.1
UEM	Usability Evaluation Methods	2.3
UI	User Interface	1.1
WWW	World Wide Web	3.2

APPENDIX A- The set of heuristics for Heuristic evaluation

http://www.useit.com/papers/heuristic/heuristic_list.html

Ten Usability Heuristics

by Jakob Nielsen

Visibility of system status
: The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

Match between system and the real world
: The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom
: Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards
: Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention
: Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Recognition rather than recall
: Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use
: Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design
: Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors
: Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation
: Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

APPENDIX B- The Delta checklist

http://www-interact.eng.cam.ac.uk/CAL95/Eval-Checklist1.html

EC DELTA Project ILDIC

Integrating Learning Design in Interactive Compact Disc

The Evaluation Check-list

The check-list presented below can be used in order to gauge the quality of a computer-based interactive learning facility with respect to its basic aesthetics, the nature of the learning environment which it provides and the types of pedagogy involved. Individual products should be assessed with respect to each of the broad evaluation categories presented in the check-list.
Basic Evaluation Check-list

1. Engagement
2. Interactivity
3. Tailorability
4. Appropriateness of multimedia mix
5. Mode and style of interaction
6. Quality of interaction
7. Quality of end-user interfaces
8. Learning Styles
9. Monitoring and assessment techniques
10. Built-in intelligence
11. Adequacy of ancillary learning support tools
12. Suitability for single user/group/distributed use
13. Availability in terms of cost and delivery platforms
14. Outstanding strengths and attractive features
15. Outstanding limitations and weaknesses

Notes are provided in the questionnaire which outline what is involved in applying each of the evaluation categories listed above. These notes should be used as an aide memoire; they should be read prior to conducting either a single or a batch of assessment experiments. Included with the notes is a list of key questions which are best felt to encompass the learning design features of each evaluation category. Such questions should be applied to each software product and used to form the overall conclusion about that product.

It is important to realise that no attempt is being made to assess the learning effectiveness of products since this would require more extensive controlled experiments involving both pre-tests and post-tests.

---

interact-evaluation@icbl.hw.ac.uk

APPENDIX C- Ravden and Johnson checklist

3

The evaluation checklist

INSTRUCTIONS FOR COMPLETING THE CHECKLIST

Sections 1 to 9: criterion-based questions

(1) Each of these sections is based on a different criterion, or 'goal', which a well-designed user interface should aim to meet. The criterion is described at the beginning of the section, and consists of:

— a heading (e.g. 'Visual clarity'), followed by
— a statement (e.g. 'Information displayed on the screen should be clear, well-organized, unambiguous and easy to read').

(2) A number of checklist questions follow, and these aim to assess whether the user interface meets the criterion.

For example, in Section 1 ('Visual clarity'), the questions check whether information which is displayed on the screen is clear, well-organized, unambiguous and easy to read.

(3) To the right of the checklist questions, you will see four columns, labelled **'Always'**, **'Most of the time'**, **'Some of the time'** and **'Never'**.

For each checklist question, please tick the column which best describes your answer to the question.

(4) Then write any comments which you feel you could make when answering a checklist question, in the column labelled: **'Comments'**.

For example, when answering question 12 in Section 1: 'Is information on the screen easy to see and read?', you may tick the column 'some of the time', and you may mention particular screens where information was very difficult to see and read, in the 'Comments' column.

(5) If you feel that a checklist question is not relevant to the interface which you are evaluating (e.g. questions relating to colour if the system does not use colour, questions referring to printouts if there is no printer attached), then please write: **'Not applicable'** or **'N/A'** in the 'Comments' column beside that question, and move on to the next question.

(6) After the checklist questions in each section, you are asked for: **'. . . any comments (good or bad) . . .'** which you would like to add concerning the issues in that section.

For example, you may wish to describe a particular problem, or make a particular point which you did not have room to make beside the checklist question, or you may feel that the checklist questions have not covered a particular aspect of the interface which you feel should be mentioned.

(7) At the end of each section, you will see a rating scale, ranging from **'Very satisfactory'** to **'Very unsatisfactory'**. Please tick the box which best describes the way you feel about the user interface in terms of the issues in that section.

Section 10: system usability problems

(1) The questions in this section ask you about specific problems which you experienced when carrying out the evaluation task(s).

(2) To the right of each question you will see three columns labelled: **'No problems'**, **'Minor problems'** and **'Major problems'**.

For each question, please tick the column which is most appropriate.

(3) As in Sections 1 to 9, please write any particular comments, descriptions of problems, and so on, in the column labelled **'Comments'**, beside each question.

(4) If there are any questions which you feel are not relevant to the interface which you are evaluating, then please write: **'Not applicable'** or **'N/A'** in the 'Comments' column for that question.

Section 11: general questions on system usability

This section asks you to give your views on the interface which you have been evaluating. Please feel free to write as much as you like in answer to each question.

SECTION 1: VISUAL CLARITY

Information displayed on the screen should be clear, well-organized, unambiguous and easy to read.

	Always	Most of the time	Some of the time	Never	Comments
1. Is each screen clearly identified with an informative title or description?					
2. Is important information highlighted on the screen? (e.g. cursor position, instructions, errors)					
3. When the user enters information on the screen, is it clear: (a) where the information should be entered?					
(b) in what format it should be entered?					
4. Where the user overtypes information on the screen, does the system clear the previous information, so that it does not get confused with the updated input?					
5. Does information appear to be organized logically on the screen? (e.g. menus organized by probable sequence of selection, or alphabetically)					
6. Are different types of information clearly separated from each other on the screen? (e.g. instructions, control options, data displays)					
7. Where a large amount of information is displayed on one screen, is it clearly separated into sections on the screen?					
8. Are columns of information clearly aligned on the screen? (e.g. columns of alphanumerics left-justified, columns of integers right-justified)					
9. Are bright or light colours displayed on a dark background, and vice versa?					
10. Does the use of colour help to make the displays clear?					
11. Where colour is used, will all aspects of the display be easy to see if used on a monochrome or low resolution screen, or if the user is colour-blind?					
12. Is the information on the screen easy to see and read?					

	Always	Most of the time	Some of the time	Never	Comments
13. Do screens appear uncluttered?					
14. Are schematic and pictorial displays (e.g. figures and diagrams) clearly drawn and annotated?					
15. Is it easy to find the required information on a screen?					

16. Are there any comments (good or bad) you wish to add regarding the above issues?

17. Overall, how would you rate the system in terms of visual clarity?
(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 2: CONSISTENCY

The way the system looks and works should be consistent at all times.

	Always	Most of the time	Some of the time	Never	Comments
1. Are different colours used consistently throughout the system? (e.g. errors always highlighted in the same colour)					
2. Are abbreviations, acronyms, codes and other alphanumeric information used consistently throughout the system?					
3. Are icons, symbols, graphical representations and other pictorial information used consistently throughout the system?					
4. Is the same type of information (e.g. instructions, menus, messages, titles) displayed: (a) in the same location on the screen?					

	Always	Most of the time	Some of the time	Never	Comments
(b) in the same layout?					
5. Does the cursor appear in the same initial position on displays of a similar type?					
6. Is the same item of information displayed in the same format, wherever it appears?					
7. Is the format in which the user should enter particular types of information on the screen consistent throughout the system?					
8. Is the method of entering information consistent throughout the system?					
9. Is the action required to move the cursor around the screen consistent throughout the system?					
10. Is the method of selecting options (e.g. from a menu) consistent throughout the system?					
11. Where a keyboard is used, are the same keys used for the same functions throughout the system?					
12. Are there standard procedures for carrying out similar, related operations? (e.g. updating and deleting information, starting and finishing transactions)					
13. Is the way the system responds to a particular user action consistent at all times?					

14. Are there any comments (good or bad) you wish to add regarding the above issues?

15. Overall, how would you rate the system in terms of consistency?
(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 3: COMPATIBILITY

The way the system looks and works should be compatible with user conventions and expectations.

	Always	Most of the time	Some of the time	Never	Comments
1. Are colours assigned according to conventional associations where these are important? (e.g. red = alarm, stop)					
2. Where abbreviations, acronyms, codes and other alphanumeric information are displayed: (a) are they easy to recognize and understand?					
(b) do they follow conventions where these exist?					
3. Where icons, symbols, graphical representations and other pictorial information are displayed: (a) are they easy to recognize and understand?					
(b) do they follow conventions where these exist?					
4. Where jargon and terminology is used within the system, is it familiar to the user?					
5. Are established conventions followed for the format in which particular types of information are displayed? (e.g. layout of dates and telephone numbers)					
6. Is information presented and analysed in the units with which the users normally work? (e.g. batches, kilos, dollars)					
7. Is the format of displayed information compatible with the form in which it is entered into the system?					
8. Is the format and sequence in which information is printed compatible with the way it is displayed on the screen?					
9. Where the user makes an input movement in a particular direction (e.g. using a direction key, mouse, or joystick), is the corresponding movement on the screen in the same direction?					
10. Are control actions compatible with those used in other systems with which the user may need to interact?					

	Always	Most of the time	Some of the time	Never	Comments
11. Is information presented in a way which fits the user's view of the task?					
12. Are graphical displays compatible with the user's view of what they are representing?					
13. Does the organization and structure of the system fit the user's perception of the task?					
14. Does the sequence of activities required to complete a task follow what the user would expect?					
15. Does the system work in the way the user thinks it should work?					

16. Are there any comments (good or bad) you wish to add regarding the above issues?

17. Overall, how would you rate the system in terms of compatibility?
(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 4: INFORMATIVE FEEDBACK

Users should be given clear, informative feedback on where they are in the system, what actions they have taken, whether these actions have been successful and what actions should be taken next.

	Always	Most of the time	Some of the time	Never	Comments
1. Are instructions and messages displayed by the system concise and positive?					

	Always	Most of the time	Some of the time	Never	Comments
2. Are messages displayed by the system relevant?					
3. Do instructions and prompts clearly indicate what to do?					
4. Is it clear what actions the user can take at any stage?					
5. Is it clear what the user needs to do in order to take a particular action? (e.g. which options to select, which keys to press)					
6. When the user enters information on the screen, is it made clear what this information should be?					
7. Is it made clear what shortcuts, if any, are possible? (e.g. abbreviations, hidden commands, type-ahead)					
T 8. Is it made clear what changes occur on the screen as a result of a user input or action?					
T 9. Is there always an appropriate system response to a user input or action?					
10. Are status messages (e.g. indicating what the system is doing, or has just done): (a) informative?					
(b) accurate?					
11. Does the system clearly inform the user when it completes a requested action (successfully or unsuccessfully)?					
12. Does the system promptly inform the user of any delay, making it clear that the user's input or request is being processed?					
13. Do error messages explain clearly: (a) where the errors are?					
(b) what the errors are?					
(c) why they have occurred?					
14. Is it clear to the user what should be done to correct an error?					
T 15. Where there are several modes of operation, does the system clearly indicate which mode the user is currently in? (e.g. update, enquiry, simulation)					

16. Are there any comments (good or bad) you wish to add regarding the above issues?

17. Overall, how would you rate the system in terms of informative feedback? (Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 5: EXPLICITNESS

The way the system works and is structured should be clear to the user.

	Always	Most of the time	Some of the time	Never	Comments
T 1. Is it clear what stage the system has reached in a task?					
T 2. Is it clear what the user needs to do in order to complete a task?					
3. Where the user is presented with a list of options (e.g. in a menu), is it clear what each option means?					
4. Is it clear what part of the system the user is in?					
5. Is it clear what the different parts of the system do?					
6. Is it clear how, where and why changes in one part of the system affect other parts of the system?					
7. Is it clear why the system is organized and structured as it is?					
8. Is it clear why a series of screens are sequenced as they are?					
9. Is the structure of the system obvious to the user?					
10. Is the system well-organized from the user's point of view?					

	Always	Most of the time	Some of the time	Never	Comments
11. Where an interface metaphor is used (e.g. the desk-top metaphor in office applications), is this made explicit?					
12. Where a metaphor is employed, and is only applicable to certain parts of the system, is this made explicit?					
13. In general, is it clear what the system is doing?					

14. Are there any comments (good or bad) you wish to add regarding the above issues?

15. Overall, how would you rate the system in terms of explicitness?
(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 6: APPROPRIATE FUNCTIONALITY

The system should meet the needs and requirements of users when carrying out tasks.

	Always	Most of the time	Some of the time	Never	Comments
1. Is the input device available to the user (e.g. pointing device, keyboard, joystick) appropriate for the tasks to be carried out?					
2. Is the way in which information is presented appropriate for the tasks?					
3. Does each screen contain all the information which the user feels is relevant to the task?					
4. Are users provided with all the options which they feel are necessary at any particular stage in a task?					
5. Can users access all the information which they feel they need for their current task?					

	Always	Most of the time	Some of the time	Never	Comments
6. Does the system allow users to do what they feel is necessary in order to carry out a task?					
7. Is system feedback appropriate for the task?					
8. Do the contents of help and tutorial facilities make use of realistic task data and problems?					
9. Is task-specific jargon and terminology defined at an early stage in the task?					
10. Where interface metaphors are used, are they relevant to the tasks carried out using the system?					
11. Where task sequences are particularly long, are they broken into appropriate subsequences? (e.g. separating a lengthy editing procedure into its constituent parts)					

12. Are there any comments (good or bad) you wish to add regarding the above issues?

13. Overall, how would you rate the system in terms of appropriate functionality?
(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 7: FLEXIBILITY AND CONTROL

The interface should be sufficiently flexible in structure, in the way information is presented and in terms of what the user can do, to suit the needs and requirements of all users, and to allow them to feel in control of the system.

	Always	Most of the time	Some of the time	Never	Comments
1. Is there an easy way for the user to 'undo' an action, and step back to a previous stage or screen? (e.g. if the user makes a					

	Always	Most of the time	Some of the time	Never	Comments
2. Where the user can 'undo', is it possible to 'redo' (i.e. to reverse this action)?					
3. Are shortcuts available when required? (e.g. to bypass a sequence of activities or screens)					
4. Do users have control over the order in which they request information, or carry out a series of activities?					
5. Can the user look through a sequence of screens in either direction?					
6. Can the user access a particular screen in a sequence of screens directly? (e.g. where a list or table covers several screens)					
7. In menu-based systems, is it easy to return to the main menu from any part of the system?					
8. Can the user move to different parts of the system as required?					
9. Is the user able to finish entering information (e.g. when typing in a list or table of information) before the system responds? (e.g. by updating the screen)					
10. Does the system prefill repeated information on the screen, where possible? (e.g. to save the user having to enter the same information several times)					
11. Can the user choose whether to enter information manually or to let the computer generate information automatically? (e.g. where there are defaults)					
12. Can the user override computer-generated (e.g. default) information, if appropriate?					
13. Can the user choose the rate at which information is presented?					
14. Can the user choose how to name and organize information which may need to be recalled at a later stage? (e.g. files, directories)					
15. Can users tailor certain aspects of the interface for their own preferences or needs? (e.g. colours, parameters)					

16. Are there any comments (good or bad) you wish to add regarding the above issues?

17. Overall, how would you rate the system in terms of flexibility and control?
(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 8: ERROR PREVENTION AND CORRECTION

The system should be designed to minimize the possibility of user error, with inbuilt facilities for detecting and handling those which do occur; users should be able to check their inputs and to correct errors, or potential error situations before the input is processed.

	Always	Most of the time	Some of the time	Never	Comments
1. Does the system validate user inputs before processing, wherever possible?					
2. Does the system clearly and promptly inform the user when it detects an error?					
3. Does the system inform the user when the amount of information entered exceeds the available space? (e.g. trying to key five digits into a four-digit field)					
4. Are users able to check what they have entered before it is processed?					
5. Is there some form of cancel (or 'undo') key for the user to reverse an error situation?					
6. Is it easy for the user to correct errors?					
7. Does the system ensure that the user corrects all detected errors before the input is processed?					
8. Can the user try out possible actions (e.g. using a simulation facility) without the system processing the input and causing problems?					
9. Is the system protected against common trivial errors?					

	Always	Most of the time	Some of the time	Never	Comments
10. Does the system ensure that the user double-checks any requested actions which may be catastrophic if requested unintentionally? (e.g. large-scale deletion)					
11. Is the system protected against possible knock-on effects of changes in one part of the system?					
12. Does the system prevent users from taking actions which they are not authorized to take? (e.g. by requiring passwords)					
13. In general, is the system free from errors and malfunctions?					
14. When system errors occur, can the user access all necessary diagnostic information to resolve the problem (e.g. where and what the fault is, what is required to resolve it)					

15. Are there any comments (good or bad) you wish to add regarding the above issues?

16. Overall, how would you rate the system in terms of error prevention and correction?

(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 9: USER GUIDANCE AND SUPPORT

Informative, easy-to-use and relevant guidance and support should be provided, both on the computer (via an on-line help facility) and in hard-copy document form, to help the user understand and use the system.

	Always	Most of the time	Some of the time	Never	Comments
1. If there is some form of help facility (or guidance) on the computer to help the user when using the system then: (a) Can the user request this easily from any point in the system?					
(b) Is it clear how to get in and out of the help facility?					
(c) Is the help information presented clearly, without interfering with the user's current activity?					
(d) When the user requests help, does the system clearly explain the possible actions which can be taken, in the context of what the user is currently doing?					
(e) When using the help facility, can the user find relevant information directly, without having to look through unnecessary information?					
(f) Does the help facility allow the user to browse through information about other parts of the system?					
2. If there is some form of hard-copy guide to the system (e.g. user guide or manual) then: (a) Does this provide an in-depth, comprehensive description, covering all aspects of the system?					
(b) Is it easy to find the required section in the hard-copy documentation?					
3. Is the organization of all forms of user guidance and support related to the tasks which the user can carry out?					
4. Do user guidance and support facilities adequately explain both user and system errors, and how these should be corrected?					
5. Are all forms of user guidance and support maintained up-to-date?					

6. Are there any comments (good or bad) you wish to add regarding the above issues?

7. Overall, how would you rate the system in terms of user guidance and support?
(Please tick appropriate box below.)

Very satisfactory	Moderately satisfactory	Neutral	Moderately unsatisfactory	Very unsatisfactory

SECTION 10: SYSTEM USABILITY PROBLEMS

When using the system, did you experience problems with any of the following:

	No problems	Minor problems	Major problems	Comments
1. Working out how to use the system				
2. Lack of guidance on how to use the system				
3. Poor system documentation				
4. Understanding how to carry out the tasks				
5. Knowing what to do next				
6. Understanding how the information on the screen relates to what you are doing				
7. Finding the information you want				
8. Information which is difficult to read clearly				
9. Too many colours on the screen				
10. Colours which are difficult to look at for any length of time				
11. An inflexible, rigid system structure				
12. An inflexible HELP (guidance) facility				
13. Losing track of where you are in the system or of what you are doing or have done				
14. Having to remember too much information while carrying out a task				
15. System response times that are too quick for you to understand what is going on				
16. Information which does not stay on the screen long enough for you to read it				

	No problems	Minor problems	Major problems	Comments
17. System response times that are too slow				
18. Unexpected actions by the system				
19. An input device which is difficult or awkward to use				
20. Knowing where or how to input information				
21. Having to spend too much time inputting information				
22. Having to be very careful in order to avoid errors				
23. Working out how to correct errors				
24. Having to spend too much time correcting errors				
25. Having to carry out the same type of activity in different ways				

SECTION 11: GENERAL QUESTIONS ON SYSTEM USABILITY

Please give your views on the usability of the system by answering the questions below in the spaces provided. There are no right or wrong answers.

1. What are the best aspects of the system for the user?

2. What are the worst aspects of the system for the user?

3. Are there any parts of the system which you found confusing or difficult to fully understand?

4. Were there any aspects of the system which you found particularly irritating although they did not cause major problems?

5. What were the most common mistakes you made when using the system?

6. What changes would you make to the system to make it better from the user's point of view?

7. Is there anything else about the system you would like to add?

APPENDIX D- LTDI checklist

Appendix 2: Evaluation Instruments

This appendix provides examples of the various types of instruments that can be used in the evaluation of learning technology and/or its implementation. The majority of these instruments are generic in their nature and could be used as they are in a variety of different evaluation exercises, a few are more subject/situation specific. ***You may freely copy and/or adapt any of these instruments for use in your own evaluation exercises provided this publication is acknowledged in any publications arising from their use.*** *We thank the originators for permission to reprint these examples in this publication.*

Some of these instruments are referred to in the chapters dealing with practical evaluation issues (as noted below), others are reproduced as examples of the types of instruments that may be used in evaluations.

Contents of Appendix 2

A. First Step Evaluation 'Checklist'

A comprehensive instrument to help guide a teacher through the process of reviewing a new piece of software. Provided by Jen Harvey. The use of this instrument is fully discussed in chapter 7.

B. Pre & Post Intervention Questionnaire

A two part questionnaire for establishing students' expectations and learning from using LT materials, discussed in chapter 12. Provided by Philip Crompton.

C. Program Questionnaire

A short instrument designed to elicit information from academics or students about the usability and content of a piece of LT material. Discussed in chapter 12. Provided by Philip Crompton.

D. Software Usability Evaluation

A short 'Likert scale' style questionnaire designed to elicit information from teachers or to summarise overall opinions about the usability and content of LT materials. Provided by Nora Mogey.

E. Student Confidence Log

A proforma of a pre and post confidence log questionnaire form as discussed in chapter 12. The specific concepts or skills descriptions have been left blank. Provided by Philip Crompton.

F. Observation Log

A blank photo-copyable proforma of the type of form used to collect information in an observation based evaluation exercise. Observation based evaluation is discussed in chapter 12. Provided by Philip Crompton.

G. Economics Profile Questionnaire

This questionnaire has been used in the evaluation of the TLTP product WinEcon and provides an example of the type of questions that might be asked to elicit information to build a profile of students' past experience and attitudes towards computers, as discussed in chapter 12. Provided by Philip Crompton.

First Step Evaluation 'Checklist'

A Guide for reviewing a new piece of software

Rather than considering a piece of learning technology in isolation it is important to think about the aims and objectives of a piece of software relative to its planned usage. For example, if one of the aims of a course is to encourage your students to relate different subject areas together you might feel that a piece of software does not quite fulfil your needs but the same software used in association with some paper based problems or as part or a group project might be far more effective.

This Checklist is designed to act as a guide to be used when reviewing a piece of software for the first time. Aspects of technical and systems support are not included but are also important considerations prior to the implementation of any of the learning technologies.

Part 1 - some points to consider as you work through

◊ **How easy is it to navigate your way through the software ? Does the package have a clear structure ?**

◊ **By what process is the user expected to learn about the subject as they work through the package ? Is this a good way of teaching this subject ?**

◊ **How do the learning strategies encouraged in this software fit into those of your existing courses and related teaching and learning materials ?**

◊ **What background knowledge or additional support would be required by your students if they were to use this package?**

◊ **Could you customise the package to suit your course requirements ?**

Part 2 - A step by step guide

This part of the checklist is designed to focus on different aspects of a new piece of software, as they might be encountered by a first time user. Therefore, you are asked to stop and review the package at certain points as you go through.

INTRODUCTION

Look through the introduction to the package

(this might be several screens or paper based material).

First Impressions

Are you presented with :	YES	NO
an explanation as to how you might use the package.	___	___
a series of topics in a menu format	___	___
some navigational hints about finding your way through the software	___	___
no introduction, you are straight into the first section	___	___
other (please specify)		

How much information is provided relating to the academic content of the package?	YES	NO
Is there an outline of how the material is structured or to be prioritised ?	___	___
Are the objectives of the package clearly laid out ?	___	___
Is there an indication as to level of prerequisite knowledge required ?	___	___
Are there links made to other relevant sources of information ?	___	___
Can you select from different sections according to difficulty ?	___	___
Do you know what to expect next ?	___	___
Other (please specify)		

Is some background knowledge of how to use a computer necessary before using the package ?
YES NO

Do you feel encouraged to work through the package ? YES NO

INTO THE FIRST SECTION

Move on through the next few screens and into one of the sections.

Level of User Control and Interaction

What level of interaction is required to move through the package i.e. are there a range of responses required or do you just press the return key?

High interaction				Minimal interaction
5	4	3	2	1

Can you control the rate at which you move through the package? YES ___ NO ___

From your current screen can you (*please tick*):

___	exit the programme	___	go back to the beginning
___	go forward a screen	___	go back a screen
___	move to a main menu	___	change to another section
___	save a copy of your work, so far, to file	___	print a copy of the screen
___	modify the material to suit your needs	___	refer to another section and return to the same point
___	use a calculator		
___	take notes into a scrapbook/notepad	___	other (please specify)

Which of the above functions would you say was the most important ?

Package Design and Layout

How much information is presented on each screen at one time ?

too much					too little
5	4	3	2	1	0

How clear is the on screen presentation i.e. graphics, text etc. ?

very clear					difficult to see
5	4	3	2	1	0

If graphics are used on screen - why are they used ?

Are they really necessary ?

If graphics are not used on screen - would supplementary graphics be beneficial ? YES NO

If so, can you suggest some possible sources e.g. software libraries, books etc. ?

MID - SECTION

Move forward a few screens

Prioritisation and presentation of information

How do you differentiate between the relative importance of pieces of information on a screen ?

How are key words in the text highlighted ?

Just by looking at a screen would it be possible to identify whether or not you were looking at a main, sub-menu or a help screen etc. ? YES NO

Have you any comments regarding the colour, font and the type style used so far?

How many icons appear regularly on the screens. Can you describe each of their functions?

Screen Icon	Function
1.	
2	
3.	
4.	
5.	

Provision of student support

If you require assistance, is a help facility available from your current screen? YES NO

If so, how is the help facility accessed ? i.e. via a menu or icon, highlighted words etc.

What kind of help is available ? (*tick those available*)

- ____ navigational
- ____ meanings of key words/information
- ____ alternative strategies
- ____ a glossary
- ____ other (*please specify*)
- ____ additional useful information
- ____ worked examples
- ____ reference to other material
- ____ content based material

Which of the above help facilities do you think is the most important ?

END OF FIRST SECTION

Move on to the end of this section or unit of the package

Matching strategies with objectives

As you worked your way through the section did you feel that :	YES	NO
the information was being related to your existing knowledge	___	___
you were being encouraged to think more about the subject area	___	___
your interest was being maintained through use of a range of strategies	___	___
the courseware was responsive to your own particular learning needs.	___	___
material was structured in a way that facilitated an overall understanding	___	___
any other general comments ?		

Feedback support for users

Was there any feedback provided as you were working through the software? e.g. on your rate of progress, performance etc. YES NO

If so, in what form did this take ?

Do you think this form of feedback is useful ? YES NO

Would you say that you felt encouraged to obtain feedback ? YES NO

Assessment

Were you assessed while you were working through the section? YES NO

If so when ?

Were exact responses required in the assessment ? YES NO

What would you say these assessments were testing ?

- ____ factual recall
- ____ your ability to guess
- ____ your ability to relate pieces of information together
- ____ other
- ____ your overall understanding of the subject
- ____ whether or not you had solved a problem
- ____ your creativity
- ____ the attainment of the package's objectives

Can you move to the next topic without completing the assessments ? YES NO

Did you obtain any feedback on your assessment responses ? YES NO

If YES, was the feedback only related to the wrong answers ? YES NO

Were your errors explained ?

Was the feedback constructive ? YES NO

Moving between sections

From the end of each section can you ?

- ___ exit the programme
- ___ move to a main menu
- ___ save a copy of your work to file
- ___ take notes into a scrapbook/notepad
- ___ go to some form of assessment
- ___ go to a more/less difficult section
- ___ stop and then return to this point later
- ___ go back to the beginning
- ___ change to another section
- ___ print out selected information
- ___ see the point reached in the programme and what is left to do
- ___ obtain a summary of the section
- ___ other (*please specify*)

WHEN YOU HAVE FINISHED LOOKING THROUGH

Was the information contained in the package correct in terms of factual content ? **YES** **NO**

Draw a diagram of how you perceive the structure of the package.

How long do you think the package is :

- ___ in terms of the number of screens ?
- ___ the student time to complete ?

How would you envisage this package might be used ?

- ___ remedial assistance
- ___ as part of a group project
- ___ in a class with supporting material
- ___ flexible learning
- ___ as part of a tutorial
- ___ other

Do you think the subject area benefited from the use of technology ? **YES** **NO**

Was the learning strategy reflected, appropriate to the learning objectives of the package? **YES** **NO**

Does the software test that its objectives have been attained? **YES** **NO**

IN SUMMARY

Having reviewed the package how would you rate the package in terms of :

Usability

excellent 10	9	8	7	6	5	4	3	2	poor 1

What kind of additional support would be required for a student to work through the package by themselves ?

Layout

excellent 10	9	8	7	6	5	4	3	2	poor 1

Is it possible to customise the package to suit your course requirements ?

Academic Content

excellent 10	9	8	7	6	5	4	3	2	poor 1

Could this package be used as part of one of your courses ?

If so - HOW ?

If not - WHY NOT ?

Attainment of Learning Objectives

excellent 10	9	8	7	6	5	4	3	2	poor 1

Which of the following best describes the type of approach you were encouraged to adopt as you worked through the package ?

- ___ deep approach (looking to an overall understanding of the material)
- ___ strategic approach (driven towards high attainment i.e. not to make mistakes)
- ___ surface approach (minimal interaction, no need to understand material)

How do the learning strategies encouraged in this software fit into those of existing courses and related teaching and learning materials?

Pre and Post Intervention Questionnaire

Date:

Lecturer:

Hardware:

Student Name:

Department:

Evaluator:

Program:

Time:

SECTION A: Pre-Program Questions

Students should complete this section before using the software.

1. What are you expecting to learn from today's session ?

2. (a) Are there any parts of this subject area which you have difficulty with ?

2. (b) How do you hope that this session will help you ?

Do not answer any of the other questions until after you have used the software.

SECTION B: Post-Program Questions

Students should only complete this section <u>after</u> using the software.

1. What do you feel was the most important thing that you learned today ?

2. Look back at what you wrote for the Pre-Program Questions and note down below:

(a) What you have learned and compare this with what you had hoped to learn

(b) what you did <u>not</u> learn that you had hoped to:

(c) was there anything that you learned that was unexpected:

SECTION C: Software

1. To what extent do you agree with these descriptions of the computer program?

(1 = Strongly Agree, 2 = Agree, 3 = Neutral, 4 = Disagree, 5 = Strongly Disagree)

	Please circle one				
Easy to use	1	2	3	4	5
Enjoyable to use	1	2	3	4	5
Provides good support for the exercise	1	2	3	4	5
Provides good advice on how to work through the material	1	2	3	4	5
Helps you learn about the subject	1	2	3	4	5
Fits well with the rest of the course material	1	2	3	4	5
Well worth the time spent on it	1	2	3	4	5
It would help me to revise the subject	1	2	3	4	5
I would use it, in my own time, again	1	2	3	4	5

Please add any other comments you wish

2. This program is meant to help you improve your knowledge of /skill in ____________ (the subject).
Could you please comment on any improvements you are aware of in:

(a) your general knowledge of ____________ (the subject)

(b) the way in which you approach ____________ (the subject

(c) the way in which you might apply what you have learned in the future

Thank you for your time and effort in completing this questionnaire

Program Questionnaire

Please tick the box [✓] for those characteristics of the program you feel are well designed, and put a cross [X] against those you feel need improving, together with a comment.

1. Navigation

Clear what options are available to you [ ] ____________

Easy to get where you wanted to go [ ] ____________

Easy to find out what you have completed already and what is still to be completed [ ] ____________

2. Interface

Easy to understand functions, menus, icons etc. [ ] ____________

Terms and procedure for navigation are consistent [ ] ____________

Screen easy to read, pleasing to look at [ ] ____________

3. Interaction

Presentation sufficiently informative [ ] ____________

Presentation interesting [ ] ____________

Content is - challenging [ ] ____________

right level for the course [ ] ____________

Feedback provided enabled you to learn better [ ] ____________

Clear about what you had to do [ ] ____________

Clear what you have achieved [ ] ____________

4. General

Please add any other comments you would like to make about the computer program.

Thank you for your time and effort in completing this questionnaire.

Software Usability Evaluation

Package:

	Strongly agree					Strongly disagree
I found this package easy to use	1	2	3	4	5	6
I would like to use this package again	1	2	3	4	5	6
I would feel confident using this package with students	1	2	3	4	5	6
I think my students would enjoy using this package	1	2	3	4	5	6
I think my students would benefit from using this package	1	2	3	4	5	6
I think that my students would find this package easy to use	1	2	3	4	5	6
I think that the material in this package would be challenging for my students	1	2	3	4	5	6
I think that this package contains material similar to my course content	1	2	3	4	5	6
I think that this package contains material that is desirable pedagogically	1	2	3	4	5	6
I think that this package could be a useful tutorial aid	1	2	3	4	5	6
I think that this package could be a useful lecture aid	1	2	3	4	5	6
I think that this package could be used as part of students' directed study	1	2	3	4	5	6
I think that this package could be useful for students encountering difficulties with the subject	1	2	3	4	5	6
I think that this package could be a useful revision aid	1	2	3	4	5	6
I think that this package could easily be integrated into my course	1	2	3	4	5	6
This package could be useful for more than one course	1	2	3	4	5	6

Student Confidence Log

If you have not as yet had a lecture on this course material, you might think it inappropriate to say anything at all about your understanding of the topics below before working through the program. However, completing this form will give us baseline data for describing any changes in confidence levels which may be found following the use of this program.

Please indicate by ticking [✓] in the relevant box, how confident you feel about your understanding of the concepts or development of the skills listed below.

Date: **Department:**
Lecturer: **Evaluator:**
Hardware: **Program:**
Student Name: **Time:**

A: Pre-Program Confidence Levels

Concepts / Skills	Very Confident	Confident	Some Confidence	Little Confidence	No Confidence

B: Post-Program Confidence Levels

Do not complete this table until after using the program.

Concepts / Skills	Very Confident	Confident	Some Confidence	Little Confidence	No Confidence

Observation Log

(page -________/___1_)

Date:	**Department:**
Lecturer:	**Evaluator:**
Hardware:	**Program:**
Student Name(s):	**Time:**

CODE for interactions:

TS	=	Teacher to Student	**SS**	=	Student to Student
GD	=	Group Discussion	**SC**	=	Student to Computer
GC	=	Group to Computer	**SI**	=	Student Individually

#	Time	Code	Comment	Note

Observation Log (page -________/_____)

CODE for interactions:

TS	=	Teacher to Student	**SS**	=	Student to Student
GD	=	Group Discussion	**SC**	=	Student to Computer
GC	=	Group to Computer	**SI**	=	Student Individually

#	Time	Code	Comment	Note

Economics Profile Questionnaire

This questionnaire has been designed to gather information about the first year Economics course.
Please try to answer all questions honestly.

Please tick the boxes as appropriate

1. Sex a) Male ☐ b) Female ☐

2. What age group are you? a) 27 or under ☐ b) 28 and above ☐

3. Have you studied Economics before e.g.Highers? a) Yes ☐ b) No ☐

4. How would you rate your Mathematical ability?

a) University level ☐ b) A Level/Higher ☐ c) 0 Level ☐ d) Beginner ☐

Below are some statements about computers. To what extend do you agree with them?

6. "Given a little time and training anybody could learn to use a computer."

a) strongly agree ☐ b) agree ☐ c) neutral ☐ d) disagree ☐ e) strongly disagree ☐

7. "I find a computer difficult/complicated to use."

a) strongly agree ☐ b) agree ☐ c) neutral ☐ d) disagree ☐ e) strongly disagree ☐

8. "Computers isolate you from other people."

a) strongly agree ☐ b) agree ☐ c) neutral ☐ d) disagree ☐ e) strongly disagree ☐

9. "I think computer literacy will make me more employable."

a) strongly agree ☐ b) agree ☐ c) neutral ☐ d) disagree ☐ e) strongly disagree ☐

10. "I am afraid of looking silly if I make a mistake while using the computer."

a) strongly agree ☐ b) agree ☐ c) neutral ☐ d) disagree ☐ e) strongly disagree ☐

11. Name and Student No.

Please return the completed form to ____________________, by ____________________.

Thank you very much for your time and effort.

If you would like to take part in a research project into evaluating the economics package WinEcon, please make sure that you have filled in your name and student no. There will be a payment make to those students willing to assist in this research project.

APPENDIX E- The outline of the TUP system

The following text serves as an outline of the TUP tool and the following evaluation. It gives further explanations and references to the TUP checklist. These notices should be used altogether with the checklist questions in order to give evaluator a help.

Proposed informational model

The TUP tool should gather various information about users, environments being evaluated and about the evaluations, in the TUP model called the reviews. From the previous chapters it follows that the TUP system has to inquire the user's background, institution and professional field. Further, concerning the environments under evaluation, the TUP system collects identification and description of an application, its purposes and destinations, learners' target level, resources by which the environment is served, authors, language of an application and so on. The exact set of information which is needed to collect is the subject of an additional research.

Additional notices concerning the TUP checklist

In the chapter 6.2 the TUP checklist has been established. However, in order to facilitate the evaluators' performance the following explanations should be used altogether with the references to access more comprehensive information, if needed and available. The evaluators has to be supported by the system. The keywords in the questions should be highlighted and the further explanations offered.

The following table contains the examples of the explanations and references of the keywords used in the TUP checklist questions, which might cause the difficulties in the comprehension.

Aspect	Keyword	Explanation	References
Technology			
	External resources	The tool enables to use e.g. textual, graphic files, or databases created in other environments. These files can be freely imported and used by the environment and exported to be used by the external applications.	

Aspect	Keyword	Explanation	References
	Additional software	In order to run the environments, other software packages have to be installed which might or might not be delivered with the environment.	
	Components	E.g. various libraries (e.g. graphical engines as OpenGL), language interpreters, libraries.	http://www.opengl.org
	Open source	The package is freely redistributive, of no cost, and does not restrict any other software.	http://opensource.org/docs/definition.php
	Additional equipment	E.g. an equipment or hardware which are not usually available and have to be purchased in order to complete function (e.g. joysticks, tablets).	
	Privacy	E.g. in the network environments users have their separate accounts.	
	Security	The environment protects the sensitive and confident information, e.g. by access rights.	
	Fault	E.g. it is possible to safely save user's work if the fault occurs.	
	Diagnostics	E.g. the diagnostic tool checks the environment for possible defects.	
Usability			
	Cognitive load	E.g. users of the tool are forced to process the information (not related to the learning) from the previous interaction stages in order to complete the future dialogues by which their attention is split	
	Level of interaction	Different user groups embodies different characteristics, e.g. the level of abstraction in tool for the children differs from the software for adults.	

Aspect	Keyword	Explanation	References
	Feedback	E.g. the tool responses in the reasonable time so the users do not feel uncomfortable.	
	Search	E.g. (in multimedia databases) the seach facility is offered in order to access easily the required information.	
	Aesthetic	The user interface is visually pleasing.	
	Legible	E.g. the size and the color of the text is appropriate to the light conditions.	
	Organization	E.g. related figures or images are not placed separately and thus offers easy confrontation.	
	Consistency	E.g. the sequences of actions are same in all parts of the tool, the tool uses same terminology and layout throughout the interaction.	
Pedagogy			
	Authentic context	E.g. the context is not artificial and naturally corresponds with the real life.	
	Roles	E.g. in a groupware, the roles of managers, teachers and learners are distinguished by the environment.	
	Self-adaptable	The tool persistently maintains a model of users' knowledge and reflect their growth.	
	Cultural issues	E.g. the tool presents information fairly to the culturally different users groups.	
	Trust	E.g. there are no logical defects in the environment which would lead to the misinterpretation.	
	Selection of learning goal	I.e. the set of learning goals defined by the environment	
	Multiple representation of knowledge	There are offered more views to the learning matters and knowledge.	

Aspect	Keyword	Explanation	References
	Secure process management	E.g. the learners have an access only to the materials regarding the course of learning	
	Organizational requirements	E.g. the tool follows the grading system of an organization	
	Share learning materials	E.g. teachers can exchange learning materials, artifacts or the results of learning	
	Learning artifacts	I.e. objects created during the learning processes and cooperation, e.g. texts, pictures, link collections.	
	Tailorization	The managers, teachers or users are allowed the specify or adjust the interface's means according to the learners.	