E-Learning and Self-Assessment for Hands-On Labs in Higher European Education

Fritz Laux
Fakultät Informatik
Reutlingen University
D-72762 Reutlingen, Germany
fritz.laux@reutlingen-university.de

Thomas Connolly
School of Computing
University of the West of Scotland
Paisley PA1 2BE, UK
thomas.connolly@uws.ac.uk

***Abstract*—“Learning by doing” in Higher Education in technical disciplines is mostly realized by hands-on labs. It challenges the exploratory aptitude and curiosity of a person. But, exploratory learning is hindered by technical situations that are not easy to establish and to verify. Technical skills are, however, mandatory for employees in this area. On the other side, theoretical concepts are often compromised by commercial products. The challenge is to contrast and reconcile theory with practice. Another challenge is to implement a self-assessment and grading scheme that keeps up with the scalability of e-learning courses. In addition, it should allow the use of different commercial products in the labs and still grade the assignment results automatically in a uniform way. In two European Union funded projects we designed, implemented, and evaluated a unique e-learning reference model, which realizes a modularized teaching concept that provides easily reproducible virtual hands-on labs. The novelty of the approach is to use software products of industrial relevance to compare with theory and to contrast different implementations. In a sample case study, we demonstrate the automated assessment for the creative database modeling and design task. Pilot applications in several European countries demonstrated that the participants gained highly sustainable competences that improved their attractiveness for employment.**

***Keywords*—learning by doing, virtual laboratory, hands-on lab, e-learning concept, self-assessment, e-grading.**

I. Introduction

Effective knowledge transfer at Higher Education (HE) institutions and Vocational Educational Training (VET) should be tailored to the needs of its clients. VET, in contrary to HE, comprises of all non-academic professional education and training provided either by state or private organizations, e.g., in-company training. It is characterized by teaching practical skills that are needed in the daily work of employees, e.g., learning how to use a specific software product like a relational database product or how to normalize a relation.

Employees are highly motivated to acquire new skills but are often hindered to follow a scheduled training program. Students face a denser curriculum due to the Bologna process with a high degree of optional courses whose schedules and prerequisites are not aligned. Therefore, it is essential to provide self study courses with small module sizes to enable the participants to learn in their spare time at their own pace. In addition, in financially difficult times, knowledge transfer should be highly scalable in terms of costs. E-learning offers this capability but has the difficulty of keeping motivation high and to impart in-depth knowledge. Another challenge is the automated assessment of higher level understanding, a key requirement for large scale on-line courses. In engineering disciplines, e-learning has to deal with skills how to use commercial software or other technical devices.

As consequence, e-learning has to solve a multidimensional problem. Laux et al. [1] identified and presented at ICIW 2012 the following problem dimensions:

- content granularity
- theory level
- technology
- pedagogy
- assessment
- competence level

This leads to the following challenges. The learning content needs to be sliced into “digestible” portions while keeping the necessary context. Technological reality has to match and sometimes contrast with the theoretical underpinning. Technological aspects in Information and Communication Technology (ICT) are of particular importance to empower students and employees for a competitive labor market. Aiming for technological competence will stimulate the secondary motivation of the learners.

In our case study, which extends the results of [1], we focus on one of the most important areas in ICT competency for information management professionals: *database management systems* (DBMS). Databases are now the underlying framework of information systems that have fundamentally changed the way organizations and individuals structure and handle information.

Two crucial competences within the database domain are, how to structure efficiently a database and how to correctly process the data. For example, in the case of a banking application the database has to process the financial transactions correctly and reliably under any circumstances. This requires a sound understanding of the theory of transaction management and practical skills of software products at the same time. Such a highly specialized knowledge cannot be only theoretically taught neither could it be trained only by examples like a cookbook.

A second example is the database design, where profound understanding of the database model and the business domain is necessary in order to create a database structure that is efficient, resilient, and flexible for future change. It requires a clear view of the important entities of the business and its requirements, but also leaves room for abstraction and individual viewpoints. This makes the assessment of a data model difficult and subjective.

Both examples are scenarios for our e-learning concept with hands-on labs where we demonstrate how to teach effectively theory combined with hands-on labs for practical skills and problem solving competence.

A. Structure of the Paper

With the following overview on related work in cognitive science the context for our learning theory will be settled. In Section II, we point out the pedagogical requirements, the modularization constraints dictated by the learning objects and the tension between industry demands and long term knowledge for the students. This clarification is used in Section III as criterion for developing a unique reference model for the example learning object *database systems*.

With the help of Bloom's Taxonomy, we identify different knowledge levels in Section IV and show how to assess the higher levels of understanding.

Section V describes the supporting technology, in particular, the environment for the hands-on labs. We evaluate the e-learning reference model and discuss our findings during pilot runs of the learning modules in Section VI. The paper ends with a conclusion and ideas for future work.

B. Related Work

E-learning is a promising research subject and there is an abundance of publications on the foundation of on-line learning (e.g., [2][3][4][5][6]) as well as on its problems. For instance, the decreasing motivation was described by Prenzel [7] and Paechter et al. [8]. This is also confirmed by our own experience with e-learning.

According to constructivism [9] the learner generates knowledge by individual experience (radical constructivism [10]) or by social interaction within a cultural context (social constructivism [3]). As consequence, knowledge should be acquired by the learner in authentic situations that keep motivation high [11]. Connolly and Begg [12] report similar experiences and recommend teaching database analysis and design in a problem based environment.

Communication with fellow students and team work are also factors that support learning motivation [5]. This makes a communication and collaboration tool an indispensable ingredient of an e-learning system.

Multimedia support through e-learning systems is an enabler for flexible and scalable HE and VET, but is no guarantee for a successful on-line course. Critical voices raised the issue of superficial and routine knowledge that may easily be transferred. This knowledge refers to the cognitive domains one (remember), two (understand), and three (apply) of the revised Bloom's taxonomy [4].

In this taxonomy, Bloom [13] and Anderson [4] distinguish six cognitive levels

1) remember
2) understand
3) apply
4) analyze
5) evaluate
6) create.

For a deeper understanding the learner should acquire the higher cognitive levels. But, profound insights (analysis, synthesis/creation, and evaluation in Bloom's categories) are difficult to convey with a computer based learning environment as the study conducted by Spannagel [14] reveals. Krathwohl [15] gives a concise comparison of the original and revised taxonomy also stressing that the new taxonomy has four dimensions - factual, conceptual, procedural, and meta-cognitive knowledge.

The first three cognitive levels include the substance of subcategories of knowledge in the original framework. New is the meta-cognitive knowledge, which provides a distinction between knowledge and cognition in general as well as awareness of and knowledge about one's own cognition. This was not widely recognized at the time the original scheme was developed but is now of "increasing significance as researchers continue to demonstrate the importance of students being made aware of their meta-cognitive activity, and then using this knowledge to appropriately adapt the ways in which they think and operate" [15].

The Bloom/Anderson knowledge taxonomy was chosen because it structures knowledge according to the level of understanding and it fits well into the evaluation of skills-related learning. It is possible to distinguishes between the ongoing formative assessment (giving feedback about the student's performance or the assessment of educational materials during the course) and the summative assessment (evaluation at the end of the instructional cycle). So far, there are a couple of articles regarding e-learning assessment. Richards and DeVries [16] use the formative evaluation to dynamically monitor the learning activities in order to improve its course design. Their work focuses on the instructional design methodology and uses embedded questionnaires for the feedback. Velan, Jones, McNeil and Kumar [17] show in detail, how continuous online formative assessments helped medical science students to achieve better grades.

Experiences with summative assessment are reported by Chew, Jones and Blackey [18]. They introduced a range of of online assessment tools, such as electronic submission, partial tutor-intervention or a complete end-to-end computer-assisted assessment, at their university. As result from their experiences they recommend seven practices to follow and eleven to avoid. All these practices concentrate around organizational or technical aspects and how to gain a positive attitude towards e-learning and e-assessment. The real assessment work and feedback was based on closed questions that cover only the lower levels of understanding.

An approach for automatic marking of short essays from graduate students in computer science is described by Thomas, Haley, deRoeck and Petre [19]. They use a technique called

Latent Semantic Analysis (LSA) to infer meaning from a natural language text. This type of association analysis was used for marking free-form short essays. LSA produced - depending on the question - between 83% and 66% similar marks to an experienced human marker. For only one question out of six a significant statistical correlation was found.

A more promising result was achieved be Higgins and Bligh [20] when they applied computer based formative assessment in a diagram based domain. Compared to LSA this is understandable as the rules for diagram based models are far simpler and more precise by definition than a natural language. In their setting they assigned to a self programmed diagramming tool an explanation and feedback text for every diagramming rule. It was possible to add composite rules or features tailored for a specific task and context. From this approach raises a problem based on the fact, that each feature is assessed exactly once. It turned out that “several equally valid model solutions with slightly differing, mutually exclusive, features” [20] could not be assessed appropriately.

In our paper we will use state of the art products, like Oracle SQL Developer, MS SQL Management Studio, or Aqua Data Studio Entity Relationship Modeler. We do not want to insist on a preset model solution. The e-assessment process should allow to assess innovative, original and unexpected design solutions.

Problem based learning helps students to keep motivation high. But it seems difficult to ensure that theory and the necessary abstraction are drawn from an example. There are concepts that try to overcome these problems with the use of multimedia technology [6]. Blended learning, for example, tries to combine classroom learning with e-learning [2, chap. 10 and 29]. Classroom teaching can provide for theory and the e-learning session practice the knowledge in the form of exercises or experiments. We apply this technique for our virtual laboratory workshops described in Subsection III-C. This hybrid learning does not ensure sustainable and deep understanding, but, a well thought concept may help to convey deep insights as Astleitner and Wiesner [5] point out.

Our concept aims further: it contrasts and reconciles theory with the reality of commercial software products. This is important because software professionals and experts need the competence to verify the real behavior of a database system for instance and compare it with the theory. As a consequence real products are necessary as training tools and for assessment. No learning concept, so far, has tried to deal with the peculiarities of commercial graphical modeling products and provide an automated assessment of the resulting model.

II. Problem Description and Contribution

The goal is to provide a highly modularized e-learning environment for the specific theoretical and practical needs of HE and VET in the domain of ICT. For the proof of concept we have chosen the material produced during two EU funded projects: DBTech Pro (funded by the Leonardo da Vinci Programme) and its successor DBTech EXT (funded by the EU Lifelong Learning Programme).

The first project, DBTech Pro (http://myy.haaga-helia.fi/~dbms/dbtechnet/project2002-05_en.HTML), started with a survey to find out the needs of ICT-industry with regard to database technology competencies. We identified important knowledge areas of *database systems* and syllabi for course units. Based on these findings a framework and syllabus for the essential knowledge areas was developed, covering the related database standards, specifications, technology trends and understanding of the mainstream DBMS products. Course modules, including laboratory exercises, were developed and pilot courses executed in all five partner countries (Finland, Greece, Germany, Spain, and United Kingdom). The experiences from the pilot tests have been evaluated by students and experts.

The successor project DBTech EXT (http://myy.haaga-helia.fi/~dbms/dbtechnet/DBTechExtDescr.pdf) was formed by 7 universities, 3 VET institutes, and 1 industry representative. It extended the work from the previous project with a learning reference model with knowledge taxonomy and laboratory environment described in Section III. The model integrates instructional, active, and constructive learning concepts, which are applied as appropriate by the learning subject. Focus was on the in-depth knowledge with hands-on labs for learning by doing and verifying theories based on e-learning technology. Example topics have been database design, transaction processing, and data mining. Again, all courses and hands-on labs have been evaluated by the participants and teaching experts. More information about both projects may be found at http://www.dbtechnet.org.

From a pedagogical view, we identify the following requirements for a successful on-line course:

- self controlled learning
- authentic problem oriented learning
- most importantly, cooperative learning
- self assessment
- feedback and evaluation

Self controlled learning is important because of the above mentioned time constraints and with regard to different precognition of the learners. For a high motivation it is necessary to pose authentic real world problems and to let the learners solve them as a team [12]. This requires state-of-the-art software as used in industry.

Cooperative learning has two positive effects, one for the learner and one for the teacher, as follows. Communication among the students and working in groups keep motivation high and yield better learning results. It strengthen personal confidence of the learner through the positive feedback from the team. From the teacher’s view the communication provides feedback on the effectiveness of the teaching and exercise material. The assessment of solutions for real-world problems is easier to justify and the acceptance from the students is more likely. In addition, communication among students reduces teacher intervention.

Employers demand key competences and skills that are predominately conveyed by cooperative learning:

- ability to solve real world tasks (problem solving)
- knowledge about state-of-the-art technology

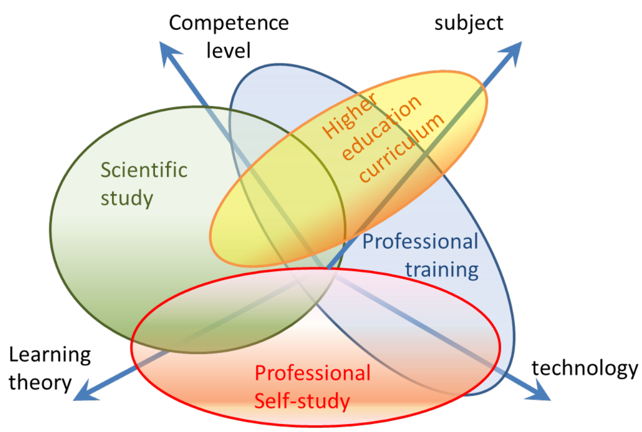

Fig. 1. Learning modules and its emphasis on target groups

- social skills, so called *soft skills*

Employees and students have increasing interest in learning skills that give a fast and easy to see return on their learning investment in form of directly applicable knowledge at their working place. This validates the first two requirements. Problem solving is a daily task at every workplace. Software problems most often can only be tackled successfully with state-of-the-art knowledge about technology. Social skills are indispensable for highly demanding ICT jobs [21] that involve cooperation and collaboration of many people.

In addition to the requirements mentioned above, the teaching units (modules), which we want to create, need to comply with the taxonomy of that domain, which defines how to decompose the content along the aspects:

- competence level
- subject area
- technology

Decomposing the content along the competence level provides different degrees of detail in line with target competencies and work profile. Students of HE institutions prefer a different learning concept than in VET courses. The latter have a tighter time schedule with less time for reflection of theoretical issues than HE students.

So, apart from the challenging content we tried to address all of the above requirements by dividing the learning content so that it can be combined and composed in multiple ways. Depending on the target learning group we produced different learning modules with specific consideration of the above described aspects. Figure 1 visualizes the emphasis of different target learning groups. For example, a module for professional training typically focuses on technology with a high level of competence and skills. Higher education curricula are less concerned about technology but concentrate on the subject theory. In contrast, professional self study needs to take special care of the didactic aspects as this is crucial for enabling self study and keeping the motivation high.

A. *Contribution*

The contribution of this paper consists of an integrated learning model for e-learning addressing the needs and constraints of HE and VET. For each learning unit the most appropriate learning theory was applied. Furthermore, the framework solves the problem of content modularization. Exemplary e-learning material that was used in multiple pilot runs proved the usefulness of this approach and resulted into more sustainable knowledge compared to traditional university teaching. The main advantage lies in the practical skills acquired using real DBMS products in the hands-on labs. The necessary lab environments are easy reproducible and provide full control of license restrictions. A major achievement of our approach is that we are able to assess automatically truly original and innovative data models produced by mainstream modeling tools. This is demonstrated with a commercial graphical Entity-Relationship diagramming and modeling tool.

III. The Reference Model

The reference model applies several learning concepts reflecting the different aspects and challenges presented in the previous section. The interrelation of these requirements make it difficult to optimize the learning concept. For better understanding we treat the dimensions content, lab environment, and project work separately and discuss the global optimization in Subsection III-E at the end of this section.

A. *Knowledge Taxonomy*

It is common to define a syllabus for the learning content. Structuring the syllabus results in a knowledge taxonomy of the teaching domain. From this structure we are able to deduce pre-requisites, identify learning elements, and designate learning outcomes. Structuring the teaching domain along the knowledge levels defined by Bloom [13] and Anderson [4] helped us to modularize the content according to knowledge depth and to provide teaching units for different target groups.

As an example, Figure 2 shows an extract of the DBTech database taxonomy [22] depicting the comprehension levels. Based on this layering we were able to deduce pre-requisites for every learning unit. For instance the unit *data modeling* (see Silberschatz et al. [23]) requires knowledge about the *relational, hierarchical*, and *network model*.

The knowledge levels are exemplified with the *transaction management*. On the lowest comprehension level it is sufficient to *recall* that transaction management coordinates transactions in a way that no undesired results may occur including the enduring protection of transaction results. For the second level the learner *understands* what this protection and coordination means in terms of the ACID[1] properties [24] and for level three the learner knows how to apply this knowledge. So far, it is not necessary to know how these properties are achieved by the transaction management. For the higher knowledge levels non functional aspects such as performance of different implementations of the ACID properties and its approximations are important. In order to *analyze* two concurrency mechanisms it is necessary to know their conceptual differences. Implementation knowledge is helpful to *evaluate* different mechanisms in terms of performance. For level six (*create*), the highest knowledge level of the Bloom/Anderson taxonomy, the learner needs a creative idea beside some experience in the implementation of a transaction management system.

[1]ACID stands for the characteristic properties of a transaction: Atomic, Consistent, Isolated, and Durable

	DBTech Pro Framework Reference courses and Topics	European Qualification Framework	IEEE/ACM CS2008	EUCIP initiative of CEPES	ACM AIS AITP IS 2002	BCS Professional Examination 2003	SweBOK
9							
10	**Principles of Database Systems** (Level: Introduction, - obligatory)	Knowledge Level	CS2:IM1, IM2	EUCIP core IM knowledge		BCS Diploma (D) - Database Systems	CS 5 IM
11	Database Principle	Level 5	IM2.1, IM2.2	3.2.2.3		D 5.2	
12	Concepts of Database Systems and Environments	Level 3		3.2.2.1 3.2.2.4		D 5.1	
13	User roles	Level 3		3.2.2.4			
14	Ansi/Sparc Architecture	Level 3					
15	Conceptual models: ER and UML	Level 3					
16	Data Modeling	Level 5	IM2	3.2.2.2			
17	- Relational Model (RM) - Hierarchical (for XML) - Network Model (for ODBMS)					D 5.4 (RM)	
18	Relational Theory	Level 3					
19	- Relational Algebra	Level 2		3.2.2.6		D 5.4	
20	*Normalization*	Level 4				D 5.4	
21	Object-oriented Model	Level 5					
22	ODMG Standard	Level 5					
23	SQL Basics	Level 5		3.2.2.7			
24	QBE	Level 5					
25	Security	Level 3					
26	Transaction Processing	Level 4					
27	*Transaction Principle*	Level 4				D 5.6	

Fig. 2. Mapping of DBTech Database Taxonomy to other CS curricula (partial view) [22], sheet 2

B. Virtual Laboratory

The most important component of our e-learning model is the "learning by doing". The concept of "learning by doing" became known in pedagogy through the work of Comenius [25]. From the perspective of developmental biology learning by doing is known even from animals [26] and experimenting (the systematic learning by doing) is fundamental in the development of the *homo sapiens* [27][28].

The psychomotoric learning keeps motivation high and supports a high degree of practical skills needed by companies. Moreover, the endurance of knowledge is much better and profound than without hands-on labs. Small, practical exercises and experimenting prepares the way for problem based learning.

In the case of ICT we have to deal with sophisticated, interdependent software systems like database management systems, application servers, data warehousing, On-Line Analytical Processing (OLAP) systems, or business intelligence suites. A student would need excessive time to install and set up the lab environment. This is unfeasible, considering only the risk that the system might be (unconsciously) misconfigured.

Another obstacle could be the different hard- and software equipment that might impede the installation of a certain product. But a realistic scenario consists of a suite of software products that must be configured in such a way that the programs can work together. For instance, an ERM-Modeling tool should produce a schema output that can be executed on the target database. The application server needs to co-operate with the database system and the web server. The only technical solution that actually works without problems is the virtualization technology. It provides a lab environment independent of the physical computer, which can be copied across the Internet to the learners' computers. Even if a student accidentally damages the virtual system he can reset it to its original state. He is also able to save his results in a snapshot and continue later or at a different computer. There exist virtual image capturing and playing software that is freely available.

C. Virtual Laboratory Workshops

The technological complexity of the Virtual Laboratory makes it necessary to provide detailed, step-by-step tutorials for experimenting. The step-by-step tutorial is illustrated by screen-shots or from some complex procedures exist video sequences (animated screens). With this support the students can work through the lab experiments and design tasks without instructor intervention. Review questions allow the students to check their understanding and quizzes and open problems motivate for further investigations.

In order to make the learning more effective, we decided to use blended learning techniques and gather students for live workshops using the virtual laboratory. One trainer for about 10 students was sufficient to answer questions or to provide help with the virtual lab environment. Blended learning turned out to be effective for larger assignments or project work.

Between workshop sessions and for remote participants Skype telephone and remote assistance via web conferencing tools have been available. This allowed interactive help directly with the laboratory environment. The teacher could take over the students screen and demonstrate how to overcome a blocking situation.

The students had to submit their deliverables electronically via the e-learning platform for grading. The e-learning system was also heavily used as a discussion board, for self-assessment (see Section IV), and for feedback from students. The feedback was used to improve the presentation of the learning modules.

D. Project Work

While teaching theory in a didactic way and practicing or verifying the transferred knowledge in hands-on labs there is no guarantee that the students really acquire a problem solving competence. It is necessary to combine different knowledge pieces, then abstract and apply them as a whole. This systemic knowledge gap can be easily seen when students know about the ACID properties [24] of a transaction, but cannot relate a real world problem like the concurrent on-line reservation of flights with the concurrency issue. In the lab, such situations can be explored with real products and it is possible to test the behavior of the software in case of concurrent clients.

Moreover, students might be skilled in technological aspects of application servers but do not realize the danger of a compromised transaction due to technological tricks like pooled connections or disconnected components.

To ensure problem solving competences beyond technical issues students have to develop their ability to work in teams, manage tasks, organize releases and orchestrate different versions. All this knowledge can be learned from real world projects.

E. E-Learning Model

We believe it is best to decide from the learning content, which learning concept will be best suited for a specific content. The e-learning model we present integrates different learning concepts (see Issing [2]):

- Learning as behavioral modification for *practical skills* and verification of the theory
- Learning as active information processing using assimilation and accommodation processes to build a mental model of the *theory*
- Learning as construction of knowledge used for problem based learning as in *project work*

All these concepts are used in an integrative way in order to get the most effective results in terms of applicable knowledge and profound cognition that enable abstraction and problem solving to a large extent.

The design of the e-learning model (see Figure 3) starts with structuring the learning area guided by a taxonomy. The area is partitioned with a minimum of dependencies and each chunk of learning content is represented in a theory unit together with examples and demonstrations. Hands-on experiments help the students to verify and reflect the theory. At the same time they memorize situations and learn the necessary skills that help them to produce a solution for a related problem.

Examples and demonstrations explain the theory, making it easier to understand and familiarize with the concepts. Hands-on experiments motivate and stimulate students to reflect the theory. Examples provide the students with analogous situations that can be applied and abstracted in the project work.

The concrete real world problem forces the students to abstract from examples and construct a model of the problem world in order to find a solution. Assessment of knowledge, skills, and solutions is driven by the expected competences that the learner should gain.

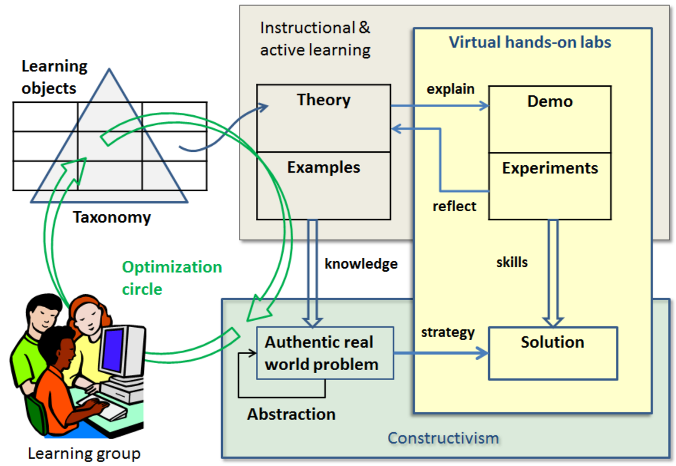

Fig. 3. E-Learning Model Overview

The interrelation of all these elements provided in a virtual lab environment with theory units, examples, and experiments are as shown in Figure 3. The global optimization task for the teacher is to select the learning objects, demos, examples, and experiments that lead to knowledge and skills necessary for solving the authentic real world problem and to put together all aspects in balance with the target learning group.

IV. Assessment

Knowledge levels 1 - 2 of Bloom's Taxonomy [13][4] seem easy to assess with multiple choice questions. Level 3 could be checked with a cloze test. The cloze text should refer to application knowledge that requires students to fill the gaps in the text with words such that the text gives meaningful instructions for a task. In case of programming language skills the assessment of level 3 knowledge could be automated by running the submitted source code against a compiler and executing the result. This is what we usually do when we assess and grade the students' knowledge of SQL. This procedure has the potential for self-assessment. The student enters the SQL-Query into a database system. Then, parser and syntax checker verify that the query is syntactically correct. If there is a syntax error, the student gets feedback in the form of a descriptive error message. When there is no syntax error the query is executed and the output can be compared with the expected result.

If the e-learning system supports a programming interface to the database system, then the assessment can be automated. This approach is used by SQL Training Programs like SQLZoo (http://sqlzoo.net/) by Andrew Cumming [29], SQL Tutor (http://www.cosc.canterbury.ac.nz/tanja.mitrovic/sql-tutor.html) by Antonija Mitrovic [30], the GNU SQLTutor (http://sqltutor.fsv.cvut.cz/cgi-bin/sqltutor), or SQL Trainer (http://www.inf-classic.fh-reutlingen.de/bisic) by Tomislav Bisic (Bachelor Thesis, Reutlingen University, 2008).

There are proven techniques how to grade the results taking into account guessing and "exclusion reasoning" [31]. For

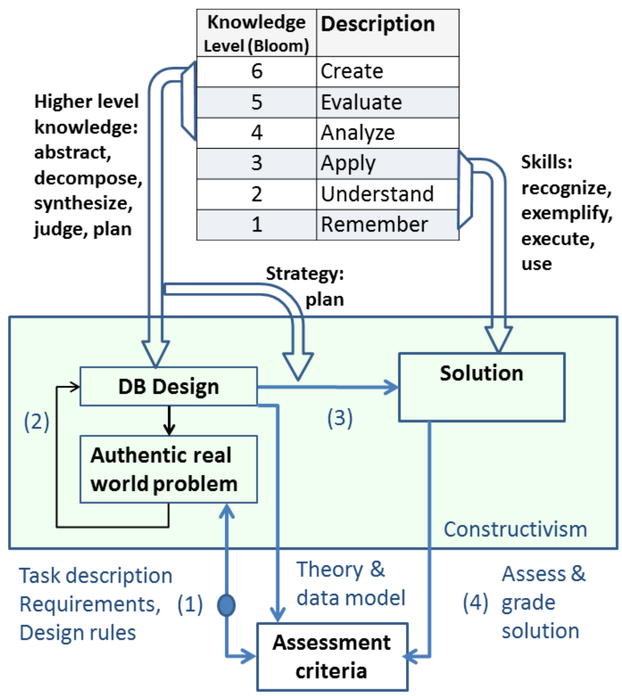

Fig. 4. Interrelation of knowledge levels to problem solving (constructivism) and assessment

instance the use of the correct aggregate functions in SQL should only count if the appropriate grouping is chosen for the query. For an unnecessary complex solution a penalty could be applied as those solutions tend to be pure guesses or the result of copying solutions from a similar task.

Much more challenging to assess are creative tasks and real project work. Here we have to assess artifacts that are the result of an analysis, or an evaluation, which flows into a creative process that eventually leads to a new solution. Especially design tasks like a data model, an application design, or a software architecture model are of this sort and cannot be measured only by the degree of fulfillment of the requirements. Further, important design criteria like simplicity, homogeneity, elegance, and structure of the solution are hard to put into a metric scale. Figure 4 shows the interrelation between knowledge levels and constructivist learning applied to problem solving with the focus on database design.

The approach by Kulkarni and Klemmer [32] includes detailed rubrics for the assessment. They break down the problem to a smaller granularity, but the main problem how to compare and assess different creative solutions in a reproducible way remains unsolved in our opinion. In their paper they provide rubrics and use a mix of students' self-assessment and assessment by the teacher. The example guidelines they give contain sentences like "The storyboards are hard to follow ..." or "The storyboards reasonably address the point of view ..." or "The storyboards are easy to follow ...". Our experience with these general type of rubric leads to a rather individual assessment result. What is "easy to follow" for one grader seems "hard to follow" for another. It mainly depends on the knowledge background of the grader and his ability to put himself into the position of the submitting student.

Kulm [31] recommends involving students in the construction of a scoring rubric. This not only helps to get a higher acceptance of the assessment but also makes the learning goals more transparent. Nevertheless, Ross et al. [33] argue that the students might not pay attention to the rubric if it is too general or to task-specific. If it is too general, it will fail to indicate what is essential in order to assess an artifact or a result. If it is to specific, it is too complicated for the students to use and it will hide the learning objectives or, it will lead to a non-adequate assessment of innovative solutions [33].

We have tried to reproduce the grading concept. But even when the rubrics where created together with the students, there was always a potential for disagreement whether a criteria was "reasonably addressed" or not and whether it "was easy to follow" or not. When we came up with criteria that were objectively measurable like: "Is it possible to query for a customer?" and "Does the query for a non-existent customer produce the message *customer not found*?". The number of criteria exploded and our impression was that a genuine innovative solution could not be foreseen and hence no appropriate question rubrics could be formulated in detail.

Fortunately, in the domain of database modeling the situation is not that hopeless because the data modeling task is underpinned and guided by a sound theory and a formal model. The data modeling task is given in form of a narrative description of the situation. The goal is to first produce a conceptual model, say an Entity-Relationship Model (ERM), and then transform the model to a normalized Relational Model (RM). Finally an SQL Schema will be produced for a specific database product that can be executed to create a database.

Figure 4 shows how different factors influence the assessment of database modeling.

The first step (1) from reading the task description to understanding what data will be necessary and need to be stored in the database is the most crucial. Sometimes, textbooks recommend identifying nouns and verbs which give hints what objects (nouns) to store and how they interrelate (expressed by the verbs). This works only partially as the description contains also activities and procedures that are applied to the data. The procedures are described by sentences using nouns and verbs, but neither of these are stored in the database, except possibly for their results. Therefore, some nouns and verbs have to be ignored, but others manifest itself as data in a database.

(2) The structure of the data depends on how the scenario is perceived. As an example take the address of a person. When we deal with that person as a customer, the address will be some property of the person. If the scenario is at the land registry office, the address of a lot is an entity of its own and the owner is only a property of the lot. In other cases it might be appropriate to have something in between like using the determining dependency between ZIP code and city.

(3) The outcome of the conceptual data model depends largely on the view of the task. This makes automatic assessment and grading of such a model difficult. Nevertheless, if two conceptual models with the same semantics are transformed to relational structures and normalized they will converge to the same normalized relational model. This proposition is supported by transformation rules, structural design patterns [23, chap. 7], and the relational normalization theory [34].

(4) Instead of manual assessing the conceptual data model (the ERM in our case), our idea was to rather assess and grade the transformed and normalized data model. In order to transform the model automatically to a relational model we used algorithms that applied heuristics, like transformation rules and design patterns. The normalization is done by a program that uses the semantics, i.e., functional dependencies, of the conceptual model. In the following subsection, we give some examples for how to transform different relationship types into relations.

A. *One-to-many Relationship*

One-to-many resp. many-to-one relationships are ubiquitous. This is also the reason, why there exists a number of synonyms for it, such as master-detail, body-feet, trunk-rootage, trunk-branches, one-level hierarchy. They may be used to model any kind of hierarchy: folder-files, house-rooms, order-items, bill of materials, collections, document structure, drainage system, etc. The structure could be expressed as ERM in various ways.

As an example, Figure 5 shows one-to-many (1:*) relationships in UML-notation that represent the same situation. All three model a customer order. Model (a) models the order in one single entity using user-defined data types (UDT) for the *customer* and *details* attributes. The *Details* attribute is a list of order details. Model (b) shows two entity types, *OrderHead* and *OrderDetail*. The *OrderHead* uses the a customer type UDT and *OrderDetail* are components of *OrderHead*. The third model (c) uses three entity types, one for the customer and two for the order. The relationships between Customer and OrderHead (*places*) as well as between OrderHead and OrderDetail (*contains*) are one-to-many. *OrderDetail* -a weak entity- depends on *OrderHead*.

If each model is transformed in a canonical way [23, chap. 7] into a relational model the result is similar to the ERM model (c) with foreign keys added to *OrderDetail* and *OrderHead* to represent the relationships *orders* and *contains*. In the case of ERM (a) the complex list data type of attribute *Details* is transformed to a separate relation connected by a 1:* relationship to the original *Order* relation. The UDTs used in model (a) and (b) need to be broken up into atomic data types in order to bring the relations into the first normal form. The result is shown as Bachman-diagram in Figure 6. If the model is further normalized to the third normal form (3NF) the relation *OrderDetail* will be broken up because of the functional dependency (FD) of *description* and *price* from *itemId*. This FD should be deduced from the scenario description.

B. *Many-to-many Relationship*

Another important structural pattern is the many-to-many relationship. The participation of employees in different projects is a good example for this pattern. An employee can work in many projects and a project will usually be worked on many employees. The worked hours shall be recorded per employee and project. This situation is modeled in the classical Chen-Notation in Figure 7. The entities can be directly transformed to relations. As the relational model can basically only represent 1:* relationships it is necessary to express the relationship in the form of a relation whose primary key is formed by the keys of *Employee* and *Project*. Both key attributes are foreign keys referring to *Employee* resp. *Project* as shown in Figure 8. The relationship attribute *workedHours* is added to the relationship table.

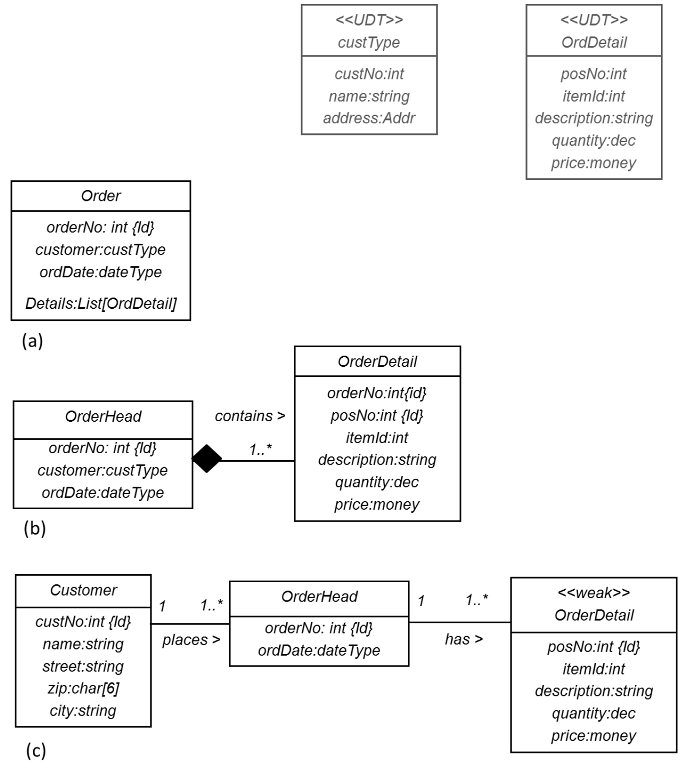

Fig. 5. Entity relationship models for a customer order. (a) one complex entity type using UDT (b) order composition (c) order split into 3 entities (Customer, OrderHead, OrderDetail)

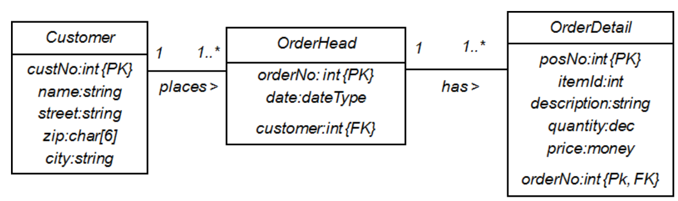

Fig. 6. Normalized relational model generated from the ER-Models shown in Figure 5

C. *General Transformation Rules*

The previous examples may be generalized in the following way:

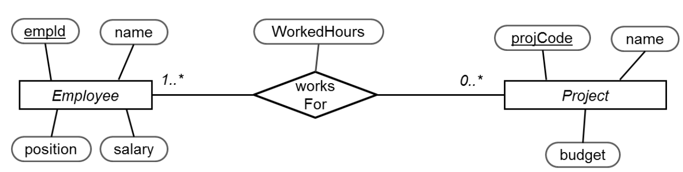

Fig. 7. Entity relationship model (Chen notation) with many-to-many relationship representing employees that work in different projects

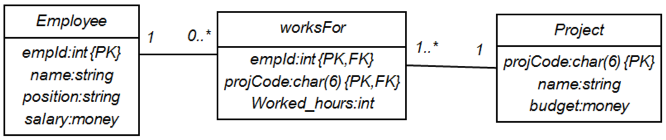

Fig. 8. Normalized relational model generated from the ER-Model shown in Figure 7

1) Map each entity type without attribute change to a relation type.
2) Transform each one-to-many relationship into a foreign key on the many-side. Special case one-to-one can be achieved by a foreign key that is a primary key as well. If the relationship has attributes then place them on the relation with the foreign key.
3) Construct a new relation for every many-to-many relationship using the primary keys from the connected entities as a compound primary key. Add the relationship attributes to the relationship table.
4) Construct a separate relation type for every collection type attribute and add an foreign key to this new relation pointing to its original relation.
5) Break up each complex UDT into its atomic data types in order to bring the relations into first normal form.
6) Normalize the relations into third normal form.

Applying these rules will transform equivalent entity relationship models into the same relational model. Even different viewpoints will merge to the same relational schema because of the normalization process. The final result of the given examples from Figure 5 is the normalized SQL schema listed in Figure 9. The aim of this SQL schema is that the graphical model is now available as textual schema that can be further assessed with the help text analysis methods.

The transformation process can be largely automated with data modeling tools provided by database vendors. To cope with possible naming variance it is possible to provide a list of relevant synonyms and its abbreviations. As example, we demonstrate the process by using the Oracle SQL Developer in the next subsection.

D. Practical Issues for Automatic Mapping an ERM to a normalized RM

Our goal is to automate the assessment of creative and analytic knowledge like the design of an entity relationship model. We want to show that the assessment of an ERM can be considerably automated by the help of standard database tools. Oracle's SQL Developer is a graphical entity relationship modeling tool. Figure 10) shows its graphical user interface. The graphical representation of the ERM is entered on the right pane. The elements of the ERM (entities, attributes, domains) are recognized by the tool and listed in the lower left pane of the display. The purpose for using this tool is threefold:

1) make students acquainted with a leading commercial modeling tool.
2) provide the students with a widely used rendering (crow foot notation) of the ERM.
3) allow the transformation of the ERM into a textual version (SQL schema).

```
CREATE TABLE Customer
  (
  custNo INTEGER NOT NULL primary key,
  name VARCHAR(22),
  street VARCHAR(22),
  ZIP CHAR(6),
  city VARCAHR(22)
  )
;

CREATE TABLE OrderHead
  (
   orderNo INTEGER  NOT NULL PRIMARY KEY,
   customer INTEGER NOT NULL,
   ordDate DATE,
   FOREIGN KEY customer REFERENCES Customer
             on delete restrict
  )
;

CREATE TABLE OrderDetail
  (
   posNo INTEGER  NOT NULL ,
   itemId INTEGER NOT NULL ,
   quantity NUMBER ,
   orderNo INTEGER  NOT NULL,
   FOREIGN KEY orderNo REFERENCES OrderHead
             on delete cascade,
   FOREIGN KEY ItemId REFERENCES Item
             on delete restrict
  )
;

ALTER TABLE OrderDetail
  ADD CONSTRAINT "OrderDetail PK"
  PRIMARY KEY ( orderNo, posNo ) ;

CREATE TABLE Item
  (
   itemId INTEGER NOT NULL PRIMARY KEY,
   description VARCHAR2 (44) ,
   price NUMBER(8,2)
  )
;
```

Fig. 9. SQL Schema in 3NF generated from the relational model of Figure 6

The student uses this tool to edit his ERM model of a given scenario. This model can be transformed to an SQL schema by just pressing a button. Structured data types are mapped to user defined data types (UDT). Multivalued attributes are mapped to an array type (SQL:1999). The schema should be normalized in order to match a normalized reference model. The normalization is done by a program that implements the normalization algorithm given by Vinek et al. [35]. Functional and multivalued dependencies are derived from a description of the modeling task. The dependencies are input to the program in the following form

<attrIds> -> <functional-dependent-attrIDs>.
<attrIds> ->> <multivalue-dependent-attrIds>.

The program handles multivalued dependencies and structured attributes. For assessment purposes, we count the number of matching elements. Primary and foreign keys are specially weighted to reflect their importance. We deal with possible naming variants by supplying an editable list of names for all relevant elements (entities, attributes, and relationships) and use their identifier (Id) instead. This is feasible, since a typical modeling assignment contains usually less than 30 names with at most 4 equivalent variants each. The synonym list for customer order is similar to Table I. With such a list we

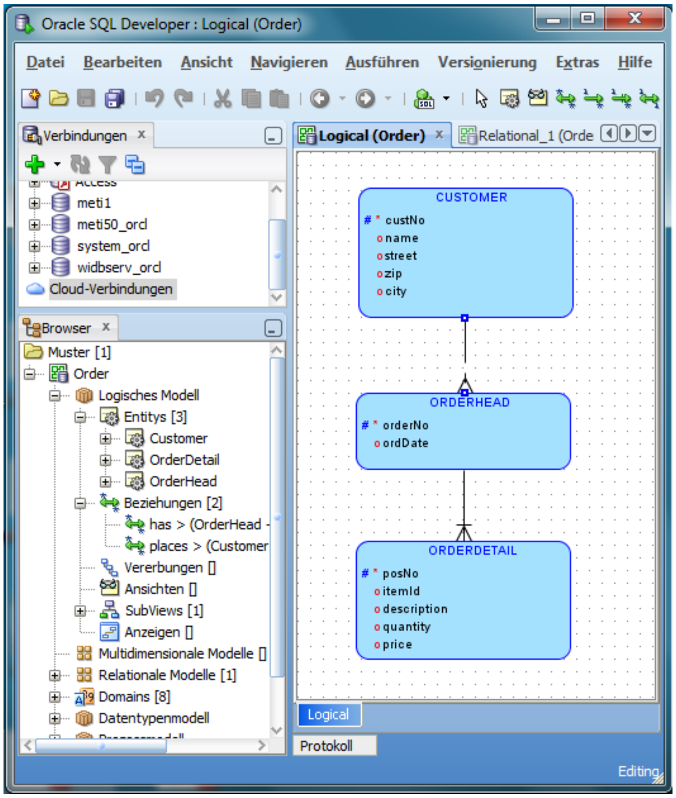

Fig. 10. Screen-shot of Oracle's SQL Developer showing the ERM of Figure 5 (c)

TABLE I. SAMPLE SYNONYM LIST FOR CUSTOMER ORDERS

Id	name1	name2	name3	name4
1	customerNo	customerId	custNo	custId
2	custName	name	cName	
3	ZIP	postCode	PLZ	
...	...	...	...	...
12	itemId	itemNo	productId	prodId
13	description	descr	itemName	
14	price	unitPrice	salesPrice	

identify and match automatically most synonyms that appear in students solutions.

V. TECHNICAL FRAMEWORK AND INFRASTRUCTURE

Our framework of technologies provides a central, web based repository for teaching material, lab environments, multimedia, communication and collaboration tools.

A. E-Learning Portal

We provide all e-learning material through a portal (see http://dbtech.uom.gr and [36]) using Moodle as the software platform. It contains all theory units, mostly as reading material, video lectures, tests, assessments and experimental lab environments that will be described in the following subsection. Local versions, like translations or modifications that fit the curriculum constraints, are hosted and maintained at the project partners' sites (https://relax.reutlingen-university.de for Reutlingen University, or https://elearn.haaga-helia.fi/moodle/login/index.php for Haaga-Helia University for Applied Sciences).

B. Virtual Laboratory Infrastructure

The lab environments are available either through technologies like desktop virtualization or virtual machines running computer software images. The latter is used when the image only uses free software. In this case, there is no need to control the number of downloads or to provide licenses. After downloading the image it can run off-line. Free players for the image are widely available, e.g., VirtualBox.

The virtual infrastructure contains first of all, a database management system (DBMS), like PostgreSQL or free versions of commercial systems like Oracle XE or DB2 UDB Express Edition. For the database modeling we use the free Oracle SQL Developer in conjunction with a normalization software that was developed by a student.

For commercial software products, which require licenses, the use of a desktop virtualization is more appropriate since it allows easy control of the number of remote application accesses. Citrix XenDesktop or VMware View are examples that provide a Virtual Desktop Infrastructure (VDI) for different operating systems.

VDI provides remote access to a pool of virtual machines through a connection broker. If the license policy is only for a certain number of concurrent users it is no problem to limit the concurrent users with this software. Access control may be enforced by LDAP or Active Directory. The virtual machines are automatically managed for every user in terms of multiple, customized instances of computer systems and applications. Independent virtual machines may be assigned to avoid any resource access conflicts. Access to different operating systems is possible and the assignment to a client's PC may be persistent or transient.

The virtual machines are accessed from a client machine via local or public area network. Client computers only need a web browser with ActiveX or Java Applet technology support. Such a support is given by the most common web browsers.

DBTech EXT uses a VDI operated by the University of Málaga. The number of concurrently active virtual machines depends on the resources (processor cores, memory, and disc space) provided. For the DBTech EXT labs Málaga uses two VMware servers with two quad-core processors and 32 Gigabytes of RAM each [37]. This infrastructure has enough power to run 96 concurrent virtual systems, each with 512 Megabytes of memory. The VDI architecture is presented in Figure 11 showing the VMware architecture consisting of a virtual center and two Hypervisor ESX servers that provide for multiple operating systems running on a single physical server. The broker is responsible for dispatching the connection requests from clients and to control the access with the help of an authentication service.

VI. EVALUATION AND EXPERIENCES

The experiences mainly stem from two EU funded projects described earlier that were carried out during the years 2002-2005 and 2009-2010 (see http://www.dbtechnet.org).

A couple of example e-learning modules have been developed as testing materials and these courses were used for virtual workshops conducted during and after the second project.

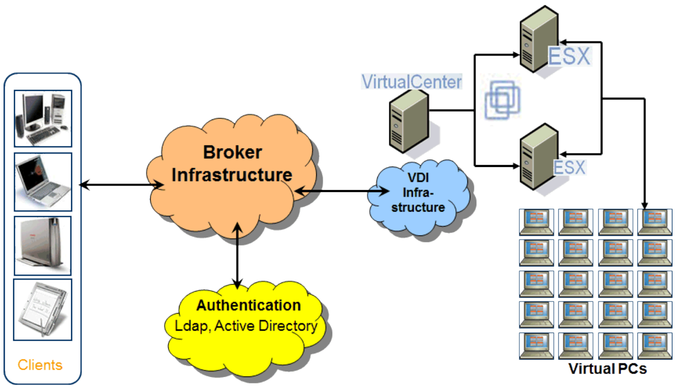

Fig. 11. Virtual Desktop Infrastructure for virtual labs [37]

For the virtual workshop the e-learning platform was enhanced by communication and collaboration tools like Skype, discussion boards and upload areas for deliverables. Teaching materials were structured and furnished with exercises and assignments for the students. The exercises used the previously described virtual infrastructure to guarantee a predefined and fully functional environment. Assessment of the students was done by on-line tests preferably in form of multiple choice questions.

In addition, self-assessment was provided for design tasks using a commercial data modeling tool in conjunction with a normalization software. The software supported semi-automatic grading and self-assessment for data models described in Section IV, which relieved the teacher from time consuming manual assessment.

The effectiveness of our hands-on labs was assessed by weekly online-tests. For evaluation purpose we divided the students into two groups: one group of 23 students took the traditional paper exercises and the other group of 17 students enrolled in the e-learning hands-on labs using the virtual lab environment.

Both groups participated at weekly self-assessment tests. The answers to the multiple choice questions were collected and assessed with the help of the e-learning system. However, the e-learning system provided only support for multiple choice questions to test the analytic skills and not the construction of knowledge or innovative solutions. As a consequence, additional tools were used in the virtual lab environment to assess creative tasks like data modeling.

It turned out, that the e-learning group with hands-on labs performed 28% better than the control group. The difference was even more apparent, when we ranked all students and found that 64% of the e-learning group were found better than the median, whereas only 41% of the control group ranked better than the median.

At the end of every semester the students have to pass written examination under supervision. We compared the results of our e-learning course students ($n = 17$) with previous year's students (our control group, $n = 36$) who were taught in the traditional way. The learning topics of the written examination were weighted according to its importance for the learning goal. For an easy comparison of the results we scaled the grades for each topic from 0 (least) to 100 (best). For example, a grade of 66 indicates that $2/3$ has been achieved.

The results from the weekly tests could not only be confirmed, but surpassed. The significance of the results was tested with Student's *T-Test* under the hypothesis "E-learning with hands-on lab is *not* superior to traditional learning" (H_0) and the standard deviation was tested with Fisher's *F-Test* with the hypothesis "The variance of the mean value of the results are the same" (H_0).

TABLE II. STATISTICAL EVALUATION RESULTS

criteria	e-learn $n = 17$	control $n = 36$	% diff of control	T/F test for $\bar{x}/\sigma$, $\alpha =$
$\bar{x}$ total grade	66	48	+37%	0.009
σ total grade	17.7	18.6	−4.8%	0.966
$\bar{x}$ design grade	76	48	+58%	0.0004
σ design grade	12.9	22.8	−52%	0.077
$\bar{x}$ SQL grade	74	46	+61%	0.0002
σ SQL grade	24	18	+33%	0.196
$\bar{x}$ TaMgmt grade	65	38	+37%	0.009
σ TaMgmt grade	34	26	+31%	0.187

The e-learning group achieved on average 37% better total grades than the control group (see Table II). The hypothesis H_0 was rejected and H_1 "E-Learning with hands-on lab is superior to traditional learning" was accepted. The standard deviation of the e-learning group was slightly smaller than for the control group, which indicates that the spreading of the learning success was similar.

Looking at the examination results in detail revealed that the e-learning group achieved 58% better grades for database design tasks and even better results (+61%) for their SQL competence. For the database design task H_0 was rejected with an error $\alpha = 0.0004$, which means that H_1 "E-Learning with hands-on lab is superior to traditional learning" was accepted with a confidence level of 99.96%. The standard deviation of the e-learning group was 52% smaller compared to the control group leading to the rejection of H_0 with an error $\alpha = 0,077$ and acceptance of H_1 "The variance of mean value of the results are different" with an confidence level of 92.3%. This leads to the statement that the learning success is more uniform when using hands-on labs.

The results for the SQL competence of the students with hands-on labs was even more impressing with 61% improvement (error $\alpha = 0.0002$, 99.98% confidence). The grades for the transaction management (TaMgmt) topic improved by 39%. In both cases, the learning improvement was accompanied by a larger standard deviation (+33% and +31%), which is in contrast to the result of database design topic. The result is not very significant ($\alpha = 0.196$ and $\alpha = 0.187$) but the reason for this result is unknown and needs further investigation.

The methodologies used to evaluate and assess our concept was not only statistical, but included informal and formal (survey conducted via the e-learning platform) feedback, and discussions with students and experts (professors and industrial trainers). The students pointed out the motivational aspect of real products. They claim higher competence and labor market attractiveness. A negative point was that the products are "black boxes" that allow no inside view. Students felt happy with the e-learning support material as this allowed them to check most of their solutions on-line.

The teachers/instructors indicated that some students took wrong reasoning from the products' behavior that was difficult to correct later. This was especially problematic, when a strong theoretical foundation was essential, as with normalization theory and serializability. Teachers appreciate that they had more time to discuss difficult questions with students. The time saved was attributed to the self-assessment capability of the virtual lab environment.

At Reutlingen University, study projects of real world problems have been incorporated into the curriculum for more than 10 years. Over many generations of students the feedback was uniformly positive. Students appreciate the real life character of the projects. In about one third of the projects, the problem was posed by a company that also collaborated with the student teams. From a didactic point of view the motivation was kept high if the company or the university committed itself to use the project results. In most cases, this was a software developed by the students.

Problem based learning confirmed the proposed high motivation, if the knowledge background of the project team was sufficient to master the problem. It was not necessary for every participant to have management competence or to be an expert programmer. It was sufficient to have at least one with the necessary capability. In most cases, this stimulated the team and resulted in an intensive team internal learning process. The supervising professor has the responsibility to make sure that the students with less knowledge will not get frustrated. A possible intervention could be an additional training for the "weaker" students or to assign a different role to the "dominant" student. In individual situations we have successfully been able to let more advanced students act as trainers for the group over a certain time period.

Comparing student teams that work physically together outperform teams that only work together virtually. In feedback discussions the students state a lower motivation and commitment to the project team if they worked remotely without meeting each other. Asking for reasons the students named the missing personal contact and commitment. In contrast, the teams that met regularly developed a culture of responsibility that supported motivation and contributed to the project success.

VII. Conclusion and Future Work

The outstanding lessons learned of this long term e-learning experience can be summarized in five statements:

1) A key success factor is the adequate decomposition of the knowledge domain. Only if this requirement is granted, the necessary small chunks of information are identified and can be prepared according to our e-learning model. If the chunks are not small and sufficiently independent it is hard to provide e-learning modules that can be worked through without the constant help of the teacher.
2) E-learning is not superior to face-to-face teaching but together with the hands-on lab it provides about 30% to 40% better learning results. The preparation of study material is much more elaborate than for traditional teaching.
3) E-assessment scales better only for lower comprehension level (Bloom's taxonomy) and partially for the application level.
4) For higher level (deeper understanding) as synthesis (create), evaluation and analysis, specific tools are necessary that are able to capture and measure the work results. For the case of *data modeling* we demonstrated this successfully.
5) Conducting real world projects with small groups of students provides the highest motivation and yields long lasting competence and enduring confidence for the students.

The e-learning method with hands-on lab should be generally applicable if the subject has a strong formal foundation. This is the case for engineering and formal sciences. For these domains it should be possible to create programs or machines which students can use and experiment with. Possible examples are computer aided electronic design, process simulation, and structural design.

Another challenge is the reuse of e-learning material. Even though there exists an e-learning standard, modules are usually technologically incompatible, e.g., a Moodle (https://moodle.org/ course unit cannot be used in a Blackboard (http://www.blackboard.com/Sites/International/EMEA/index.html) environment. The reuse of e-learning material might be further restricted by different versions of the same product or browser.

The question of self-assessing the learning outcome depends on the rubric. If the topic is formally grounded, sufficient criteria for a quantitative rubric should exist. If a satisfactory metric for the assessment can be found, it is only a question of implementation. Expert systems with explanation component can be used to assess the students artifacts and explain the grading or comment the students work.

The use of hands-on labs is applicable for engineering, ICT, and partly in natural sciences, where a kind of sandbox or virtual lab environment can be provided with design and simulation tools used in industry. Possible examples are architecture, mechanical engineering, electronics, and chemistry. In all these disciplines, the results of creative processes are depicted in a formal way (construction drawings, formal language, model, schemata, etc.), which can be analyzed and assessed automatically according to precisely defined criteria. Case studies for these disciplines are subject of further research.

It is less likely that creative tasks with low formal underpinning can be dealt in a similar way. The result of a complex project that involves different technologies and if the solution includes manual tasks will be still too hard to assess automatically. Examples for this type of projects are process optimization or product development. The general task to assess any kind of creative oeuvres still remains a challenge for future work.

References

[1] Fritz Laux, Martti Laiho, and Thomas Connolly, "E-Learning with Hands-On Labs in Higher Education", In F. Laux and P. Lorenz (eds.), *The Seventh International Conference on Internet and Web Applications and Services* (ICIW), pp. 231-237, Stuttgart, Germany, 2012; ISBN: 978-1-61208-200-4.

[2] Ludwig Issing, "Psychologische Grundlagen des Online-Lernens" [engl. Psychological Foundations of Online Learning] , in: Paul Klimsa and Ludwig Issing (eds.), *Online-Lernen - Handbuch für Wissenschaft und Praxis*, 2^{nd} ed., Oldenbourg V., München, 2011.

[3] André Kukla, *Social Constructivism and the Philosophy of Science*, London: Routledge, 2000; ISBN 0415234190.

[4] Lorin W. Anderson and David R. Krathwohl, (Eds.)(2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Blooms Taxonomy of Educational Objectives*, New York, Longmans, 2001.

[5] Hermann Astleitner, Iwan Pasuchin, and Christian Wiesner, "Multimedia und Motivation - Modelle der Motivationspsychologie als Grundlage für die didaktische Mediengestaltung" [engl. Multimedia and Motivation - Models of Motivation Psychology as Foundation for Didactic Media Design], Medienpädagogik, Zeitschrift für Theorie und Praxis der Medienbildung, 22.3.2006. [Online]. Available: http://www.rz.uni-frankfurt.de/internet/elearning_2008/pdf/Artikel_Astleitner_und_co.pdf, last access: 16.08.2013.

[6] Hermann Astleitner and Christian Wiesner, "An Integrated Model of Multimedia Learning and Motivation". Journal of Educational Multimedia and Hypermedia, 13(1), pp. 3-21, 2004; ISSN 1055-8896.

[7] Manfred Prenzel, "Sechs Möglichkeiten, Lernende zu demotivieren" [engl. Six Possibilities to Demotivate Learners], in: H. Gruber and A. Renkel (eds.), *Wege zum Können, Determinanten des Kompetenzerwerbs*, Bern, Huber, pp. 32-44, 1997.

[8] Manuela Paechter, Karin Schweizer, and Bernd Weidenmann, "Lernen in virtuellen Seminaren: Neuigkeitsbonus oder Adaption an ungewohnte Lernbedingungen" [engl. Learning in Virtual Seminars: Novelty Bonus or Adaption to Unusual Learning Conditions], in: F. Scheuermann (ed.), *Campus 2000 - Lernen in neuen Organisationsformen*, Münster, Waxmann, pp. 279-288, 2000.

[9] Seymour Papert, *Constructionism: A New Opportunity for Elementary Science Education* , Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, 1986.

[10] Ernst von Glasersfeld, "Konstruktion der Wirklichkeit und der Begriff der Objektivität" [engl. Construction of Reality and the Concept of Objectivity]; in: Heinz von Foerster: *Einführung in den Konstruktivismus*, publ. of Carl-Friedrich-von-Siemens-Stiftung, vol. 5; München: Piper, 1992; ISBN 3-492-11165-3.

[11] Jochen Gerstenmaier and Heinz Mandl, "Wissenserwerb unter konstruktivistischer Perspektive" [engl. Knowledge Acquisition under Constructivist Perspective], Zeitschrift für Pädagogik, 41. Jg, Nr. 6, pp. 867-888, 1995.

[12] Thomas Connolly and Carolyn Begg, "A Constructivist-Based Approach to Teaching Database Analysis and Design", Journal of Information Systems Education (JISE), Vol. 17(1), pp. 43-53, 2006.

[13] Benjamin Bloom, Max Engelhart, Walker Hill, and David Krathwohl, *Taxonomy of educational objectives: the classification of educational goals*; Handbook I: Cognitive Domain, New York, Longmans, Green, 1956.

[14] Christian Spannagel, *Benutzungsprozesse beim Lernen und Lehren mit Computern* [eng. Processes for Learning and Teaching with Computers], PhD-Thesis, Hildesheim, Franzbecker, 2007; ISBN-10: 3-88120-443-9.

[15] David R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview", Theory into Practice, Volume 41, Number 4, Autumn 2002, College of Education, The Ohio State University.

[16] Griff Richards, Irwin DeVries, "Revisiting Formative Evaluation: Dynamic Monitoring for the Improvement of Learning Activity Design and Delivery", In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (LAK 2011), Banff, AB, Canada, February 27 - March 01, 2011.

[17] Gary M Velan, Philip Jones, H Patrick McNeil, Rakesh K Kumar, "Integrated online formative assessments in the biomedical sciences for medical students: benefits for learning", in BMC Medical Education, Volume 8:52, 2008; DOI: 10.1186/1472-6920-8-52.

[18] Esyin Chew, Norah Jones, Haydn Blackey, "Implementing Institutional Online Assessment Addressing the Challenges", in Proceedings of: Hybrid Learning, Third International Conference (ICHL 2010), Beijing, China, August 16-18, 2010.

[19] Pete Thomas, Debra Haley, Anne deRoeck, Marian Petre, "E-Assessment using Latent Semantic Analysis in the Computer Science Domain: A Pilot Study", Proceedings of the Workshop on eLearning for Computational Linguistics and Computational Linguistics for eLearning (eLearn '04), pp. 38-44, 2004.

[20] Colin A. Higgins and Brett Bligh, "Formative Computer Based Assessment in Diagram Based Domains", Proceedings of the 11th annual SIGCSE conference on Innovation and technology in computer science education, pp. 98-102, Bologna, Italy, 2006

[21] Luis Fernández Sanz, "Analysis of non technical skills for ICT profiles", 5^{th} Iberian Conference on Information Systems and Technologies (CISTI), Santiago de Compostela, 2010.

[22] Dimitris Dervos, Martti Laiho, and Fritz Laux, "DBTech Pro Framework Reference Courses and Topics", [Online] http://dbtech.uom.gr/mod/resource/view.php?id=592 (Excel-Format), last access: 16.03.2012.

[23] Abraham Silberschatz, Henry Korth, and S. Sudarshan, *Database System Concepts*, New York, McGraw-Hill, 2011.

[24] Gerhard Weikum and Gottfried Vossen, *Transactional Information Systems*, Morgan Kaufmann Publishers, 2002.

[25] Johann Amos Comenius, *Didactica Magna*, 1657, In: Klaus Schaller (ed.), Ausgewählte Werke. vol. 1, Hildesheim: Olms, 1973, p. 82.

[26] Gordon M. Burghardt, "The evolutionary origins of play revisited: Lessons from turtles". In M. Beckoff and J. A. Byers (eds.), *Animal Play: Evolutionary, Comparative, and Ecological Perspectives*, Cambridge, UK: Cambridge Univ. Press, 1998.

[27] Doris Bergen (ed.), *Play as a Medium for Learning and Development*, Portsmouth, NH: Heinemann, 1987.

[28] Andreas Flitner, *Spielen - Lernen : Praxis und Deutung des Kinderspiels* [engl. Play - Learning : Practice and Interpretation of Children's Game] , 3^{rd} ed., 1998.

[29] Dr Gordon Russell and Andrew Cumming, "Improving the Student Learning Experience for SQL Using Automatic Marking", IADIS International Conference Cognition and Exploratory Learning in Digital Age (CELDA), Lisbon, Portugal, 2004, pp. 281-288.

[30] Antonija Mitrovic, "An Intelligent SQL Tutor on the Web", International Journal of Artificial Intelligence in Education 13, 2003, pp. 171-195, IOS Press.

[31] Gerald Kulm, Mathematics Assessment: What Works in the Classroom. San Francisco, Jossey-Bass, 1994, ISBN: 978-0-7879-0040-3.

[32] Chinmay Kulkarni and Scott R. Klemmer, "Learning design wisdom by augmenting physical studio critique with online self-assessment", Technical report, Stanford University, July 2, 2012.

[33] John A. Ross, Carol Rolheiser, Anne Hogaboam-Gray, "Effect of self-evaluation on narrative writing". Assessing Writing, 6(1), 1999, pp. 107-132.

[34] Edgar Codd, "A Relational Model of Data for Large Shared Data Banks", In Communications of the ACM, Volume 13, Number 6, New York, NY, USA, 1970.

[35] Günter Vinek, Paul F. Rennert, A.Min Tjoa, *Datenmodellierung - Theorie und Praxis des Datenbankentwurfs* [engl. Data Modeling - Theory and Practice of Database Design], Physica, Heidelberg, 1982, ISBN 978-3-7908-0225-2.

[36] Georgios Evangelidis, Evangelos Pitsiougas, Dimitris Dervos, Martti Laiho, Fritz Laux, and José F. Aldana-Montes, "DBTechNet portal: A Gateway to Education and Training for the European Database Technology Professional", in Proceedings of the eRA-4 International Conference on the Contribution of I.T. to Science, Technology, Society and Education, 24-26 September 2009, Spetses, Greece.

[37] Ismael Navas Delgado, Maria del Mar Roldán García, José J. Farfán-Leiva, Martti Laiho, Fritz Laux, Dimitris A. Dervos, and José F. Aldana Montes, "Innovative Unity in the Diversity of Information Management Skills Teaching and Training across Europe", in IADAT Journal of Advanced Technology on Education, Vol. 3, No. 3, July 2009, pp. 407-445, ISSN: 1698-1073.