

Automatically Generated and Marked E-Learning Exercises for Logistics Cost Accounting

Markus Siepermann, and Christoph Siepermann

Abstract—This paper presents the concept and realisation of an e-learning tool that provides predefined or automatically generated exercises concerning logistics cost accounting. Students may practise where and whenever they like to via the Internet. Their solutions are marked automatically by the tool while considering consecutive faults and without any intervention of lecturers.

Keywords—Automatic marking, e-learning environment, online practicing, randomly-generated exercises.

I. MOTIVATION

WITHIN university teaching, we can distinguish the following three key parts:

- Imparting knowledge to students (Teaching)
- Letting the students apply their new knowledge and deepening it (Practicing)
- Testing the students' knowledge and their comprehension (Assignment and Grading)

Concerning the teaching task, students mostly prefer traditional lectures [1], even if they are criticized for being antiquated, and they don't want them to be replaced by electronic lectures [2]. Therefore, e-learning should not replace traditional lectures, but rather serve as an additional feature that assists traditional lectures with interactivity and multi-media elements.

With regard to the practising and assignment tasks, the student's individuality, by contrast, is on top. In order to do justice to that individuality, the teacher has to deal individually with each student and his or her abilities and deficits. Because of the uneven ratio between students and lecturers, such an occupation with individual abilities and solutions of students cannot be managed [3]. A way out of this situation can be e-learning. E-learning allows us to provide exercises that are suitable to the individual situation and knowledge of each student, to mark the students' solutions automatically and to give individual hints concerning the lessons a student ought to revise again.

M. Siepermann is with the Department of Business Information Management, Technical University of Dortmund, 44221 Dortmund, Germany (phone: +49 231 755 3159, fax: +49 231 755 3158, e-mail: markus.siepermann@uni-dortmund.de).

C. Siepermann is with the Department of Production and Logistics, University of Kassel, 34109 Kassel, Germany (e-mail: siepermann@wirtschaft.uni-kassel.de).

A suitable e-learning environment should be based upon the interactive and multi-media features of the Internet, which offer numerous possibilities for e-learning exercises and tools. Thus, students are able to practice whenever and wherever they wish. Such self-steered learning is one of the most efficient paths to comprehension [4].

One part of such an e-learning system should be that students can practise with sophisticated exercises in an interactive way [5], such that they have to find the answer on their own by using the learned approaches and their own knowledge. The exercises should not only be composed of simple forms like multiple choice, true-false questions, jumbled sentences or fill-in-the-blank [6]. In these cases, the practising students don't really need their knowledge because often they easily can guess the correct answers by systematically reducing the number of possible answers [7].

Didactically good exercises that really help students understand the contents of lectures should not contain the answer and the problem-solving process in a more or less apparent form. Students should rather be forced to prove their abilities to solve a problem. That can be done by not only evaluating the final result of an exercise, but also by considering the chosen way of problem-solving [3]. In this context, it is important that students can choose their own problem-solving process without any restrictions. Restrictions should only appear if there are technical reasons [8].

Unfortunately, such interactive and sophisticated exercises either do not exist or are very rarely supported by e-learning systems because of their complexity [3]. In most cases today, those exercises are still corrected by human beings [9], [10], [7].

But the disadvantages of traditional exercises are obvious: The manual marking of exercises absorbs resources and results in a time delay between practising, marking and feedback about mistakes and lessons to repeat although immediate feedback would be very valuable [11], [12]. To remedy these problems, automatically marked exercises are needed [13], [14].

Due to the various degrees of freedom, this task is quite difficult to accomplish because often there is not only one correct answer, but rather several answers that are more or less correct. Thus, students' solutions cannot only be classified in the two categories right or wrong. We can instead

identify a scale of correctness concerning the solutions because of consecutive faults and several more or less correct ways to solve the problem [15].

In the following, we will present an e-learning system that satisfies the requirements above. This e-learning system provides exercises for logistics cost accounting. It allows lecturers to manually predefine exercises as well as to generate them automatically. The system automatically marks the students' solutions without intervention of a lecturer. The marked exercises are presented to the students with hints about their deficits and lessons to be repeated.

II. APPLICATION

As traditional cost accounting mainly focuses on production, a source-related allocation of logistics costs to cost units cannot be achieved. The reason is that logistics costs are mostly overhead costs. Traditional cost accounting now argues, that those overhead costs suffer from a missing connection to products and therefore can only be allocated to cost units via value-based allocation measures like direct material costs, direct wages and production costs. Thus, the higher direct material costs, direct wages and production costs of a product are, the more this product is charged with logistics costs, neglecting the real use of logistics services. For example, within traditional cost accounting, a product that is composed of many low cost parts will hardly be charged with logistics costs, although it causes much more procurement costs than a product composed of only a few but expensive parts. The same situation holds for production logistics costs. Applying machine hours as allocation measures supposes a coherence between the production time of a product and its claim for production logistics services. But, rather, the complexity of the production processes is the appointing determinant. Finally, distribution logistics costs are not determined by production costs (as implicitly assumed by traditional cost accounting), but by a product's storage and transport attributes (e.g. dimension, weight etc.) [16]. Taking into account that logistics costs make up 10-25% of industrial enterprises' total costs [17] or 5-10% of turnover [18], [19] and that logistics services have the utmost importance for differentiation in competition [20], these faults in cost allocation may lead to fatal errors in product-related decision-making due to wrong information concerning a product's logistics costs.

Another lack of traditional cost accounting can be seen in the fact that logistics costs are not reported separately in product costing, but as a part of procurement, production, sales and administration overhead costs (lack of transparency). For these reasons, basically two approaches have been developed or can be applied in order to achieve a source-related allocation of logistics costs to cost units:

- Weber proposes a refinement of traditional cost accounting [15]. A similar approach is provided by Reichmann [21]. Both approaches can be applied as absorption accounting or marginal costing.

- The second alternative consists of applying activity-based costing (ABC), which was explicitly developed for indirect service types and therefore can be assumed to be suitable for logistics cost accounting as well. But ABC originally was designed as full-absorption accounting [22], [23], which implies that it does not separate costs into its fixed and variable parts. Therefore, it is not able to provide any information with regard to short-term decision-making, such as accepting or refusing an additional order. Certain further developments in ABC try to remedy this deficiency by separating costs according to their dependency on the operating level and/or convertibility in time, in addition to the separation between costs for process-volume-induced and process-volume-neutral activities introduced by Horváth/Mayer [22]. In this context, we especially have to mention the approaches by Reichmann/Fröhling [24], Glaser [25], Mayer [26] and Dierkes [27].

Although each approach is different from the others they have the following in common:

- In all of these approaches, logistics cost centers are defined as final cost centers.
- Except for the Reichmann approach, which only defines logistics specific surcharge rates, all approaches try to allocate logistics costs of cost centers to cost units via transfer rates that are based on volume-based allocation measures. This implies that we can clearly identify a relation between the output of logistics cost centers and the usage of this output by cost units. If not, we have to do without the allocation of the respective logistics costs or value-based surcharge rates have to be used as a remedy.

The fundamental difference between the approaches according to Reichmann and Weber on the one hand and the activity-based costing approaches on the other hand consists of the different number of calculation steps: Within the Reichmann and Weber approach, the costs of logistics cost centers are immediately allocated to the cost units via cost center-based allocation measures. By contrast, activity-based costing firstly allocates logistics costs to activities. In a second step, these activity costs are allocated to the cost units via activity-based allocation measures. Activities can be aggregated hierarchically over several levels. Commonly there are two hierarchy levels.

The further developments in activity-based costing mainly differ in the way of cost splitting. Cost splitting can refer to the costs' dependency on the operating level, which leads to the differentiation between variable and fixed costs, whereby fixed costs additionally can be differentiated according to the readiness to operate, and/or to the costs' convertibility in time (i.e. their commitment period), which leads to the differentiation between costs that are degradable in the short-, medium- and long-term.

III. E-LEARNING CONCEPT

A. Overview

Altogether, we can distinguish six different approaches of logistics cost accounting: The approaches according to Weber and Reichmann, the original form of activity-based costing developed by Horváth/Mayer and the further developments in ABC developed by Reichmann/Fröhling, Glaser, Mayer and Dierkes. An overview of the principles of each approach is given by [28] and [29].

In order to be able to compare the results of these six approaches to those of traditional cost accounting, the latter is also implemented in the e-learning system. This helps to demonstrate the mistakes of traditional cost accounting in allocating logistics costs to the products and therefore the need of a special logistics cost accounting system.

The major task of each logistics cost accounting systems consists of allocating a manufacturer's logistics costs properly (i.e. according to the products' claim of logistics services) to the cost units and in providing transparent information about the composition of these (product-related) logistics costs. Therefore, the subject of each exercise in logistics cost accounting is the calculation of product costs with special regard to logistics costs.

The relevant approaches and therefore the exercises that shall be implemented differ in

- the allocation measures used,
- the manner of cost splitting,
- the logistics cost categories (resulting from cost splitting) which are allocated to the product units and which are not, and
- the applied calculation scheme.

All types of exercises have in common the master data concerning cost types, cost centers, allocation measures, activities (if needed) and the logistical attributes of the products to be calculated. These master data are independent of the chosen type of exercise and can easily be extended if necessary.

When practising, each type of exercise comprises the following three steps:

- Calculating cost type-based, cost center-based and/or activity-based allocation measures,
- Calculating the product-related values of the logistical allocation measures (i.e. the activity coefficients)
- Performing product costing by using the results of steps one and two.

B. Generation of Exercises

The e-learning system provides two different ways to generate exercises: They can either be created manually by lecturers, or they can be generated automatically by the e-learning system. When *manually* creating an exercise, we have to choose the exercise type (i.e. the logistics cost accounting approach to be applied), the mattering cost types, cost centers, allocation measures, activities (if needed) and the (logistical) attributes of the products to be calculated. If the

master data pool isn't sufficient, new master data can be added to the pool by lecturers. Based on the chosen master data, the e-learning system creates empty data sheets for the initial data that can be filled by lecturers with appropriate values. The calculation can either be done manually by the lecturer in order to test the exercise – the system then checks the solution and shows potentially made mistakes – or it can be performed automatically by the system. Finally, the lecturer can revise the exercise and make some changes before saving it.

When *automatically* generating an exercise, there are several interdependencies between the different data of exercises that have to be considered. For example, a realistic ratio between staff costs and material expenses, variable and fixed costs, direct and overhead costs should be guaranteed as well as a realistic level of the transfer and surcharge rates. These interdependencies are stored as rules in a rule database and are used each time an exercise is generated. The e-learning system randomly generates the basic data and then adjusts these data by using the rules. The dependent and derived values are computed.

C. Difficulty Levels

In order to achieve a broad acceptance of e-learning, exercises should be suited to the actual knowledge of students [8]. For this reason, different difficulty levels should be offered. The possibility of choosing different underlying approaches leads to different difficulty levels and handling times, yet. Additional difficulty factors are the number of cost centers, activities, activity types (e.g. output-based and non-output-based ones), the kind of cost splitting etc. These elements are parameterised and build the fundamentals in computing difficulty level and target time of an exercise.

The number of cost centers or activities, for example, affects the number of calculation steps and therefore must have an impact on the target time. By contrast, the content-oriented difficulty level is influenced by the kind of cost splitting or the number of different activity types occurring in the exercise. Thus, the target time is computed with the number of calculation steps, and the difficulty level is computed with the help of the different difficulty parameters.

When automatically generating an exercise, the components of an exercise are chosen with respect to the given difficulty level. The target time is then also influenced by the chosen components: Complex components lead to longer target time, easy components can be handled faster.

Another variation of difficulty can be derived from the different conditions that may hold at the beginning of an exercise. At easy levels the whole calculation scheme can be provided to users. In this case, students only need to compute the correct values. At top levels, no presets are made and students will have to design the problem solution process, as well as to compute the correct values.

D. Practising with Exercises

Normally, no presets are made to students. Necessary variables have to be defined by students themselves.

Therefore, variables can be created by defining two inputs: The name of the variable and the values. The name of the variable can either be chosen out of a list of possible names or it can be defined in a free text field. In order to recognise free user inputs, a fault-tolerant word recognition function, based upon well known metrics like the Levenshtein or the Damerau distance, is implemented. With the help of these metrics, the correct variable name can be identified out of the user's input.

In exam mode, an exercise has to be solved within the target time. When the target time expires, the solution is sent to the e-learning system and is automatically marked. Beyond the exam mode, a message that the normal target time has expired is displayed. After finishing, the total time of practising is compared to the target time and a grading is presented to the user.

E. Automatic Marking

Each type of exercise is based upon a more or less simple form of calculation that can be modelled by calculation rules [30].

Because of the free user input and the various possibilities to solve the problem provided by an exercise, a simple marking by comparing the results of the students' calculations with the reference solution won't succeed. Rather, we have to follow the students' solution processes in order to understand how they have reached their results. Therefore, we firstly recognise the used variables of the students' solutions. After that the calculation is reconstructed by using the rule database. Doing so, three kinds of faults can occur:

- 1) Necessary variables are missing.
- 2) Values of variables are faulty.
- 3) Needless variables are used.

If values are faulty the system marks the mistakes and inserts a hint. At first sight, the usage of needless variables is not critical. But it may happen that those needless variables aren't needless: They might be missing variables that haven't been recognised correctly. In order to avoid this marking mistake, not only the names but also the values of the variables are used while recognising the variables. If a variable is missing, a percentage is subtracted from the total achievable score and the fault is marked. If there are further calculation steps, the marking algorithm proceeds despite missing or faulty variables. Doing so, we have to care about consecutive faults that result from missing or faulty values.

In order to recognise these consecutive faults, the marking procedure uses the already marked values and recalculates the following calculation steps with these faulty values. Thus, correct calculations are recognised as correct with respect to faulty values. Consecutive faults are also marked, but they don't lead to a reduction in scoring.

IV. SYSTEM ARCHITECTURE

The following objectives were the major focus during the design of the e-learning system:

- Automatic marking of students' solutions
- Provision of an exam mode

In order to put these objectives into practise, the system requires an exercise administration module, a user administration module, an exercise generation module, a master database and a rule database, a configuration module, a content module and a representation module (see fig. 1).

The *exercise administration module* stores and manages the exercises predefined by lecturers as well as the automatically generated ones. All exercises are classified according to their difficulty level. The difficulty level results from the different accounting methods and the calculation components which occur in an exercise. The *user administration module* manages every single user. Every user can act in different roles: Students can practice with exercises, automatically generate exercises and have a look at the marking of their solutions. Lecturers can predefine exercises or automatically generate them, work on exercises like students and have a look at students' solutions to gain an insight into student's knowledge. Administrators assign roles to each user, work on fundamental system parameters and adjust the parameterisation of the difficulty levels. The solutions of users are also stored in the user administration module.

The *exercise generation module* can be used by students and lecturers. Students can choose a difficulty level and choose whether they want to work on that exercise in the exam modus or not. Lecturers as well can automatically generate exercises and select a difficulty level. Additionally, they can in- or exclude single parameters to create a more specialised exercise. The generation module provides an exercise according to the chosen preferences and calculates the target time for solving with regard to the difficulty parameters. The exercise is generated with respect to the master data stored in the *master database* and to the rules stored in the *rule database*. The lecturers are able to add master data and rules to these databases via the *configuration module*.

After the expiration of the target time (if the exam modus was chosen) or after the exercise has been finished by the student, the solution is sent to the *marking module*. This module evaluates the solution using the rule database, marks right and wrong elements and gives hints as to which lessons should be repeated via the *content module*. The exercise is represented via the *representation module*.

The presented e-learning system is a client-server-based system, developed with classical web technologies. Work on exercises and therefore exercise representation takes place at the client. The frontends of the administrative modules also run at the client side. All other modules are only operated on the server side.

