Available online at `http://www.ges2007.de`

Saving Expenses With Technology Enhanced Learning

Richard Hackelbusch

Carl von Ossietzky Universität Oldenburg, Escherweg 2, Oldenburg, Germany
email: `{hackelbusch@uni-oldenburg.de}`
phone: (+49(0)441) 9722 208, *fax:* (+49(0)441) 9722 202

Abstract

Long study terms and a large number of students who do not successfully finish their academic programs are damaging the national economies to a large tune. In addition, personal study guidance to attenuate these problems is very expensive, too. Misunderstanding of examination regulations and a possible adversely balanced supply and demand of courses can be reasons for increasing average study terms. In this paper, we describe how technology enhanced learning can help to save expenses in this field. Our approach is based upon an ontological representation of academic programs and examination regulations. It can help students in planning their curricula and identifying the contents of academic programs as well as academic boards to forecast the number of students which will presumably take a certain course in a certain term.

1 Introduction

A lot of students of higher academic institutions do not finish their academic program within the standard period of study or do not finish their academic program at all (see [8]). These (long) study terms are damaging the national economies to a large tune (see [10]). One reason for this problem is the fact that students often do not know how to interpret the contents and conditions of academic programs. This information is written in documents using a legal language which can be very complicated. In many cases, the publication of explaining subsidiary documents like study guides cannot address that problem because those documents are often to generic, and therefore do not fit for students' individual situations. This can lead to a large demand for study guidance (see [5]). That study guidance is mostly offered by special employees or employees of the scientific staff. Thus, this non automatic guidance is very expensive. Given by employees of the scientific staff, study guidance — in addition — is error-prone because these employees are often not enough skilled. Furthermore, offered study guidance is not availed by many students who actually demand a guidance for different reasons (like bad skilled advisers or shame).

Teaching and examination regulations are forming the statutory framework of academic programs. These regulations are legally binding and, therefore, worded in a legal terminology, and are — as already mentioned — often very complicated, too. In addition, examination regulations of different academic programs can be very heterogeneous. This can be a problem, *e.g.*, if courses of different academic programs have to be integrated into one curriculum (for example "minor subjects"). In addition, there also might exist in parallel different valid examination regulations of academic programs leading to the same degrees even at a single academic institution. This can happen, *e.g.*, after an introduction of a new version of examination regulations. All these facts can increase the demand on course and study guidance.

The field of technology enhanced learning contains information and communication technologies to support activities in learning and teaching. It also comprises activities without influence on the result of learning and teaching activities which is an addition to the field of electronic learning (e-learning) (see [11]). In order to use applications of the field of technology enhanced learning to offer a computer assisted course and study guidance, it is needed to represent academic programs and the corresponding examination regulations in a computer-readable language.

In addition, these facts can lead to problems from the point of academic boards and lecturers. Without computer-readable representations of academic programs and examination regulations it is very hard to forecast the number of students which will presumably take a certain course in a certain term. Required information about, *e.g.*, who is able to take a certain course is not available. This can lead to another bad expense factors. For example, it is difficult to decide how many resources (like rooms of adequate size and equipment, appropriation of enough tutors) should be provided in conjunction with the offer of certain courses. In addition, an adversely balanced supply and demand of courses can increase the average study term, too.

In the next section, we explain or ontological based approach to allow a computer-based decision support system with ontological represented academic programs and examination regulations. The paper is closed with a short overview of related work and conclusions.

2 Approach

In order to save expenses which result from the above motivated problems, we propose the development of an application in the field of technology enhanced learning. That application — called EUSTEL[1] — can assist students in interpreting their possibilities in planning their curricula depending on their individual skills, interests and statutory options. It also can help academic boards in planning the supply of courses for the next terms.

As already mentioned, the basis for such an application is a computer-readable representation of academic programs and the corresponding examination regulations. The relevant semantics of examination regulations are re-

---

[1] Decision Support System (**E**ntscheidungs**u**nterstützungs**s**ystem) in **T**echnology **E**nhanced **L**earning, see [7]

quirements which have to be fulfilled by students in order to get their degrees. These requirements describe possible curricula and can be modeled as processes (see [5]). Basically, modeling examination regulations of academic programs as a process in a computer-understandable language is a more intuitive way than modeling them as a rule set. Among other things, that's because academic programs themselves (and their possible curricula) are kinds of processes.

There are a lot of different approaches that introduce meta-models to allow a modeling of processes or workflows — like *MQSeries* [9] — just to mention one. But there are more semantics than just the process itself to be modeled. For example, it should be possible to model what kind of course can be taken as a specific process step. In addition, a comparison of courses should be possible, too. That's why our approach bases upon ontologies. Ontological concepts allow the explicit specification of conceptualizations [3]. Thus, we decided to use them to model the processes of academic programs and their examination regulations on the one side. On the other side, we use them to allow a semantical representation of the contents of academic programs and their possible curricula, too. We put it all together into one ontological meta-model, called "Curricula Mapping Ontology" (CMO). Academic programs can be modeled as an ontology model instantiating the CMO.

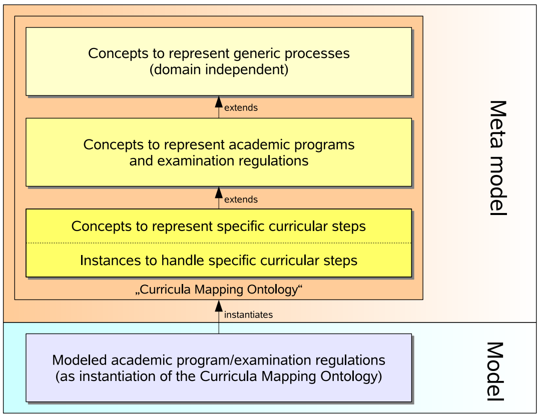


Figure 1: Overview of meta-model

The CMO is separated into three tiers: In the top tier, there are concepts defined which allow the modeling of generic processes. These basic concepts are, *e.g.*, `Process`, `Process_Step`, and `Condition`. A `Process` is a set of `Process_Step` and `Condition` instances which can be connected among each other and form a directed graph representing the process. Examination regulations forming the statutory framework for academic programs can be modeled by using concepts of the middle tier of the CMO: This tier contains concepts that are specializations of the generic concepts of the top tier or that are additional defined. Concepts of this tier are, *e.g.*, `Result`, `Grade_Scale`, and `Availability`. Finally, the bottom tier contains specializations of concepts of the middle tier representing special curricular steps. These steps can be very heterogeneous comparing different academic programs. That's why our approach provides the definition of heterogeneous curricular steps (and bottom tiers). The bottom tier can also contain some instances that connect these special descriptions to allow more generic model interpreters to analyze individual situations of students relating to the modeled examination regulations (see below).

In order to be able to interpret individual situations relating to the modeled examination regulations, individual achievements of students have to be mapped with the model. This can be done by a model interpreter. In our approach, we plan to use a generic interpreter that can interpret models instantiated of the middle tier of the meta-model relating to individual situations. It also can interpret models instantiated of the bottom tier of the meta-model if for each defined bottom tier (which are each specializations of the middle tier) it is modeled how that special concepts are connected to certain concepts of the middle tier. For example, if an academic program has a description of curricular steps concerning the workload (or lecturer, or anything else), this concept has to be connected with a concept representing conditions, in order to be able to define a condition like "the sum of the workload of a specific number of courses has to be bigger than 14".

In the next paragraphs, we give a short overview of the most important concepts of the CMO. We start with the major concepts of its generic tier and explain how those concepts are extended to be able to model examination regulations. We finish this overview with an explanation how special curricular steps can be defined and connected with concepts of the middle tier, in order to allow a generic model interpreter to analyze individual situations relating to such a model.

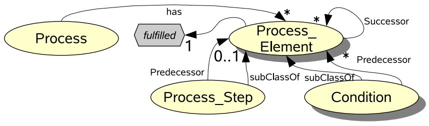


Figure 2: Basic concepts

Figure 2 shows the basic concepts of the CMO to model processes. An instance of `Process` references a set of `Process_Element` instances. These instances either have to be `Process_Steps` or `Conditions`. (`Process_Element` is defined as abstract and therefore not intended to be instantiated directly. Abstract concepts are visualized by shadows.) These instances can be connected as predecessors and successors and form a directed graph that stands for the process. For each instance, a model interpreter has to acquire a boolean value `fulfilled` that stands for a successfully taken process step or a fulfilled condition. The possibility to take a process step has to be interpreted that way that such a step can be taken if the boolean value of the predecessor process element (which can either be a process step or a condition) has to be interpreted as `TRUE` or if there does not exist any predecessor process element. The question how the boolean value of a `Process_Step` instance has to be interpreted will be discussed below. The interpretation of the boolean value of a `Condition` depends on the type of the `Condition` instance which is defined by choosing a specialization of `Condition`, and its predecessors.

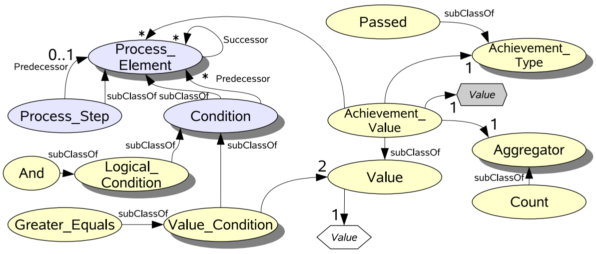


Figure 3: Concepts for special conditions

The type of condition can be chosen by instantiating a specialization of `Condition`, see figure 3. Specializations can stand for simple logical conditions like `And`, or for conditions that compare two numerical values (`Value_Condition`). Like logic gates, the boolean value of logical conditions has to be interpreted depending on its type and the boolean value of its preceding process elements. The boolean value of an instance of `Value_Condition` has to be interpreted as `TRUE` if the type of comparison (like `Greater_Equals`) — which can be chosen by instantiating a specialization of `Value_Condition` — is fulfilled. The two values that should be compared can either be modeled as constant values by instantiating `Value`, or as variable values that can be acquired by aggregating values of the predecessor set of the condition (like the number of successfully passed process steps) by instantiating `Achievement_Value`. The type of value that has to be aggregated depends on the specialization of `Achievement_Type` that is referenced (like `Passed`). Instances of `Achievement_Value` also references one instance of `Aggregator` whose type can be chosen by instantiating a specialization of `Aggregator` (like `Count`). Other concepts of the top tier — which cannot be mentioned here in detail — are, *e.g.*, concepts to dynamically choose the relevant set of predecessors of a condition (like the latest three taken steps), or the possibility to build terms of values.

Instantiating these concepts, very complex processes can be modeled. Just to give an impression how such a process can look like, a simple process with a

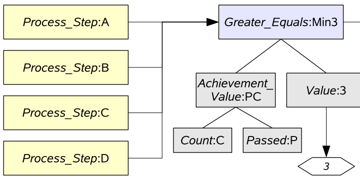


Figure 4: An instantiated part of a process

set of process steps and one condition is shown in figure 4. In order interpret the boolean value of the condition `Min3` as `TRUE`, the number of successfully passed steps of the predecessor set of `Min3` (which are the `Process_Step` instances `A`, `B`, `C`, and `D`) has to be greater or equals three. The first value that should be compared with the second value — and which has to be greater or equals than the second value — is modeled as to be interpreted as a dynamical value that stands for the number of successfully passed predecessor elements of `Min3` (`PC` is an instance of `Achievement_Value` referencing the `Passed` instance `P` which is a specialization of `Achievement_Type`, and `C` which is an instance of `Count` which is a specialization of `Aggregator`). The second value is a constant value (`3` is an instance of `Value` which has the attribute value 3). `Min3` has to be interpreted as fulfilled if the to be dynamically acquired value of `PC` is greater or equals the constant value 3.

The middle tier of the CMO includes concepts that extends the concepts of the top tier, *e.g.*, by defining specializations of `Achievement_Type`. These specializations stand for, *e.g.*, amount, grade, date, type, or attempt. In addition, every process step is connected with a result which itself is connected with a grade scale. With the knowledge of these concepts, a model interpreter can determine whether a process step is successfully passed.

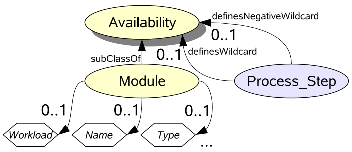
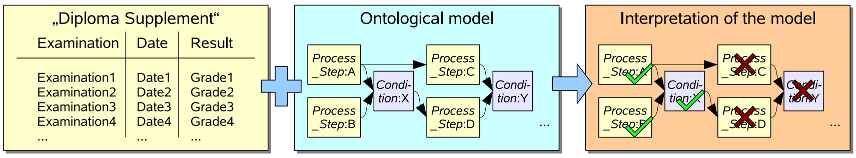


Figure 5: Wildcard concept

In order to be able to define which curricular steps are contemplable to be assigned with a specific `Process_Step` instance, the "wildcard" concept is defined in the middle tier (see figure 5). For every instance of `Process_Step`, a wildcard can be defined. A wildcard is a specialization of `Availability` and stands for a possible assignment of a concrete curricular step to a `Process_Step` instance. Because of the fact that there can be very different types of curricular steps comparing different academic programs, heterogeneous specializations of `Availability` can be defined in the bottom tier of the CMO. In figure 5, there is an exemplary specialization of `Availability` shown: A `Module` that has some optional attributes like `Workload`, `Name`, and `Type`. A concrete offered course of an academic institution can be assigned to the corresponding process step if all descriptions that are defined by the `Availability` instance are similar to the description of the concrete offered course. If there is less information than available defined by the `Availability` instance, only those information is relevant to ascertain if a concrete course can be assigned. For example, if the `Availability` instance only defines that a module has to have six credits, all courses with six credits can be assigned to the corresponding `Process_Step` instance — independent of, *e.g.*, the title or the subject. If elements of the defined description of a specialization of `Availability` should be a possible part of a condition (for example, "workload $>$ 12"), the corresponding entity type has to be connected to an existing specialization of `Achievement_Type` entity type, or a new defined specialization of `Achievement_Type` entity type. We use a connection-concept whose instances contain links to the attribute definition of the wildcard specialization and the defined subclass concept of `Achievement_Type`[2]. Thus, a generic model interpreter can even interpret models of the bottom tier of the CMO.

Due to a lack of space, unfortunately, other important concepts of the CMO cannot be discussed in this paper. These concepts are, *e.g.*, internal processes and process substitutions. The concept of internal processes can be used, *e.g.*, for modeling rules that regulate the retry of (failed) attempts to take a curricular step by modeling an (internal) process that has to be passed in order to pass the corresponding process step. Process substitutions can be used, *e.g.*, in order to model the programs of minor subjects, or to model rules that specify that, *e.g.*, a course can be substituted by two seminars, in order to keep the main process clean.



Examination	Date	Result
Examination1	Date1	Grade1
Examination2	Date2	Grade2
Examination3	Date3	Grade3
Examination4	Date4	Grade4
...	...	...

Figure 6: Functionality of the model interpreter

As already mentioned, in order to analyze students' achievements related to

---

[2] Using OWL, these are the URI of the attribute definition and the URI of the `Achievement_Value` specialization concept.

a specific academic program, it is necessary that the modeled academic program is interpreted concerning these achievements. This task can be done by a model interpreter. A descriptive view of the functionality of such a model interpreter is shown in figure 6: Based upon an ontological modeled set of individual achievements of a single student and the ontological model of the academic program, for each process element of the ontological model, it has to be examined whether its boolean value has to be interpreted as `TRUE` or as `FALSE`, and — in the case of process steps — if a certain process step can be basically taken at all.

From different points of view there are different questions concerning ontological represented academic programs and examination regulations. Students might ask which courses or examinations do they still have to take/pass. Lecturers might want to know how many students following which academic program will presumably request their courses. Academic boards might want to evaluate the quality of academic programs and examination regulations. All decision support systems that can help in answering these questions can benefit from an ontological representation of academic programs and examination regulations. But they might use different methods to calculate answers. Figure 7 shows our suggested architecture for systems like that.

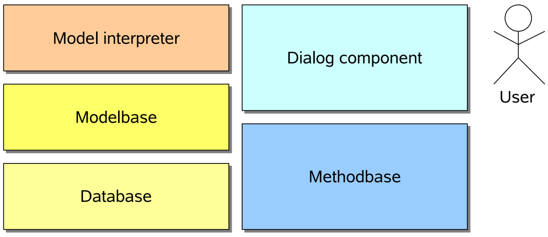


Figure 7: Suggested architecture

In order to evaluate these concepts, we are currently developing the already mentioned EUSTEL-System (see [7]). Therefore, the CMO is implemented using OWL-DL[3] and instantiated for the representation of different academic programs. These artifacts are stored inside the "modelbase". Beside other things, the "database" contains information about the students' individual achievements which are the basis to interpret the model. In order to ensure data privacy guidelines (see [14]), these information should be exclusively physically stored at the examination office. In order to be able to interpret these information related to the model, we implement a "model interpreter" as described above. EUSTEL is intended to offer decision support for students as well as for lecturers/academic boards. That's why we have to implement different methods which are offered by the "methodbase" that allow the conversion of the results of the model in-

---

[3] http://www.w3.org/TR/owl-features/

terpretation into specific knowledge (like: How many students are able to take a specific course, or which courses is a specific student able to take?). In order to offer these services at the same place where students already can take their courses and view their results, we plan to integrate the "dialog component" in the learning management system Stud.IP[4].

3 Related Work

Our approach is less a business model approach to earn money with e-learning (as described, *e.g.*, in [1]). We suggest an approach to save expenses using techniques of the field of technology enhanced learning. It is comparable with other approaches that want to offer decision support for students on basis of their achievements and academic programs like [5], [6], or [12]. As our approach, [5] also uses a process view of academic programs but the semantic representation of whole academic programs and the corresponding examination regulations is not drawn up as much as in our ontological approach. [6] stands exemplarily for rule based approaches to represent academic programs and their examination regulations. [12] is intended to offer an electronic examination office. Other approaches which are also intended to support the target group of academic boards mostly aim financial aspects, like [4]; it provides a business intelligence solution for the controlling of schools. Technically related work are other approaches of ontological representation of law like [2] or the approaches outlined in [13]. Those approaches are mostly very generic, and thus, difficult to use for the representation of academic programs and their examination regulations.

4 Conclusions and Future Work

In this paper, we presented an ontology based approach to design decision support systems relating to academic programs and examination regulations in the field of technology enhanced learning. Implementing systems like these, expenses can be saved because fewer students will probably demand expensive individual course guidance. In addition, lecturers and academic boards can make better prognoses how many students will probably ask for specific courses. This better information can help in saving expenses, too.

To evaluate our approach, we will finish our work on EUSTEL and offer a decision support system for students as well as for lecturers/academic boards. Using EUSTEL, students will be able to plan their individual curriculum on basis of the ontological representation of their academic programs with a graphical user interface. They will be able to run through different settings of their curricula (*e.g.*, choice/changing of certain courses, choice/changing of primary/minor subject) and get a visualization of their individual plans. In addition, lecturers will be supported by EUSTEL in retrieving predictions of the demand for their lessons in certain terms — broken down to different examination regulations applied for the corresponding demanding students.

---

[4] http://www.studip.de/

In addition, future work has to be done concerning the support of the transfer of legal documents containing examination regulations of academic programs into `CMO` models. We suggest a graphical tool that allows the "drawing" of processes according to academic programs which also reduces the visible complexity of `CMO` models. In the long term, we plan to develop a generator that allows the creation of such legal documents out of an existing `CMO` model. Thus, legal document and model would stand for equal semantics.

References

1. Ulrike Bentlage, Peter Glotz, Ingrid Hamm. "E-Learning - Märkte, Geschäftsmodelle, Perspektiven". Bertelsmann Stiftung, 2002.
2. Alexander Boer, Tom van Engers, Radboud Winkels. "Using Ontologies for Comparing and Harmonizing Legislation". In: Proceedings of the 9th international conference on Artificial intelligence and law, pp.60-69, 2003.
3. Tom Gruber. "A Translation Approach to Portable Ontology Specifications". In: Knowledge Acquisition, 5(2):199-220, 1993.
4. Matthias Goeken, Lars Burmeister. "Entwurf und Umsetzung einer Business-Intelligence-Lösung für ein Fakultätscontrolling". In: Proceedings der Multikonferenz Wirtschaftsinformatik (MKWI), pp.137-152, 2004.
5. Martin Gumhold, Michael Weber. "Internetbasierte Studienassistenz am Beispiel von SASy". In: doIT Software-Forschungstag, Stuttgart, Fraunhofer IRB Verlag, 11/2003.
6. Michael Hanus. "An Open System to Support Web-based Learning". In: Proceedings of the 12th International Workshop on Functional and (Constraint) Logic Programming (WFLP), 2003.
7. Richard Hackelbusch. "EUSTEL - Entscheidungsunterstützung im Technology Enhanced Learning". In: Christian Hochberger, Rüdiger Liskowsky (Hrsg.): INFORMATIK 2006 Informatik für Menschen - Band 1, Gesellschaft für Informatik, Bonn, pp.65-69, 10/2006.
8. Heidi Hudabiunigg, Monika Neuhauser. "Studienabbrecher - Gründe und Hintergründe - Erfahrungen mit Studienabbrechern". Survey, Education Highway IST. 1998.
9. Frank Leymann, Dieter Roller. "Production Workflow – Concepts and Techniques". Prentice Hall PTR, 2000.
10. Regierungsrat des Kantons Thurgau. "Studienberatung - quo vadis". Beantwortung einer Interpellation. 2006.
11. Markus Schmees. "Organizing Technology Enhanced Learning". In: Bruce Spencer, Mark S. Fox, Weichang Du, Donglei Du, Scott Buffett (Hrsg.): Proceedings of the Eigth International Conference on Electronic Commerce (ICEC'06), ACM, ACM Press, New York, pp.139-150, 9/2006
12. Elmar J. Sinz, Benedikt Wismans. "Das 'elektronische Prüfungsamt' ". In: Wirtschaftswissenschaftliches Studium (WiSt), 1998
13. Pepijn R. Visser, Trevor J. Bench-Capon. "A Comparison of Four Ontologies for the Design of Legal Knowledge Systems". In: Artificial Intelligence and Law 6, pp.27-57, 1998.
14. Bernhard C. Witt. "Datenschutz an Hochschulen". LegArtis Verlag, 2004.