

Ontological Representation of Academic Programs

Richard Hackelbusch

Carl von Ossietzky Universität Oldenburg
Department of Computing Science
Oldenburg, Germany

Abstract—Using legal terminology, academic institutions release teaching and examination regulations to form the statutory framework of academic programs. This terminology is one reason why students often do not know how to satisfy the program requirements laid down by the corresponding institutions. This can result in needlessly long study times. Frequent changes of those regulations and parallel valid different regulations forming the statutory framework of programs leading to the same degrees may aggravate those problems. Furthermore, academic boards have to supply an amount of courses which fits the students' actual demand. This is a difficult task because there is only little information available for forecasting. In this paper, we present an ontology to handle these problems. It allows semantic representations of examination regulations and academic programs. Based upon this ontology, decision support systems can be implemented which can help students to decide how to satisfy the corresponding program regulations or which can help academic boards to forecast the students' demand on certain courses.

Index Terms—Ontology, Technology Enhanced Learning, Decision Support System, Examination Regulations

I. INTRODUCTION

Legally binding teaching and examination regulations form the statutory framework of academic programs. They are worded by using legal terminology. That's why they are often hard to comprehend. A result is that a lot of students even do not try to read it. A large demand on course guidance is another result (see [1]). Yet another reason is the prevailing heterogeneity of examination regulations. Examination regulations of single academic institutions forming the statutory framework of different academic programs can already differ a lot. This can be problem if courses of different programs must be integrated in a single curriculum (like a "minor subject"). *E.g.*, questions have to be answered which "external" courses of other academic programs can/must be taken in order to satisfy the corresponding program requirements, and if/how grades have to be annualized. Another problem is the fact that valid and different examination regulations can parallel exist forming the statutory framework of programs leading to the same degrees. This can happen, *e.g.*, after an introduction of a new version of examination regulations. That's why it is possible that students of the same academic institution who are leading to the same degrees, too, have to satisfy different program requirements.

These problems motivate the introduction of subsidiary documents like study guides that are intended to describe

possible variations of correct curricula. Those documents can be used as a basis for students planning their studies. A disadvantage of those documents is that they only describe situations in general. Often, they cannot comply with individual situations of students. In those cases, they do not bring a lot of benefits. Another attempt to solve the problems is the offering of individual study guidance (which is mandatory in some countries, *e.g.*, in Germany [2]) but which can be very expensive. Reformatations of academic programs which are implemented in the course of the so called Bologna-Process reduce the problems just conditionally. Often only the required minimum of the Bologna-guidelines are implemented, see [3] (like modularization, see [4]). In contrast, a result of the reformatations is that these above described problems now become obviously.

From the point of view of academic boards/from lecturers' point of view, there are some problems concerning examination regulations, too. Information about the current state of the students' studies is missing. That's why supply and demand of courses might be balanced adversely. Academic boards normally just reach information about the number of students who have started their studies in a certain term heading certain degrees under certain examination regulations, and — in some cases — how many students already have aborted/finished their studies. It can only clearly determined how many students are studying within a certain term. But it is not clear how far those students have reached satisfying their program requirements to reach their degrees. In some cases from the lecturers' point of view, it is also not clear how many students which want to take a certain course have to satisfy which program requirements — if more than one version exists. Then, the expected number of students who want to take a certain course can only be forecasted by stating the number of students studying in certain terms on knowledge of past terms (assuming that certain courses are taken by students studying in certain terms). That might result in situations in which there are too many or too little courses of certain subjects offered to fulfill the current students' demand. These problems do not only concern the task of deciding if a course should be offered. They also concern the task of deciding 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. Again, the expected amount of students can only be forecasted on a basis of insufficient information.

Until there isn't more information available concerning the progress of the students' studies, the corresponding

academic boards are not able to create an adequate supply to fit the student’s demand. Collecting and analyzing of that information have to be compliant with privacy guidelines (see [5]).

II. APPROACH

In order to be able to answer the questions asked by students as well as by academic boards as described above computer-assisted, representing examination regulations in a computer-readable language is one precondition. The relevant part of the examination regulations is the description of requirements to reach a specific degree. These requirements describe processes which represent possible curricula. Such processes, described by examination regulations, could be modeled, e.g., using Event Driven Process Chains (EPKs) introduced by ARIS (see [6]), Unified Markup Language (UML, see [7]) or Petri-Nets/Workflow-Nets (see [8]).

Advantage of approaches which allow a direct modeling of processes (in comparison to approaches which work exclusively on rule-basis, like [9]) is the more human-engineered way to model examination regulations. The way to convert those regulations into a computer readable model can be done easier and more intuitively.

Beyond the semantic representation of processes described by examination regulations, a further semantic representation of examination regulations is preferable. The definition of concepts like modules, workload, etc., should be modelable. That is one precondition to define, e.g., which modules can be used as substitutes for others. Such semantic representations are mostly difficult to model using classic process modeling approaches. That’s why our approach uses ontological concepts in order to allow the semantic representation of examination regulations and curricula. We call this ontology *Curricula Mapping Ontology* (CMO).

GRUBER defines *ontology* as “an explicit specification of a conceptualization” [10]. A conceptualization is an abstract model in a defined domain including the relevant identifying vocabulary, reaching consent within a certain community. “In such an ontology, definitions associate the names of entities in the universe of discourse (e.g., classes, relations, functions, or other objects) with human-readable text describing what the names are meant to denote, and formal axioms that constrain the interpretation and well-formed use of these terms”.

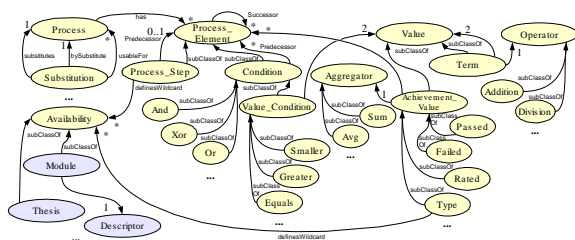


Figure 1. Curricula Mapping Ontology (CMO)

Our approach uses an ontology (the CMO) on concept level among other things as a metamodel which describe concepts to model processes as well as to model possible process elements (like modules, examinations, etc.). A concrete model of examination regulations can be done on

instance level of the CMO by shaping a precise process. The representation of concrete possible process steps offered by academic institutions (like supply of modules) can be done on instance level of the CMO, too. The interpretation of individual situations is done on application level of applications interpreting CMO-modeled academic regulations, like EUSTEL (see below).

Figure 1 shows the process representing part of the CMO on concept level. A process can be shaped using instances of the class *Process_Element*. These instances can be *process steps* or *conditions*. Each of them can have predecessors and successors. Conditions regulate the possibilities to be able to take certain process steps. Process steps are the actual achievements of a curriculum which must be reached by students.

Concrete examination regulations can be modeled on instance level of the CMO. To do so, elements of the corresponding curricula must be identified and modeled using instances of the class *Process_Step*. These instances must be modeled as a process identifying the predecessors and successors of them including instances of the class *Condition*.

Conditions can be simple logical terms (like AND, OR). They also can be comparisons of specific values, sometimes of a bigger complexity (like “modules successfully passed” ≥ 3) using the subclass *Value_Condition*.

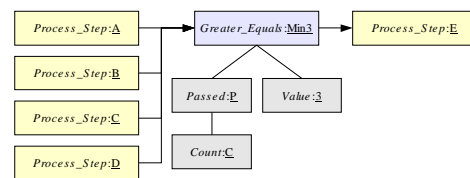


Figure 2. Curricula Mapping Ontology (CMO)

Figure 2 shows a simple process modeled on instance level of the CMO. It has to be interpreted that way that each Instance of *Process_Step* (which represents, e.g., a module in an academic program) can only be taken if it has no predecessor or if the predecessor (which either can be an instance of *Process_Step* or *Condition*) has been successfully passed. In that example only the taking of process step *E* is restricted by a specific precondition: *Condition Min3* has to be fulfilled. This condition is an instance of the class *Greater_Equals* (a subclass of *Value_Condition*). The interpretation of this class is that an instance is successfully passed if and only if the first value is greater or equals than the second value. The second value (*3*) is a simple instance of *Value* which has a numerical attribute with the value 3. The first value is a subclass of *Achievement_Value* (which is a subclass of *Value*). Depending on the type of the subclass of *Achievement_Value*, an instance “aggregates” a specific value of the set of process elements which are the predecessors of the corresponding instance of *Value_Condition* (in this example *A*, *B*, *C*, and *D*) using a certain aggregator. An instance of the class *Passed* has to be interpreted that way that it “aggregates” the Boolean value whether a process element has been successfully passed or not. One aggregator which can be used for this value is the *Count* aggregator, which “counts” the TRUE Boolean values. This type is used in

this example, too. Finally the condition Min3 can be interpreted that way that at least three modules of the set of A, B, C, and D have to be passed successfully in order to be able to take the process step E. That is a possible representation of the already above mentioned condition.

Using wildcards, it can be determined which elements (instances of the class *Availability*) are possible to take certain process steps of a curriculum. That can be, for example, wildcards for modules or theses. The concept *wildcard* means that those instances actually identify only the type or a part of the after all required attributes, where applicable. The actual offered elements (modules, *etc.*) can be assigned easily to applicable process steps of the curriculum on application level.

In addition, general regulations can be set on instance level of the CMO. These can be rules which regulate that, *e.g.*, one module can be substituted by two seminars. It can also include learned knowledge like allowed module substitutions. To do so, process substitutions can be defined using instances of the class *Substitution*. These instances can be used to connect certain process parts.

For lack of space, the representation of other aspects like grade scales, rules to successfully pass modules, free attempts, retaking or calculation of the final individual grade cannot be mentioned here, unfortunately.

In order to get a model of the corresponding examination regulations, for each degree a single ontology can be created on basis of the CMO using its concepts. The approach is validated using the ontology language OWL-DL^a and representing examination regulations for a couple of academic programs^b.

Currently, a decision support system using the concepts of CMO which is called *EUSTEL* is under development. It integrates the individual data of the students and the supply of modules. The system will be connected to the learning management system Stud.IP^c in order to allow students to plan their curricula at the same place where they already can check their individual results and the university calendar.

In particular, using *EUSTEL* students will be able to run through different settings of their individual curricula (*e.g.*, choice of primary subject, changing of primary subject, and choice of certain modules). One key element of the support by *EUSTEL* will be the possibility of visualizing the individual curricula and possibilities in continuing the studies calculating with certain settings of the corresponding curricula. Using *EUSTEL*, lecturers will be able to retrieve predictions of the demand for their lessons in certain terms — broken down to different examinations regulations applied for the corresponding demanding students.

^a <http://www.w3.org/TR/owl-features/>

^b For example „Diplomprüfungsordnung für den Studiengang Informatik der Carl von Ossietzky Universität Oldenburg vom 24.11.2004“, http://www.informatik.uni-oldenburg.de/studium/diplom/dpo2004/AM2004-05_14_DiplPruefOrd_Informatik.pdf and „Diplomprüfungsordnung für den Studiengang Wirtschaftswissenschaft an der FernUniversität in Hagen vom 09. Juli 1997 (in der Fassung 01. Oktober 2004)“, <http://www.femuni-hagen.de/FBWIWI/download/ordnungen/dpowiwi2004.pdf>.

^c <http://www.studip.de>

III. RELATED WORK

Approaches to offer computer-assisted decision support in questions of examination regulations are for example [9] and [1]. [9] exclusively uses rule-based representation of examination regulations and has no process view. [1] offers a process-based representation but only with very restricted possibilities to represent examination regulations. Semantic representations of the contents are not provided in both approaches. A support for academic boards is not provided, too. Approaches to support this target group mostly aim financial aspects, like [11]. Ambitious but to generic approaches which are intended to represent legal sources using ontological concepts, are for example [12] and [13]. *EUSTEL* itself is intended to be a part of the system described in [14].

REFERENCES

- [1] M. Gumhold, and M. Weber, “Internetbasierte Studienassistenz am Beispiel SASy”, in *doIT Software-Forschungstagung*, Fraunhofer IRB Verlag, 2003.
- [2] A. Reich, “Hochschulrahmengesetz”, Bock Verlag, 2005.
- [3] T. Hoffmann, “Vor dem Bachelor wird gewarnt”, in *Potsdamer Neueste Nachrichten*, October, 22nd, 2004.
- [4] Kultusministerkonferenz, “Rahmengesetzvorgaben für die Einführung von Leistungspunktsystemen und die Modularisierung von Studiengängen”, 2004.
- [5] B. C. Witt, “Datenschutz an Hochschulen”, LegArtis Verlag, 2004.
- [6] A.-W. Scheer, ARIS – “Modellierungsmethoden, Metamodelle, Anwendungen”, Springer Verlag, 1998.
- [7] Object Management Group, “Unified Modelling Language Specification, version 1.5”, in *OMG document formal/03-03-01*, 2003.
- [8] W. M. P. van der Aalst, and T. Basten, “Inheritance of Workflows: An Approach to Taking Problems Related to Change”, in *Theoretical Computer Science*, 270, pp. 125-203, 2002.
- [9] M. Heyduk, “Konzeption und Entwicklung eines universellen, frameworkbasierenden Studententutors”, master thesis, Carl von Ossietzky Universität Oldenburg, 1998.
- [10] T. Gruber, “A Translation Approach to Portable Ontology Specifications”, in *Knowledge Acquisition*, 5(2), pp. 199-220, 2003.
- [11] M. Goeken, and L. Burmeister, “Entwurf und Umsetzung einer Business-Intelligence-Lösung für ein Fakultätscontrolling”, in *Multikonferenz Wirtschaftsinformatik (MKWI)*, pp. 137-152, 2004.
- [12] L. T. McCarty, “A Language for Legal Discourse – I. Basic Features”, in *Proceedings of the second international conference on Artificial intelligence and law*, pp. 180-189, ACM Press, 1989.
- [13] A. Boer, T. van Engers, and R. Winkels, “Using Ontologies for Comparing and Harmonizing Legislation”, in *Proceedings of the ninth international conference on Artificial intelligence and law*, pp. 60-69, ACM Press, 2003.
- [14] R. Hackelbusch, “Handling Heterogeneous Academic Curricula”, in *Proceedings of the Seventeenth International Workshop on Databases and Expert System Applications (DEXA)*, pp. 344-348, IEEE Computer Society Press, 2006.

AUTHOR

Richard Hackelbusch is with the Department of Computing Science, Carl von Ossietzky Universität Oldenburg, D-26121 Oldenburg, Germany (e-mail: hackelbusch@uni-oldenburg.de).

Manuscript received September, 28th 2006.