Uni<form>: A prototype supporting curriculum description and course selection

Filippos Tsagkris[1], Christos Sintoris[1] and Nikolaos Avouris[1]
[1] University of Patras, Greece
filippos@tsagkris.com, sintoris@upatras.gr, avouris@upatras.gr

Abstract

This work studies the development of a prototype of a novel academic information management system that is intended for the secretarial staff of an academic program of study, offering a set of graphical tools through which the Curriculum can be described. These include constructs like academic periods, alternative specialization tracks, thematic grouping of courses, definition of rules that dictate how the students can define their program of study. Given the curriculum, a Rule Checker is also defined, which in turn, accepts students' course selection applications and examines whether they comply with the curriculum's ruleset, informing students accordingly. The prototype has been checked in a number of programs of studies and users.

1 Introduction

Academic Information Systems (AIS) allow for the interaction of university students with the program of study. They provide a set of functions, enabling various interactions between students and the facilitators of program implementation (Chi, 2009, Garrido et al., 2014). An important phase of student interactions with the program is the selection of courses for next period of study. Complex curricula often allow for a selection out of a rich pool of optional courses, for which several rules might apply. The work presented here focuses on the tools required for describing and amending university curricula and ensuring that students' applications conform to these curriculum rules.

Most AISs today do not offer a graphic language for Curriculum Description (CD) that can be used for supporting course selection. After inquiring major academic institutions in Greece (University of Patras, National Technical University of Athens, National Kapodistrian University), which all use different AISs, it was discovered that, curriculum description and updates often needed the intervention of the AIS development team, resulting in a gap between the application domain experts, i.e. the curriculum designers, and the developers implementing it. A second observation concerns the design of the systems' feedback during course selection by students, when the error messages were automatically generated and thus did not use simple and easily understood language, making it harder for students to locate faults in their course selection and succeed in planning their studies.

2 Uni<form> design

In the requirement gathering phase, 39 University curricula of major Greek Universities were analyzed. This resulted in the identification of the common abstractions that describe different curricula, which lead to the classification of these curricula as *course-centric* and *rule-centric*.

In the *course-centric approach*, the course selection rules are directly linked to the actual courses. It is the most common way of designing curricula rulesets and it consists of a set of compulsory courses, elective courses, and personal projects. These curricula can offer more than one specialization tracks, as well as define a cap on the maximum number of courses that can be selected.

The *rule-centric approach* is comprised of high complexity rules, which are usually related to the specialization tracks of the curriculum. Its fundamental characteristic is that it poses a set of minimum requirements that must be met, in course selection, trading the increased freedom offered to students in course selection with a highly complicated ruleset.

The analysis of existing rulesets led to the observation of patterns, based on which, abstractions were made that lead to the definition of three archetypal forms of course selection rules.

Summation rules. They are very common in most rulesets, allowing for counting courses based on some characteristics, making possible the enforcement of rules concerning compulsory or elective courses, with optional lower and upper limits: $n_p \leq \sum_{i=1}^{v} C_{ip} \leq m_p$ [1] where C_{ip} a course i of period p and n_p, m_p the lower and upper limit of number of courses for the specific period p.

Course correlation rules. Two types of course correlations are identified, a) prerequisite courses, requiring that students have successfully passed one or more courses, before attending another one $C_{ip} \rightarrow C_{jk}$, b) corequisites, requiring the simultaneous attendance of two or more courses $C_{ip} || C_{jp}$

Limit rules. These require that the sum of a quantitative characteristic of the total number of courses a student is enrolled in, during a period of study, does not exceed a certain limit. This kind of rule may require that the courses applying for, follow a specific order. This rule is similar to rule type [1], setting upper bound limits on a course specific measure, most usually credit units.

The Uni<form> prototype is made of two subsystems, one (CD) used by the secretarial staff for describing curricula through rulesets, and Rule Checker (RC) for students to submit their course enrollment applications. They both follow a client-server architecture supporting asynchronous exchange of XML data. More specifically, the CD module submits the curriculum as the DOM built by the secretariat personnel. The analysis is performed on the server side of the system, converting the submitted ruleset to a custom construct of curriculum description objects. Regarding the RC subsystem offered to students for submitting and checking course enrollment applications, it offers the service through a REST API, thus allowing for its functionality to be mapped to existing institutional AISs.

3 Uni<form> scenario of use

A typical scenario of use of the CD module is made of a sequence of guided steps. These concern the following Curriculum characteristics definition: [phase 1]: Curriculum meta-data: academic years over which the curriculum is applicable, number of periods of study P, names of specialization tracks ST and their extend over periods of study, [phase 2] Description of individual courses C, including the typical period(s) of study in which they are offered, credits, etc. [phase 3] Grouping of courses. Each Group $G_i = \{C_{ip}, C_{jk}, ...\}$ contains a number of courses across different periods of study and is related to specialization-tracks, [phase 4] Rules Definition. For each rule, the appropriate rule violation error messages are defined in a human-readable form. The defined rules belong to different rule patterns, e.g. summation rules are defined through upper and lower limits, scope of applicability to groups of courses, etc. (see fig.1). In similar fashion other types of rules, described in section 2, are defined.

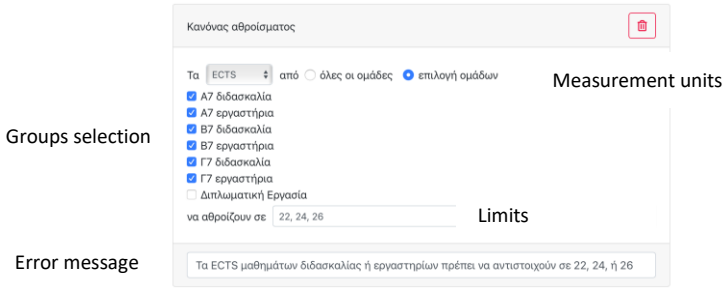

Figure 1: An annotated view of a summation rule definition wizard

4 Evaluation & Findings

The evaluation of the two offered subsystems was performed as part of this study. The CD subsystem was tested by inserting the curricula of all 35 undergraduate programs offered by the University of Patras. The usability of the system was also measured using the heuristic evaluation technique with the popular set of Nielsen's 10 empirical rules (Nielsen, 1995). It was found that due to the extensive use of scrolling for navigating in the system, sometimes editing different periods of study can feel slow. The lack of an always visible marker could also trip the user in editing a different period than the one desired. Targeted tooltips needed to be added for easier onboarding of new users.

Regarding the RC subsystem, two end-user evaluation techniques were applied, the think-aloud protocol (Lewis, 1982) and a semi-structured interview. Six users participated in the evaluation, each one taking part in two scenarios, for different levels of study. The main observation emerging from this study is that the users found error messages particularly informative and all users agreed that the system was by far easier to understand and use, than the one offered by their university.

However, some interesting new requirements also came out of this process, mainly about summative information, so that users do not have to perform mental calculations during the course selection process. It also became apparent through observation, that the same concerns noted earlier about the navigation and extensive scrolling requirements, applied here too.

In conclusion, the main goal of this work was the development of a system that bridges the gap between the description of a university curriculum and the implementation of a platform for checking and advising students' course enrollment. For this to be achieved, many curricula from Greece's major academic institutions were analyzed. Uni<form> was designed based on the findings of this study. The derived prototype was evaluated through a user study, highlighting improvements to the experience of both the students and university secretarial staff.

5 References

Garrido, A., & Morales, L. (2014). E-learning and intelligent planning: Improving content personalization. *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje*, *9*(1), 1-7.

Lewis, C. (1982). *Using the" thinking-aloud" method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

Nielsen, J. (1995). *10 usability heuristics for user interface design*. Nielsen Norman Group, 1(1).

Chi, Y. L. (2009). Ontology-based curriculum content sequencing system with semantic rules. *Expert Systems with Applications*, *36*(4), 7838-7847.

Author biographies

Filippos Tsagkris is a recent graduate of the integrated master's (MEng) program of the Dept. of Electrical and Computer Engineering at the University of Patras. His main research interests lay in the area of formal representation and validation of complex systems like the Curriculum Description and Rule Checker modules of Uni<form>.

Dr. Christos Sintoris is a Teaching and Research Associate at the Dept. of Electrical and Computer Engineering at the University of Patras. He has been a member of the University's Human-Computer Interaction Group for over ten years. He has a Computer Science background and his research interests involve human-computer interaction, location-based mobile interaction.

Prof. Avouris is professor of Human-Computer Interaction in the University of Patras. He has experience in industry and academia in Greece, UK and Italy, in the area of Human-computer interaction and design of interactive technologies. He is currently expert member of IFIP Technical Committee 13 on Human-Computer Interaction (HCI) and steering committee of INTERACT conference.