Guided Exploration of the Domain Space of Study Programs

Recommenders in improving student awareness on the choices made during enrollment

Vangel V. Ajanovski
Ss. Cyril and Methodius University
Faculty of Computer Science and Engineering
Rugjer Boshkovikj 16, P.O. Box 393
Skopje 1000, Macedonia
vangel.ajanovski@finki.ukim.mk

ABSTRACT

This paper demonstrates a solution for increasing the awareness of students with the many choices they have during course enrollment in an integrated university, and how the choice they make can impact their future studying. Three separate navigational tools are presented that have been developed to work in symbiosis as a single application to help students – a tool for exploration of the study programs at the university as a whole, to be used as a guide through the many choices at the university; a tool that helps with management of prerequisites of the offered course-curricula in order to be aware of the impact of failing or not succeeding on time with critical courses; and a tool that can generate an initial personalized future study plan for each student that she can later modify. Recommenders are used as service behind all the three tools, in order to annotate options and possibilities that might be helpful to the student, offer alternatives to well-known popular choices and make more informed choices.

CCS CONCEPTS

• **Information systems → Social recommendation**; • **Human-centered computing → Social navigation**; *Visualization*; • **Social and professional topics → Model curricula**; *Computing education programs*;

KEYWORDS

course recommendations; curricula guidelines; course enrollment; social navigation

ACM Reference format:
Vangel V. Ajanovski. 2017. Guided Exploration of the Domain Space of Study Programs. In *Proceedings of Joint Workshop on Interfaces and Human Decision Making for Recommender Systems, Como, Italy, August 27, 2017 (INTRS Workshop),* 5 pages.
DOI:

1 INTRODUCTION

Integrated universities today have many departments and ever-increasing number of study programs and possibilities for the students to pursue. While it may seem that the freshman students are the ones who should be confused the most, in fact such is the case also with the older students. The magnitude of choice especially creates the biggest problem for those who are enrolling the third or upper semesters, when many possibilities for elective courses open. As an example, the department of the authors manages 4000 undergraduate students that all study computing and offers around 300 courses, organized in 8 study programs. Only around 60 of the courses are mandatory, while all the rest are elective choices that are considered free to choose and open-up in specific points in time during studying. Add to that courses offered freely from other departments throughout the whole of the university and it is easy to get to a number of several hundreds of possible courses to enroll each and every semester. Usually students are much too busy and don't have the time to read fully the course syllabus for all available courses (have in mind that their can be 500 of them) or check all the web-sites of all professors. So they usually rely on word of mouth, ask for opinions on internet forums and social networks to get more information. But, it is impossible for anyone to personally know all the specifics of all the courses on offer, and to give relevant recommendations to a student.

There is another issue in larger institutions. Not all courses are entirely free to enter, because of prerequisites and other constraints that have to be fulfilled in terms of knowledge or competencies that should have been acquired in the past. While it may seem as a simple requirement, it can create substantial issues. The graph of prerequisites, when created at the scale of a whole department with many active study programs and many courses on offer is a rather big graph that ranges in hundreds of nodes and thousands of links, so it is impossible to know it all by heart. The implication of this is that when a student fails an exam, certain possibilities close temporarily or permanently, and whole branches of choices become blocked, while the student is not aware.

Some institutions deal with these issues by employing teachers as student advisers. The scale of the problem makes it prohibitive at larger institutions. The author believes that introduction of recommenders to make a personalized assessment of which courses would be better suited to follow-up for each student could help, as it has been established by many researchers in the past (see [8], [12], [9], [4],[10],[5],[6] and [7]). The goal of this research was in fact not in the construction of new algorithms, nor evaluation of existing ones, but a development of a flexible system that would enable introduction of recommenders in many situations, especially in the term enrollment process and enable general exploratory navigation by students when investigating future possibilities.

INTRS Workshop, Como, Italy

DOI:

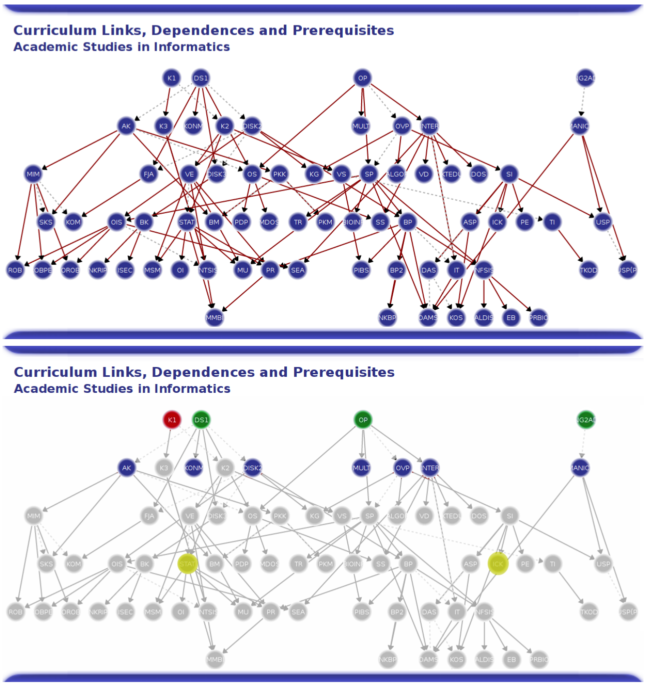

Figure 1: Real-time visualization of the graph of curriculum links. a) Management view. b) Term enrollment view for students, showing the status at the given moment.

2 PREREQUISITE MANAGEMENT

The problem of managing prerequisites was tackled with an introduction of a real-time graph visualization tool that is used to both investigate the existing inter-dependencies, and to assess their impact. Since the graph of links between curricula is almost never planar, and some link types are reversed so will create cycles in the full graph, it is understandable that a fully automated solution that gives good overviews is not always a possibility. Therefore, the author opted for a semi-automated tool that gives as good as possible initial layout that can be later tweaked to better understand it. The graph visualization tool was inspired from [13]. The graph is constructed initially by topological sorting the curricula (similar to [11] and is then modified with the help of force-direct auto layout using D3.js.

An example that shows two types of links for just a single study program, can be seen in Fig. 1a. When used by Curriculum Managers such a visualization is useful to investigate issues in the defined constraints, delete some of them or add new ones. The graph should be read from the top towards the bottom. The top (first) tier are courses which have no prerequisites and can be enrolled as early as possible (in the first term). In the second tier one can see courses that can only be enrolled after the respective prerequisites from the first tier are finished, so at earliest possible would be to have them in the second term, and so on.

Several constraint types are important to note: *Hard Prerequisite* – two curricula that should be enrolled one after the other was passed with success; *Soft Prerequisite* – a passing grade is not strictly required, but it is recommended; *Parallel* – two curricula that should be enrolled at the same time; and *Equivalence* – when two curricula are designed in such way so that the acquired knowledge and competencies would be considered the same or equivalent.

The same interface is used by students and it operates a bit differently when accessed by students. They can use it to assess their current status and the effect the constraints have on their future enrollments. Red and green are used to represent the past – green means that a course was finished, red means that it was failed and it will block others, so the student will need to enroll it once again. During enrollment time, blue is used for the nodes that are free for enrollment – when the student has passed all the respective prerequisites. All the nodes that are blocked by some prerequisite will turn gray.

The tool initially only shows mandatory links, where the output from a recommender would not be useful, since all those courses will have to be enrolled at one time or another. But this tool can be used to visualize also additional courses, that are not part of the mandatory set. So when adding chosen set of electives (or recommended ones) to the visualization, they will pull-in all the prerequisites with them, so that the recommender is able to show if some the courses have received any kind of recommendation and indicate them respectively on the graph. They will be marked with a yellow or orange hue, depending on the accessibility so that the student can easily understand which of the added curricula are recommended to her and what prerequisites will be needed to accomplish in order to be able to enroll them. (see Fig. 1b)

3 EXPLORATION OF STUDY PROGRAMS

The introduction discussed the problem of multitude of choice, and for this the author has developed a separate tool within the system, that enables the exploration of all the active study programs within the university, so that students can discover about possible new courses to enroll, learn about the context in which those courses are offered and be aware of all the details on the course curriculum.

It is important to focus that the solution acknowledges that an institution can frequently change all of its study programs and introduce complete new accredited study plans. Due to the flexibility of studying, which is something that should be taken care for, each student is allowed not to switch from the original plans she was admitted into for 8 years. This means that several variations of a study program might be active, from one study plan revision to the other, depending on the number of student admitted at the times that those revisions were considered current.

As the study programs are still active, courses that only existed in past revisions and were later planned to be revoked will be still offered for the older students, so as a result these will be open to newer students too if all prerequisites are fulfilled. Having this in mind, the solution offers the possibility to both explore the current study programs and the offered courses that are active from the accredited plan revision that the logged in student was admitted, and to explore the older and what's most important – newer revisions in case a curriculum reconstruction project occurred after admittance.

In Fig. 2, a similar view is presented at the level of the current (or chosen to investigate) study plan revision, showing all the active study programs with all the courses on offer. Hereby, one can see

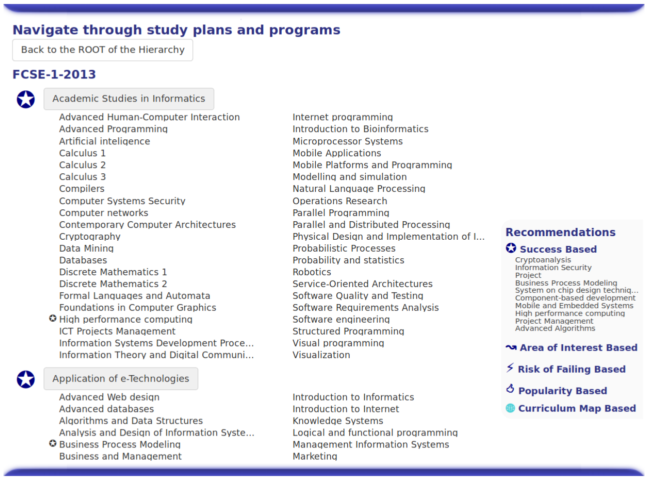

Figure 2: Exploration through active study programs, within the respective study plan revision that the student is admitted to. The programs that contain recommended courses are noted with a star, as are the respective courses.

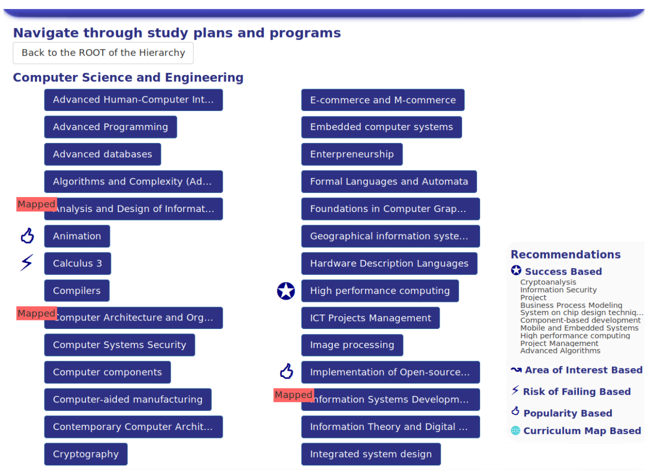

Figure 3: Exploration through the courses open for enrollment within a chosen study program. Recommender engine outputs are used to indicate recommended possibilities for the student based on various aspects.

markings on both the study programs that include courses from the recommendations list, and the respective courses themselves.

If the student decides to drill-down for more information, she can open the respective study programs and see the type of recommendations or annotations that each course has. In Fig. 3 the whole structure of the study program is presented.

As can be seen on all the figures, the lists are always alphabetically sorted in order for the student to easily list all of them and not to impose certain bias. The student can search and filter to find what she wants and the recommended courses are not grouped together in a single place, so that it is not implicitly stated that those courses are considered as the best possible options. The intention

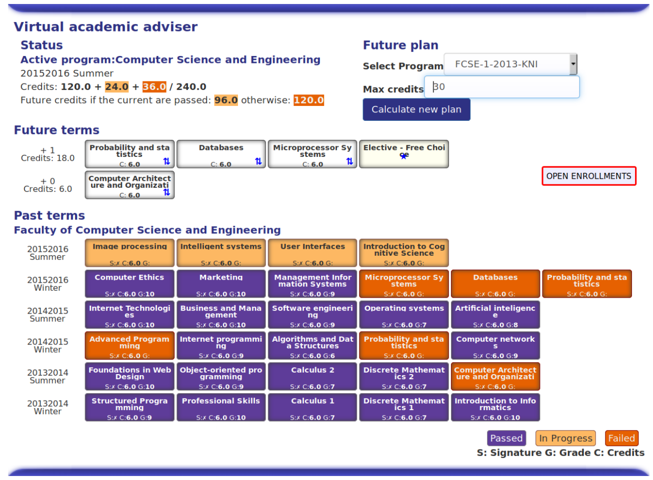

Figure 4: The virtual academic adviser shows the student's enrollments in the past and possible scenarios for the future depending on expected workload, course prerequisites and when are certain courses expected to be offered.

is to have the recommender markings only serve towards better awareness, while still showing all the other choices to the student.

4 VIRTUAL ADVISER AND ENROLLMENT

The third view that the students benefit from is the virtual academic adviser, that as a tool was presented in the past. The tool is a visualization tool that tries to predict what will happen in the future if the student continues studying with the same pace as before. The future plan is based on the chosen study program, pace of studying in number of credits per semester, while taking into account in which semesters the courses are offered and the prerequisites needed to accomplish in order to enroll such courses.

The tool is interactive, so that the student can reorder the courses as wished (if constraints allow it), and create an alternative plan that seems better fitted, depending on her real-life plans and situation. The student can also try what-if scenarios to investigate what will happen if she decides to pause a semester, go to another department as part of a student exchange program, or switch to a newer study program. See Fig. 4.

When satisfied with the generated plan, the student can directly proceed towards enrollment, so that the courses that were in the first future semester will be listed for enrollment, and she can go on with the rest of the administrative issues. While all the other interfaces can be considered as exploratory prototypes used for research only, and were not yet put into production the rest of the official administrative part of the enrollment process is a production ready module that has been in production for several years.

5 SYSTEM ARCHITECTURE

The system was envisioned to have three lightly-coupled parts, as separate subsystems, in order to be able to change the infrastructure or add additional modules.

The central system (see Fig. 5) consists of several subsystems that are of significance. The two most important parts that sit behind

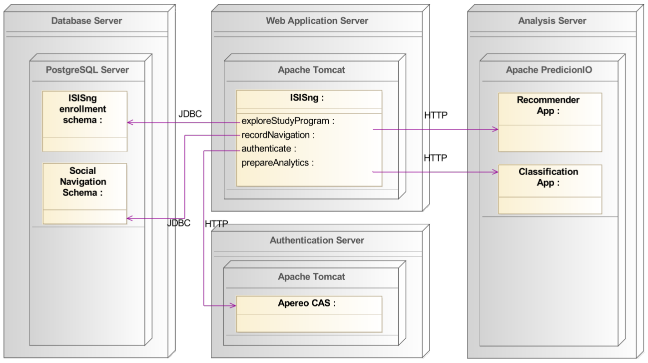

Figure 5: Deployment diagram of the complete solution.

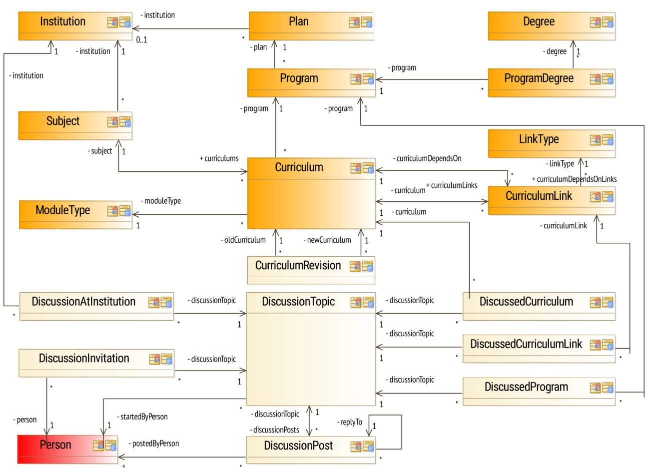

Figure 6: Model of the study programs for a whole university, enabling the tracking of evolution among revisions.

and enable the operations of the system overall are the curriculum management subsystem and the enrollment subsystem. The curriculum management subsystem is used to define the structure and the evolution of the structure of all the study plan revisions, all the study programs accredited in each revision and all the course curricula on offer for each study program and revision, for each semester. See Fig. 6 The enrollment subsystem is used for the formal course enrollment process.

The discussed navigational, exploratory and recommender components are lightly-coupled to the central system. The navigational subsystem has the role to have a record on all interactions that each student does with all shown element. For this a modified replica of the structure of the study program is kept, solely for purpose of recording navigation actions. This replica is in fact a simplified copy of the structure of the study plans, annotating only the navigation structure as shown in the user-interface (list of top elements, lists of sub-elements, etc). These are mapped to the original objects so that it is possible to understand that a certain user interaction with some user-interface list element relates to certain real-world concept (curriculum, program or plan). See Fig. 7.

The idea behind the replica-and-map type of solution is to detach the exploratory social navigation part of the interface from the real-world complex structure behind in order to be able to easily experiment with different navigational interfaces, easily reconstruct the navigation and be able to do it in several independent exploratory web and mobile applications apart from the real-world enrollment application. Developing it as a separate application and separate data model enables the concept to work even when the enrollment system is not active or it does not exist, so that the navigational part can be used independently and students can experiment with these presented interfaces without formal implications to their official enrollments.

The navigation model is easier to use for feeding data to the recommender engines that sit behind. As seen on the architecture model in Fig. 5, Apache PredictionIO project was used to implement and host the various analysis engines and communication is via the REST API to each separate PredictionIO application.

There are four PredictionIO-based recommender applications that use different algorithms and are fed the same data, but issue separate outputs. These outputs are all gathered in the same place and used in the navigation interface to indicate various icons or embellishments in the exploration interface.

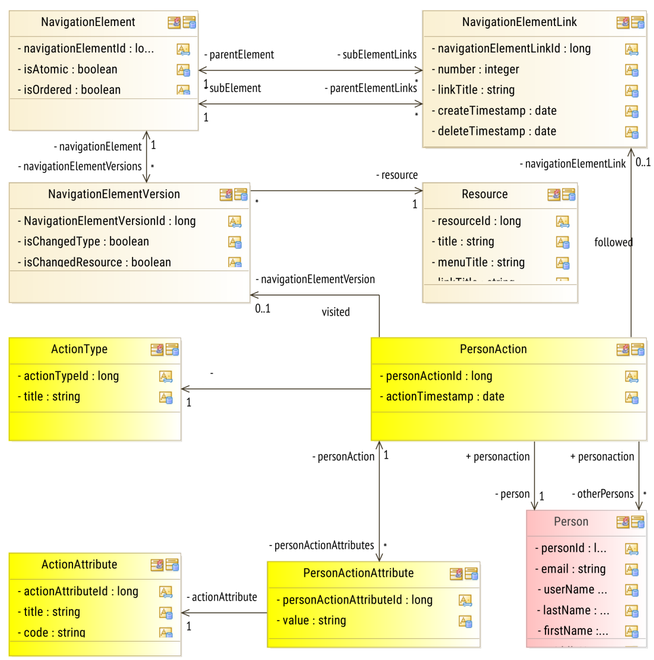

Figure 7: The navigation model tracks user interactions.

The course enrollments of the students in each term (student, curriculum, dateenrolled) are gathered to analyze real interest in certain curricula.

The exam grades of the students (student, curriculum, grade, dateexam) are gathered as an indirect way of measurement of the student rating of a course. Although it might generally be possible to use official surveys measuring the quality of the teaching process and student satisfaction, the author's institution anonymizes the data in such surveys (in order to protect students' identities) and it was not possible to rely on such data. So, the grades received during exams are used to create a personalized assessment of the satisfaction with courses.

The failed exam data (student, curriculum, grade, datefailed) are also used in a reverse rating scenario in order to create a personalized assessment which courses could introduce risk or can be considered critical (where the student has higher probability to fail). Standard criticism of the practice of indicating such courses is that the students will avoid them. Contrary to that, the author believes that the students should be transparently informed that some courses might be harder to pass, which in general relates to the fact that such courses might require more effort from the student to pass. This is useful information when planning the work load in a future term and the student can use this information to adjust the virtual academic adviser to a workload with lower number of credits then usual for terms in which the student is enrolling critical courses. Enabling the student to plan ahead for extra effort may lead to increased likelihood of success and overall satisfaction in the end without avoiding such courses.

There are other possibilities. The grades from weekly homework assignments and grades from exams per topic, can be used to look-up specific topics within a curriculum mapping that a student is excellent in, and use those topics to find other relevant courses in which the student would benefit from the already acquired knowledge. The curriculum mapping effort is part of another line of research in the project and was already presented in [2] and [1].

The architecture of the system was designed in this specific way, in order to enable an additional use-case – having many different navigational structures and different navigational applications, with separate recording of user interactions and separate feeds to the recommender engines behind.

6 CONCLUSIONS AND FUTURE RESEARCH

The implemented system offers several alternative interfaces that can be used during term enrollment, all for the single reason - enable the student to make a more informed choice of courses. The system architecture enables for the system to independently store all the users actions throughout the history, and provides them as data output to the configured recommendation engines. The system architecture enables the usage of many recommendation engine outputs (based on the same database analyzed with different algorithms, or based on different views of the same database analyzed with the same algorithm) in order to give recommendations to the student on various aspect of interest. This enables the student to assess her progress, explore possibilities, learn more about the study programs and try what-if scenarios to see what could happen in the future, based on the current choice.

Future efforts will be focused towards introduction of career path recommendations, so that the student's status, acquired knowledge per area/topics, and competencies will be used to evaluate past success and propose streamlined recommendations towards career goals.

ACKNOWLEDGMENTS

This work is a result of the project ISISng[3], partially financed by the Faculty of Computer Science and Engineering.

REFERENCES

[1] Vangel V. Ajanovski. 2017. Curriculum Mapping As a Tool for Improving Students Satisfaction with the Choice of Courses. In *Proceedings of the 2017 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '17)*. ACM, New York, NY, USA, 76–77. DOI: https://doi.org/10.1145/3059009.3072978

[2] Vangel V. Ajanovski. 2017. Information System for Mapping the Coverage of Reference Curriculum Guidelines in the Teaching Curricula of a Higher-Education Institution. In *Proceedings of the 14th International Conference for Informatics and Information Technology (CIIT 2017)*. Faculty of Computer Science and Engineering, Mavrovo, Macedonia, (in press). http://ciit.finki.ukim.mk

[3] Vangel V. Ajanovski. 2017. ISISng: Integrated Student Information System of the Next Generation – Official Project Website. (June 2017). https://develop.finki.ukim.mk/projects/isis

[4] Mohammed Almulla. 2009. School e-Guide: A Personalized Recommender System for e-Learning Environments. In *Proceedings of the First Kuwait Conference on e-Services and e-Systems (eConf '09)*. ACM, New York, NY, USA, 2:1–2:5. DOI: https://doi.org/10.1145/1836029.1836031

[5] Fábio Oliveira Garrido Carballo. 2014. *Masters' Courses Recommendation: Exploring Collaborative Filtering and Singular Value Decomposition with Student Profiling.* MSc Thesis. Instituto Superior Técnico, Lisbon, Portugal. https://fenix.tecnico.ulisboa.pt/downloadFile/563345090413333/Thesis.pdf

[6] Martina Holenko Dlab and Nataša Hoić-Božić. 2016. Increasing Students' Academic Results in e-Course Using Educational Recommendation Strategy. In *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016 (CompSysTech '16)*. ACM, New York, NY, USA, 391–398. DOI: https://doi.org/10.1145/2983468.2983506

[7] Asmaa Elbadrawy and George Karypis. 2016. Domain-Aware Grade Prediction and Top-n Course Recommendation. In *Proceedings of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, New York, NY, USA, 183–190. DOI: https://doi.org/10.1145/2959100.2959133

[8] Rosta Farzan and Peter Brusilovsky. 2006. Social Navigation Support in a Course Recommendation System. In *Adaptive Hypermedia and Adaptive Web-Based Systems: 4th International Conference, AH 2006, Dublin, Ireland, June 21-23, 2006. Proceedings*, Vincent P. Wade, Helen Ashman, and Barry Smyth (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 91–100. http://dx.doi.org/10.1007/11768012_11 DOI: 10.1007/11768012_11.

[9] Jamil Itmazi and Miguel Megías. 2008. Using recommendation systems in course management systems to recommend learning objects. *International Arab Journal of Information Technology (IAJIT)* 5, 3 (July 2008), 234–240. http://www.ccis2k.org/iajit/PDF/vol.5,no.3/4-169.pdf

[10] Youngseok Lee. 2011. An Intelligent Course Recommendation System. *The Smart Computing Review* 1, 1 (Oct. 2011), 69–84. DOI: https://doi.org/10.6029/smartcr.2011.01.006

[11] John G. Meinke and Charles R. Bauer. 1976. Topological Sorting As a Tool in Curriculum Planning. *ACM SIGCSE Bulletin* 8, 3 (July 1976), 61–66. DOI: https://doi.org/10.1145/952991.804757

[12] Michael P. O'Mahony and Barry Smyth. 2007. A Recommender System for On-line Course Enrolment: An Initial Study. In *Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys '07)*. ACM, New York, NY, USA, 133–136. DOI: https://doi.org/10.1145/1297231.1297254

[13] Ron Zucker. 2009. ViCurriAS: A Curriculum Visualization Tool for Faculty, Advisors, and Students. *Journal of Computing Sciences in Colleges* 25, 2 (Dec. 2009), 138–145. http://dl.acm.org/citation.cfm?id=1629036.1629059