COURSR2: An Integrated Time Management System for Lifelong Learners

Konstantinos Agnantis

Department of Applied Informatics, University of Macedonia, Egnatias str. 156
Thessaloniki, Greece
kagnadis@gmail.com

Anastasios Alexiadis

Department of Applied Informatics, University of Macedonia, Egnatias str. 156
Thessaloniki, Greece
talex@java.uom.gr

Ioannis Refanidis

Department of Applied Informatics, University of Macedonia, Egnatias str. 156
Thessaloniki, Greece
yrefanid@uom.gr

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

This article presents COURSR2, an intelligent system aiming at managing the personal time of lifelong learners in an integrated way. Lifelong learners usually try to combine learning with their professional and family duties. Taking in advance informative decisions about whether they have the necessary time resources to devote to a learning object is crucial, both for psychological and for financial reasons. COURSR2 encompasses information for all temporal aspects of a learning object. For example, a course may have lectures, with cardinality constraints over attending them, labs, mid-term exams, final exams, homework, projects and home reading. The user can also represent his non-educational activities, using the general purpose ontology of the SELFPLANNER intelligent calendar application. By subscribing to a learning object through COURSR2, the scheduling engine of SELFPLANNER's is employed to find viable plans, allowing rescheduling of existing activities. Thus, the learner is able to perform what-if analysis about whether a set of learning objects is compatible with each other, as well as with his other commitments. The latest version of COURSR2 has been deployed and evaluated using real users.

Keywords: Learning Objects; Intelligent Calendar Applications; Learning Management Systems; Ontologies.

1. Introduction

Lifelong learning is nowadays the best way to stay alive in the ever-changing knowledge labor market, with good prospects of successfully targeting well paid jobs. Science, technology, and finance are changing rapidly, so educated people need to update their knowledge or even focus on different areas. There are many opportunities for a lifelong learner to get more education: Traditional university programs, open universities, the recently popular massive online open courses (MOOC), books, as well as a vast amount of educational material available over the web (videos, tutorials, online books, etc). All these are difference types of learning objects with different characteristics. Some of them are free, others not. Some of them require physical presence of the learner, whereas other are totally online. Some of them require synchronous attendance, with others being totally asynchronous. However, there is one characteristic that all forms of learning objects share: they require a significant amount of the learner's time.

According to our experience (the last author has more than 10 years of experience in teaching lifelong learners at open universities), the lack of the necessary available time is the most important reason why a lifelong learner may fail in successfully achieving his goal wrt a particular learning object, which is conquering the involved knowledge or getting the related certificate (other factors, not investigated in this article, include overestimating the learner's prerequisite cognitive skills for the particular learning object or not possessing the necessary knowledge background for it). Particularly, it is quite a frequent phenomenon that a learner enrolls simultaneously in many learning objects (e.g., classes of an open university) and, eventually, fails to complete some of them, thus being forced to re-enroll and complete the rest of them at a later time. Such failures are painful in three ways: First, the learner feels disappointed for failing to achieve his goal. He might blame for the failure himself as of having limited capabilities, generally or for the specific subject, or as of taking bad decisions. This disappointment may affect other aspects of his life, thus having a broader psychological impact to him. Second, attending a learning object may involve paying some fee, which is not refundable in case the learner was not successful with it. Third, the learner has spent some time with the learning object before realizing that he does not possess the necessary resources to conquer it. Although he gets some knowledge from the time invested in the learning object, the ultimate goal is not achieved, so the invested time could be considered wasted.

So, it is crucial to take informative decisions as of whether the learner has the necessary time to conquer successfully a learning object. The necessary information involves all the learning objects the learner wants to take concurrently, as well as other commitments he has in his professional or personal life. For any activity related to the learning object, the decision maker needs to know a plethora of information details, in order to schedule it within the learner's calendars without interfering with other activities. This information includes the duration, the temporal domain, the alternative locations that are compatible with performing the activity, whether it is obligatory or not, whether it is preemptive or not, the utility gained by successfully performing the activity, the learner's preference over the way the activity can be scheduled, etc.

COURSR2, the intelligent system presented in this article, tackles exactly the problem of taking informative decisions as of whether a learner has the necessary time resources to devote to a set of learning objects, taking into account all other non-educational commitments and integrating all this information in the most flexible way. COURSR2 integrates two distinct ontologies, a general one concerning any type of activities, and a more specific, concerning learning objects. Furthermore, it possesses information about a variety of learning objects. With all this information, COURSR2 employs a powerful scheduling engine that is able to produce near optimal plans for problems with dozens of activities.

A typical scenario involves the learner investigating whether he is able to commit to another learning object, whereas he has already several other educational and non-educational commitments. In order to accommodate the new commitment within existing ones, COURSR2 tries to reschedule existing commitments, including educational and non-educational ones, provided that this is allowed by the user.

This article builds over our previous work, COURSR[1], by integrating educational and non-educational commitments in a smooth and flexible way. Particularly, the earlier version of the system was not able to reschedule non-educational activities or already scheduled educational ones. Furthermore, the article introduces an expandable architecture, adopted for the first time by the COURSR2 system, that allows in the future to handle simultaneously (using separate dedicated applications) other types of activities (e.g., leisure activities), together with educational and general activities. COURSR2 has also many new features that enhance its usability, like location based scheduling, more intuitive ways to import required information, better integration with Google Calendar services, redesigned user interface, etc. Finally, this article presents an empirical evaluation of the system, involving predefined scenarios and real learners and instructors.

The rest of the article is structured as follows: Section 2 presents background and related work. Section 3 presents COURSR2 in depth. Particularly, it presents the typical use cases for the two types of users, that is, the learning object owner and the learner; it continues with the system's architecture; it discusses scheduling issues; and, finally, it gives an overview of the underlying ontology for describing the temporal aspects of learning objects. Section 4 presents the empirical evaluation, whereas Section 5 concludes the article and poses future directions.

2. Background and Related Work

COURSR2 is related with two directions of research. The first concerns learning management systems (frequently abbreviated as LMS) and the second concerns intelligent calendar applications.

A learning management system[2, 3] helps teaching and administrative staff of an educational organization to deliver learning objects to students, track their use, assess the students' performance, facilitate communication between all involved parts, and many more. From the functionalities usually implemented by a LMS, the most related to COURSR2 is the ontological representation of learning objects. The IEEE 1484.12.1 – 2002[a] standard for Learning Object Metadata (LOM) is even nowadays the most widely used ontology for describing learning objects. However, no provision exists in this ontology, as well as in several variations of it, to describe the temporal aspects and requirements of each learning object at a fine-grained level of detail, capable of being exploitable by schedulers.

There is a lot of existing work in applying AI planning to develop learning paths, using semantically annotated learning objects. Castillo et al.,[4] using AI planning,

[a] https://standards.ieee.org/findstds/standard/1484.12.1-2002.html

developed a system capable of designing learning paths, by automatically extracting HTN planning domains without the need of planning experts. By accurately labeling learning objects using well-defined metadata, teachers can provide all the required information (e.g. content dependencies, ordering relations, user profile) allowing the system to auto-generate a learning plan for the student.

A similar approach has been followed by Garrido et al.;[5] Using metadata about learning objects, they created a system that generates learning plans based on prerequisites and goals previously defined by a teacher. Additionally to planning auto-generation, they also focus on intuitiveness and usability, providing a set of UI tools that can facilitate the teacher in the process of specifying the information needed by, and help students to have a better visual understanding of the route they are required to follow in order to succeed their goal.[6]

Limongelli et al.[7] describe a module that works as a centralized search engine, enabling teachers to retrieve Learning Objects from proprietary Learning Object Repositories (LOR) using metadata searching techniques and trying to introduce social information as well (like the popularity of a LO) in order to succeed more sensible search results. The specific module is part of Moodle LMS.[8]

COURSR2, contrary to a LMS, does not provide learning objects and, contrary to LOM, does not focus in describing such aspects of learning objects. Of course, COURSR2 could use LOM in order to facilitate looking for learning objects. However, other methods of searching, such as based on the institution and the particular academic program, could be used as well. Nevertheless, the functionality exposed by COURSR2, coupled with LOM, could be exported to or integrated in a LMS system to provide a holistic approach.

Closer to the COURSR2 attempt to measure the time requirements of a learning object is the European Credit Transfer System (ECTS)[b]. ECTS is designed to measure the total effort a student needs to invest in order to pass successfully a learning object (usually a university class). An ECTS credit equals about 25 work hours. So, a university class with 5 ECTS credits requires about 125 work hours. This total effort includes attending lectures, doing homework, reading, giving exams and every other activity related to the learning object. However, the ECTS system does not specify the distribution of this total workload among the various activities of the learning object and, more importantly, the detailed temporal and ordering constraints of the involved activities. So, it cannot be used for detailed what-if analysis.

As an example, attending four classes with 5 ECTS credits each within a time period of 17 weeks implies 500 hours of work in total, or about 30 hours of work weekly, which seems feasible. However, by looking at the details of these classes (e.g., the specific timing of the lectures or of the final exams), as well as at the other learner's commitments, several collisions might arise that render attending all these classes during the same time period infeasible.

[b] http://ec.europa.eu/education/ects/ects_en.htm

Assuming now that all this detailed information of the temporal requirements of a learning object is available, a scheduler is needed that, having access to the non-educational commitments of the learner as well, is able to provide optimal or near optimal plans. COURSR2 employs the scheduler provided by the SELFPLANNER[9,10,11] intelligent calendar system and implements in an ad-hoc way some new functionalities not supported by the latter. The scheduler supports a plethora of features for representing activities, such as:

- Variable duration, where allocating more duration in an activity, results in more utility for the person (e.g., scheduling 2 hours for homework is more useful and scheduling 1 hour only)
- Arbitrary temporal domains and utility functions over them
- Alternative locations for each activity and information about traveling time
- Interruptible and non-interruptible activities, with additional constraints and preferences supported over the way the different parts of an interruptible activity are scheduled in relation to each other
- Periodic activities, with each occurrence of them being scheduled separately within its period
- Ordering, proximity and implication binary constraints and preferences over pairs of activities
- Temporal overlapping between low-intense activities

The scheduler solves a constraint optimization problem in two phases: in the first phase, it creates an initial seed plan, by using an adaptation of the well-known Squeaky Wheel Optimization[12] framework. In the second phase, it tries to improve this plan, by employing stochastic local search[13, 14]. Furthermore, the scheduler is able to produce alternative plans of high utility and with significant deviation between each other[15]. According to experimental results, the SELFPLANNER scheduler is very effective and efficient in optimizing and solving scheduling problems with the aforementioned types of constraints and preferences.

The problem of using an intelligent system to schedule individual and joint activities has been studied extensively. Most of the earlier work focuses on meeting scheduling[16, 17, 18]. The work by Smith and Lieberman focuses on goal recognition for event management[19], whereas Yorke-Smith et al.[20] focus on proactive personal agent for task management. PTIME[21, 22] aims at facilitating meeting scheduling, by learning the user's preferences, whereas it emphasizes in adopting natural language for interfacing with the user. A more recent work focusing on intelligent assistance on to-do lists has been presented by Gil et al. [23]

3. COURSR2

This section presents in detail COURSR2[c]. It comprises an overview of its functionalities, the overall architecture, scheduling issues and the underlying ontology to represent the temporal aspects of learning objects.

3.1. *Overview*

COURSR2 can be used in two modes: The instructor mode and the learner mode. As instructor we refer to the owner or the manager of the learning object, who is responsible for providing the description of it. In the instructor mode the user can add new course descriptions in the system's database, in terms of their temporal aspects, whereas in the learner mode the system is used to schedule new learning objects within the user's calendar. All users should create an account in order to be able to use the system. Upon registration, the user will be prompted to decide what his account type is.

3.1.1. *Instructor Mode*

COURSR2 supports local accounts, as well as login using various OAuth service providers (Google, Facebook, LinkedIn, and StackExchange). The main screen of COURSR2 shows the dashboard containing recent actions, like new courses created (for instructors) or the new courses enrollment (for learners) (Fig. 1).

Each learning object is characterized by its title, the name of the instructor (if any), the semester when the learning object is offered (if any), its start and end dates, as well as an optional description. No course activity, including exams, is allowed to take place outside the temporal window defined from the "starts on" and "ends on" fields (Fig. 2).

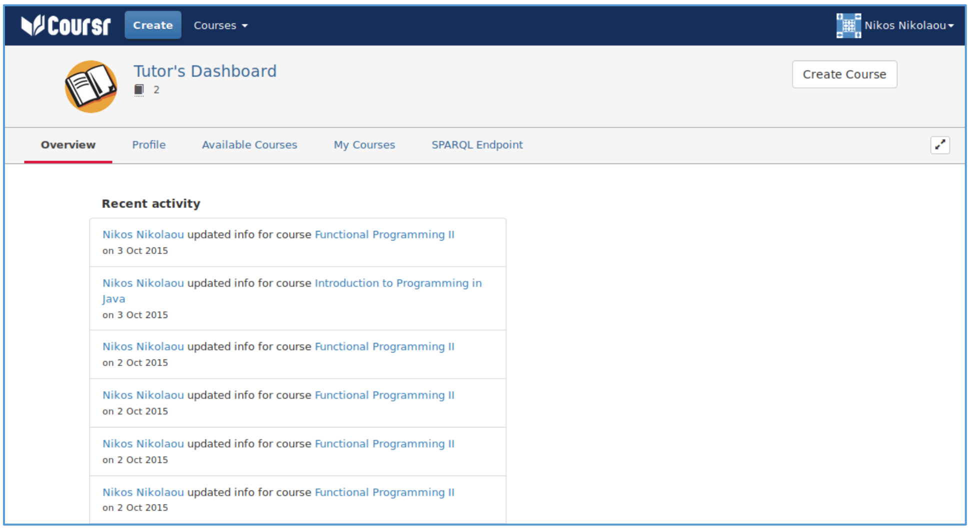

Fig. 1. The dashboard

[c] COURSR2 is available at http://coursr.uom.gr

A learning object comprises several activities. Six semantically different types of activities are supported: Lecture, Lab, Midterm exam, Final exam, Presentation, and Homework. More than one activities of the same type for a single learning object can be defined. For each activity, irrelevant to its type, the following information can be given:

- *Workload*: The time required by the learner to get prepared in advance, in order to be ready for the specific activity. For example, for a weekly lecture, the learner may need 60 mins for preparation, in order to take the most from attending the lecture, whereas for the final exams he may need 12 hours to prepare in order to succeed. This workload concerns asynchronous time, that is, a learner could allocate the workload at the time of his convenience, taking into account his other commitments. An activity may have zero workload; in this case, it should have a positive duration (next field).

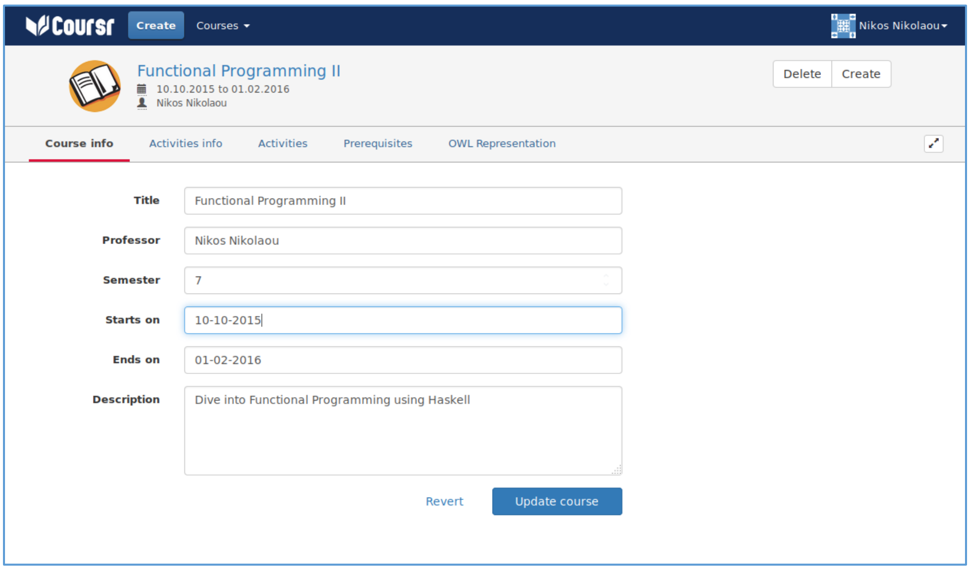

Fig. 2. Basic information of a course

- *Duration*: The actual duration of the activity. For example, a lecture may last 90 mins, whereas the final exams 180 mins. This is synchronous time, that is, a learner has no flexibility as of when to schedule this time. An activity may have zero duration; in this case, it should have a positive workload (previous field).
- *Periodicity*: An activity may be periodic, e.g., a series of lectures. COURSR2 accepts the definition of an educational activity's periodicity through natural language. The text is analyzed by a dedicated parser that has been developed within the COURSR2 project. Examples of valid periodicity text descriptions are the following:
  - One time periodicity
    - On 20/3/2016 at 15:30
  - Sometimes periodicity
    - On 20/3/2016 at 15:30 and on 21/3/2016 at 17:00

- Weekly periodicity
  - Every Monday at 15:30 and every Friday at 16:00
  - Every second Monday at 15:30 from 13/10/2015 to 04/01/2016
  - Every second Monday at 15:30 excluding 20/11/2015
  - Every third Monday at 15:30 including 20/11/2015 at 13:30

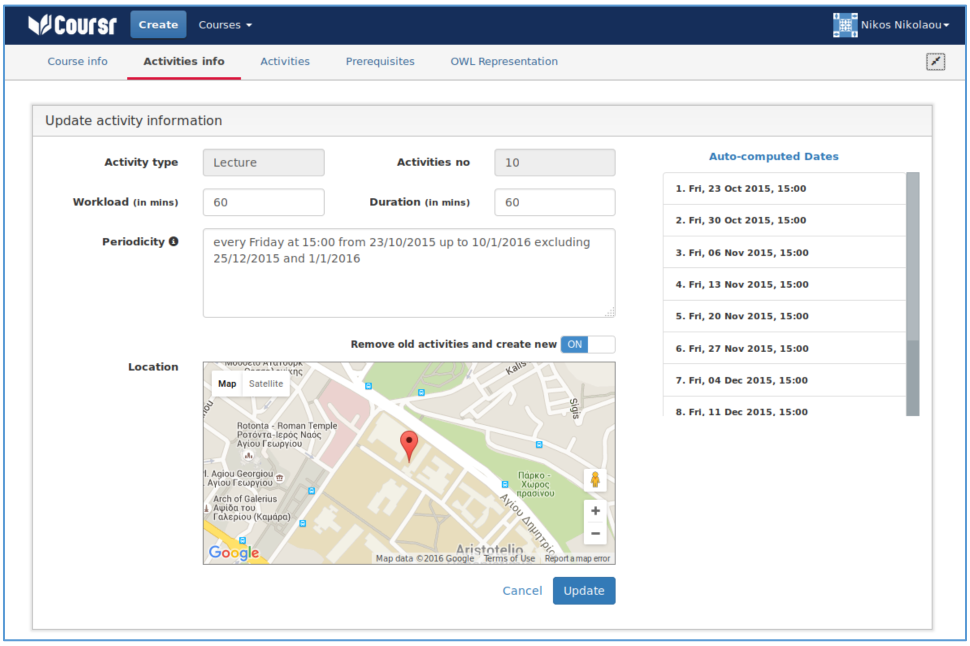

Fig. 3. Temporal aspects and location of an activity that is part of a learning object

For each valid periodicity description, COURSR2 displays the dates of the individual occurrences of the event (Fig. 3, area “Auto-completed dates”). Different occurrences of a periodic activity may be treated as different activities, that is, they might have different descriptions, or they can be treated in the same way, by having a single description. Finally, the exact location where the synchronous part of the activity takes place can be specified, in case of activities that require the physical presence of the learner. Traveling distance between locations of temporally adjacent activities is taken into account by the scheduling engine.

Defining periodicity in natural language is, to the best of our knowledge, another innovation of COURSR2; other systems nowadays use either a complex combination of form controls (e.g., various calendar applications like Google Calendar, Yahoo! Calendar or Microsoft Outlook), or a combination of patterns (e.g., Alexiadis and Refanidis[25]). Each activity’s occurrence may have a separate title, a description and a comment (Fig. 4).

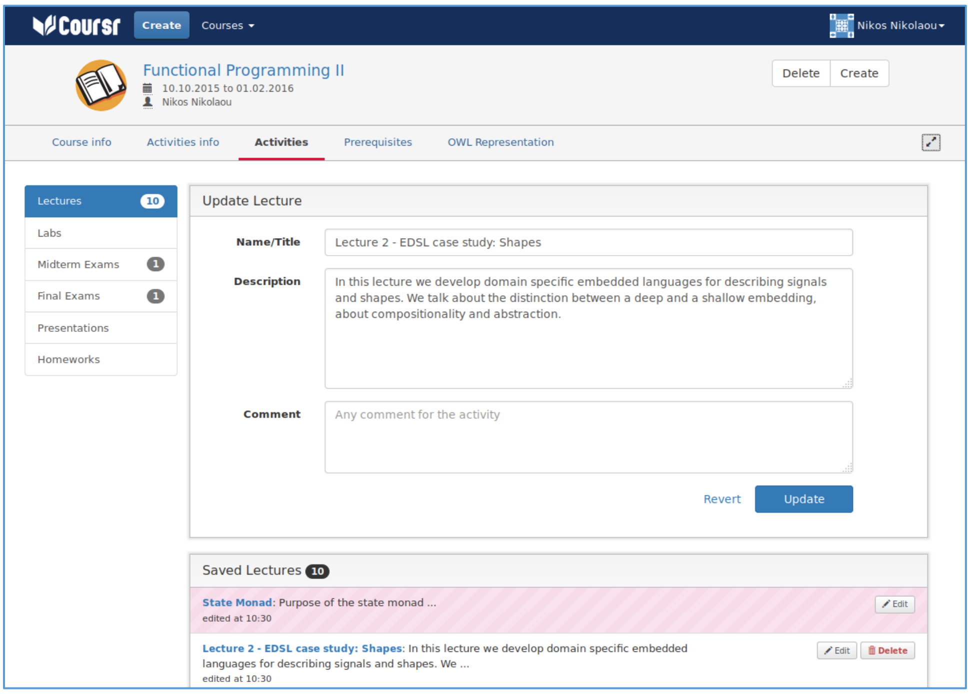

Fig. 4. Editing individual activities

Three types of constraints between a learning object's activities can be defined:

- *Course prerequisites*: These concern other learning objects that the learner should successfully conquer, in order to be capable of registering with the particular learning object.
- *Attendance prerequisites*: These are cardinal constraints about the activities of the learning object, which should be fulfilled by the learner, in order to successfully conquer it. Attendance prerequisites are given in natural language, as in the following examples:
  - At least 1 of "Lecture[1]", "Lecture[2]", "Lab[3]"
  - At most 1 of "Lecture[1]", "Homework[2]" and "Lecture[3]"
  - At least 50% of all lectures
  - At most 50% of all labs
- *Activity prerequisites*: These concern binary constraints over activities of the learning object that should be fulfilled by the learner, in order to be considered eligible to proceed with another activity of the same learning object. For example:
  - "Lecture[1]" before "Lecture[2]", "Lab[1]"
  - At least 50% of all lectures before "Final exam[1]"
  - At most 50% of all labs after "Presentation[1]"

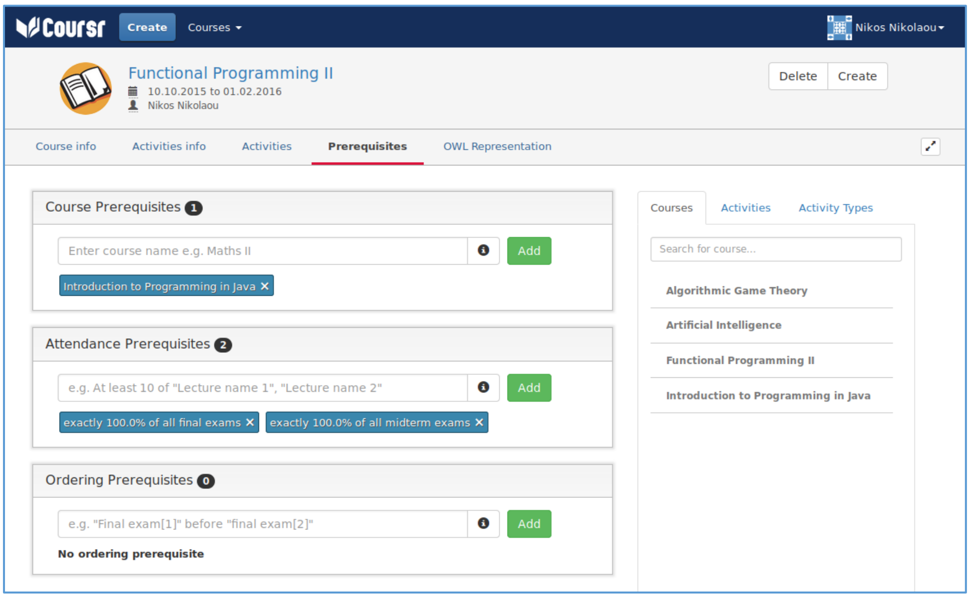

Fig. 5. Prerequisites to successfully conquer a learning object

The learning object description, with all of its activities and prerequisites, can be exported in some ontological format (Functional, Turtle, RDF/XML and simple XML are supported). Currently an RDBMS (PostgreSQL) is used to store the learning objects, mainly for performance and efficiency reasons. However, the application gives the option to additionally store all the information of each learning object to an on-disk triple-store in RDF format, using the Sesame API[26]. By doing that, this information becomes available to be queried on later, not only by the users of the system, but also by external users or other systems, through the SPARQL endpoint provided by COURSR2.

3.1.2. *Learner Mode*

The learner's mode involves functionalities concerning registering with learning objects and scheduling them within the learner's calendars. In general, the functionalities offered by COURSR2 to the learner are the following:

- Browsing and subscribing to learning objects.
- Granting and revoking access to COURSR2 to some of the learner's Google calendars. Any event in the identified calendars is considered busy time, so no COURSR2 activities can be scheduled there (except for events that have been previously created by COURSR2 or SELFPLANNER; these can be rescheduled). After scheduling, all COURSR2 activities are uploaded to an indicated calendar.
- Scheduling of learning object activities. Scheduling takes into account the location reference of COURSR2's activities.
- Specifying whether already scheduled learning object activities should be rescheduled or remain frozen.

- Connecting a COURSR2 account with a SELFPLANNER account, if any. If the learner has non-educational activities in his SELFPLANNER account, he can select to reschedule them as well, while trying to accommodate the learning object activities within his daily program. Thus, in the extreme case, in order to schedule one or more new learning objects within his calendar, the learner may ask to reschedule all already scheduled learning objects, as well as all already scheduled non-educational activities (provided that they have been initially scheduled by SELFPLANNER).

COURSR2 cannot reschedule events directly inserted into the learner's Google calendars, since no temporal domain and spatial information is provided for them. So, these are considered just as busy time. Furthermore, since these events usually do not have spatial information, it is not computed traveling time between these events and COURSR2 or SELFPLANNER activities. Thus, for users of both COURSR2 and SELFPLANNER, it is advisable not to insert events directly into their Google Calendars, in order to take advantage of the full functionality of these systems.

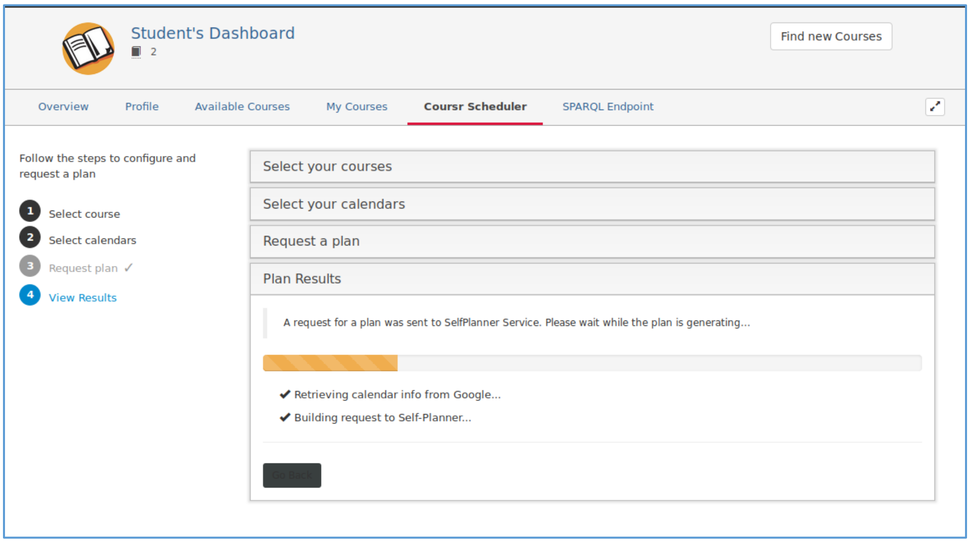

Fig. 6. The learner's user interface

Fig. 6 gives an overview of the learner's user interface. There are several sections in it. The Overview section keeps track of the last user actions within the system. In the Profile section the learner can edit his personal data, grant or revoke access to his Google calendars and connect his COURSR2 account with his SELFPLANNER account. The Available Courses section browses all the courses of the system, whereas the My Courses section lists only the courses with which the learner has subscribed. Finally, the COURSR2 Scheduler section (being active in Fig. 6) employs the scheduler to accommodate new learning objects within the learner's calendars.

3.2. *Architecture*

COURSR2 integrates with SELFPLANNER, in order to provide a holistic experience to the learner, provided that he also possesses a SELFPLANNER account. Furthermore, it requires access to the learner's Google calendars, in order to read, as well as to update information. The instructor mode of use does not present any interest from an architectural point of view, operating just as an information system. On the contrary, the learner mode of use is more complicated. The overall architecture of COURSR2, which reflects the learner's mode of use, is shown in Fig. 7.

A COURSR2 learner's session starts with the learner selecting one or more new learning objects and subscribing with them. The learner might previously have inserted static events directly into his Google calendar. Furthermore, he might have already inserted non-educational activities into his SELFPLANNER account. These activities have been already scheduled and appear into his Google calendars, with their detailed information being retained only in SELFPLANNER.

After the learner subscribes to one or more new learning objects, he requests a new plan, in order to check whether he has the necessary time resources to cope with them. COURSR2 requests from Google, as well as from SELFPLANNER, the details of all user events. With this information, COURSR2 distinguishes all acquired events in three categories: Static events, educational events and non-educational events, with the last two categories supporting the full SELFPLANNER domain model (that is, temporal domains, duration range, locations, etc). The time slots occupied by static events are removed from the temporal domains of the remaining activities and, then, a local scheduler (which is a copy of SELFPLANNER's scheduling engine) is employed to find a plan.

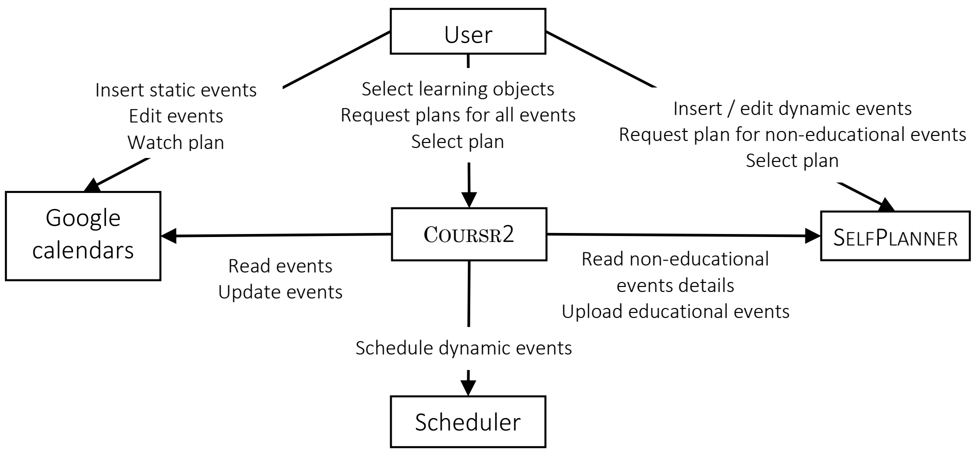

Fig. 7. COURSR2 overall architecture

If the plan is accepted by the user, then his Google calendars are updated with the new schedule of COURSR2 and SELFPLANNER activities, whereas SELFPLANNER is also updated with the new educational activities. Otherwise, no update takes place and the user can subscribe to other learning objects. If no plan satisfying the constraints is found, a message informs the user about the impossibility to accommodate the new learning objects within his existing commitments.

A user may have multiple Google calendars. COURSR2 allows the user to select which of them have non-educational activities, and which one calendar to update with his educational activities after successful scheduling.

The above architecture is extensible. Using SELFPLANNER as the central repository of dynamic activities and Google Calendar to keep the current schedule of all activities, other peripheral systems, like COURSR2, could be added to the architecture in a similar way. If the user logins through a peripheral system, he can select activities of a particular type (e.g., educational activities), and schedule them together with SELFPLANNER's activities. The details of new or modified domain dependent activities should be uploaded to SELFPLANNER, whereas the selected schedule should be uploaded to Google Calendar. Events uploaded to Google Calendar should be marked as being SELFPLANNER events.

The alternative approach would be to adopt a centralized architecture, where users use a single application for all types of activities, general and domain specific. Our developing experience suggests that such an approach would be impractical from a software engineering point of view, due to the numerous special requirements each domain specific application has.

The architecture in Fig. 7 does not present the internal details of the SELFPLANNER system. For example, if the learner logins directly to SELFPLANNER, the system communicates with Google calendar as well and solves the problem by employing (a different instance of) the scheduler.

3.3. *Scheduling Issues*

As already mentioned, COURSR2 uses a local instance of SELFPLANNER's scheduling engine. The domain model supported by the engine has been presented in Refanidis and Yorke-Smith[6], whereas the algorithms have been presented in the same publication and in Alexiadis and Refanidis[9]. A brief description has been provided in the related work section of this article.

In order to use the scheduling engine, a mapping from COURSR2 activities and constraints to the features supported by scheduling engine is required. For example, *Activity Prerequisites* can be encoded using the implication constrains supported by the domain model; should the learner have taken the Mid-term exam in order to be eligible to participate in the Final exam, an implication constraint of the form "Final-exam ⇒ Mid-term exam" has to be defined. This constraint conveys the information that in order to produce a valid plan, it must add the Mid-term exam activity, in case Final-exam activity is also present in the final plan.

There are activities that should be scheduled before, after or between other activities. For example, "Studying for the Final exam" should always precede the "Final exam" activity, while "Studying for Lecture #3" should be scheduled some time between the end of

"Lecture #2" and the start of "Lecture #3". Ordering constraints supported by the scheduling engine are used to encode these requirements. Furthermore, proximity constraints are used to limit the maximum distance between pairs of activities. As an example, preparing for a lecture should be scheduled at most one week before the lecture. On the other hand, cardinality constraints are defined in COURSR2, but they are not supported by the scheduling engine. Examples of cardinality constraints are scheduling necessarily the final exams and the preparation for them, as well as a minimum number of lectures, in order for the learner to successfully complete the learning object. The domain model of the scheduling engine does not support cardinality constraints, however it supports utilities over the activities i.e., activities with higher utilities are given priority during scheduling.

COURSR2 manages to schedule the minimum number of required activities of a learning object by assigning to them increased utility compared to the rest of the activities of the learning objects. However, this does not guarantee that they will be scheduled. So, after getting a plan from the scheduling engine, COURSR2 validates it in terms of the cardinality constraints. In case a cardinality constraint is violated, COURSR2 uses a heuristic method in order to re-assign utility values to all activities contained in the requested plan. More specifically, the utilities of the required activities that were not able to be scheduled are increased, while the utilities of the other activities are decreased.

For example, suppose there are two learning objects to be scheduled, Course#1 and Course#2, and two constraints: the student should attend 7 out of 11 lectures of Course#1, as well as 9 out of 10 lab sessions of Course#2. In the returned plan, the scheduling engine manages to schedule 10 out of the 11 lectures of the Course#1, but only 8 of the 10 lab sessions of Course#2, leading to a non-acceptable solution. So, COURSR2 decreases the utility of the activities of Course#2 in favor of the activities of Course#1, and calls the scheduling engine again. This process is repeated a (user defined) number of times; if, after all these trials, no valid plan is found, the learner is informed that the new learning objects cannot be accommodated within his daily plan.

Currently, the following formula is used to re-adjust the activities' utility:

$$u_n = \left(1 + \frac{(a_r - a_s)}{2 * a_t}\right) * u_c$$

where:

u_c, u_n: Current/new utility value

a_r, a_s, a_t: Number of required/scheduled/total activities

In the aforementioned example, the utility of each lecture of Course#1 will be decreased by ~13% while the utility of each lab session of Course#2 will be increased by ~5%.

3.4. *The Ontology*

The COURSR2 ontology (Fig. 8) was developed with the goal to facilitate data exchange between COURSR2 and other systems or the user. It reflects the internal data structures of the system. Some of the more prominent classes of the ontology are the following:

Fig. 8. A graphical representation of the COURSR2 upper ontology

- *Course*: This class can be considered as the "root" of the whole ontology, as it uses explicitly or implicitly almost all the other classes. It encodes all the information a learning object consists of, that is, activities, start/end dates, etc.
- *SchoolActivity*: This class represents a generic activity. COURSR2 supports 6 types of activities, each one being a subclass of the main class SchoolActivity.
- *SchoolActivitySynopsis*: Contrary to SchoolActivity class, this class is used to represent information not about a single activity, but meta-data info relative to the whole set of all activities of the same type. For example, a LectureActivitySynopsis

(subclass of SchoolActivitySynopsis) contains information like the number of the lectures of a specific course, the periodicity of all lectures, or the duration of them. As with SchoolActivity, for each activity type there is a corresponding ActivitySynopsis subclass.

- *Periodicity*: This class encodes the various periodicity types. Currently, the ontology supports 4 types of periodicity: One time, sometimes (that is, a list of arbitrary non-periodic dates), weekly and monthly periodicity.
- *Prerequisite*: This class contains 3 subclasses, each one representing a different type of prerequisite (course, activity and attendance) that are supported by COURSR2.
- *User*: This class is used to represent information about the two types of users: Student and Tutor. In the future, we intend to integrate this class with elements of other established ontologies like FOAF[27].

COURSR2 provides a SPARQL endpoint, in order to give the users the option to access its database in an ontological way, using SPARQL queries (the full SPARQL language is supported). The user can see all triples stored on the RDF database that match the respective SPARQL query, whereas he can also export all the results of his queries in CSV format.

4. Empirical Evaluation

The goal of the empirical evaluation is to get feedback from real users, including learning objects managers, as well as learners, about the usefulness and the usability of the overall idea, as well as particularly COURSR2. We have focused in university classes, since it was easy to find both academics and learners to participate in the evaluation. However, we believe that the results are not bounded to this type of learning objects, but they can be generalized to any type of learning objects.

For the purpose of the evaluation, we have created two simple scenarios, demonstrating COURSR2, as well as its integration with SELFPLANNER. The first scenario comprises a university class, namely “Programming I”, comprising several lecturers and exams, plus a non-educational activity, namely “Car service”, which is already scheduled by SELFPLANNER on Monday morning of some week. The user prefers that “Car service” is scheduled in the morning of any week day. After subscribing to “Programming I”, the learner asks COURSR2 for a plan. However, a lecture of the class coincides with the already scheduled “Car service”. COURSR2 reschedules “Car service” for Friday morning, in order to put a lecture on Monday morning of the particular week.

The second scenario is similar to the first, with the learner wanting to subscribe to class “Artificial Intelligence I”. In that case, a non-educational activity, “Interview”, already scheduled on Sunday morning of some week, cannot be rescheduled. Furthermore, “Interview” coincides with “Midterm exam”, with the “Interview” being considered more important. Even worse, “Midterm exam” is a required activity for the class, that is, without passing the midterm exam, the student cannot proceed to the final exam and pass the class. So, COURSR2 cannot find a plan for the class “Artificial Intelligence I” and informs the learner that he has not the required time resources to attend it.

We have created two short videos[d] that demonstrate the two scenarios, including some useful information about the use of COURSR2 and SELFPLANNER. We have also created a questionnaire with two paths, the first one concerning instructors and the second one concerning learners. There was also a general question, concerning whether the user (instructor or learner) uses electronic calendars to organize his personal life, or not. All questions are of the multiple choice type, ranging from “absolutely NO” to “absolutely YES”. Finally, the participants can leave text comments[e].

We designed the questionnaire with the goal to be simple enough for the participants, while simultaneously to give us the necessary information. Concerning the selection of the questions for the instructors, the particular goal was to investigate whether they are able and would like to describe the temporal requirements of their classes at a fine grained level of detail. Concerning the selection of the questions for the learners, the goal was to investigate whether they are interested in using an intelligent system to organize their time, as well as to be supported in taking decisions about whether they can commit to a learning object or not. Of course, in both cases we were interested in evaluating the usability and intuitiveness of the particular system.

We have asked several instructors and students to watch the video, experiment (optionally) with COURSR2 trying to setup their own scenarios and, finally, fill in the questionnaire. We made clear to them that this is a prototype system, so we do not want from them to evaluate only the specific system, but also and foremost the overall idea behind it. An invitation to participate in the empirical evaluation was sent primarily to current and former master students of several open universities over the period of the last 10 year, but also to postgraduate and selected undergraduate students of University of Macedonia. The invitation was also sent to instructors, some of them having also experience with teaching lifelong learners. Participation in the evaluation was anonymous.

We have received 11 questionnaires by instructors, with 9 of them describing themselves as electronic calendar user, and 25 questionnaires from students, with 18 of them describing themselves as electronic calendar users. In the following paragraphs we present the results of the evaluation using bar charts, starting with the instructors.

All question results have been compared against two discrete groups of people: instructors are classified based on whether they use e-calendars or not, while learners are classified as busy or not busy people. In order to examine the significance of the independence of the replies of the two groups we used the chi-square test, a well-known test suitable for small sample sizes. We used as null hypothesis the “H0: the reply is independent of the group type” and “H1: the reply is associated with the group type”. Statistical significance coefficient was set to a = 0.05, while the degree of freedom is 1.

[d] Videos can be found at https://goo.gl/wfoWiI (duration 11min 53’) and https://goo.gl/VeNe2z (duration 6min 9’)

[e] The questionnaire can be found at https://goo.gl/Ul1ocz

The p-values for H0 are given (p_A), where p-values greater than 0.05 confirm the null hypothesis (H0).

Additionally, in order to investigate the statistical significance of the replies of the users, we applied a second test with the null hypothesis "H0: the replies of the users are random following the normal-distribution (with mean=3, s = 1 and a = 0.05)", and "H1: the replies are honest and conscious". The p-values for H0 are also given (p_B).

QI1: Do you think you could describe the time requirements of a class at this fine-grained level of detail?

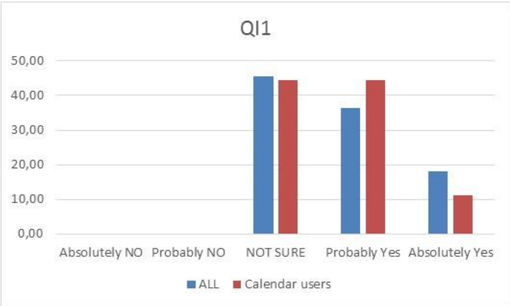

Fig. 9. Detailed results for question QI1, for all and calendar users separately

According to Figure 9, 56% of calendar users believe that they could describe the time requirements of learning objects at a fine-grained level of detail, whereas this ratio falls slightly to 55% for non-electronic calendar users ($p_a = 0.96, p_b = 0.32$).

QI2: Would you devote some of your time to describe the time requirements of a class at this fine-grained level of detail?

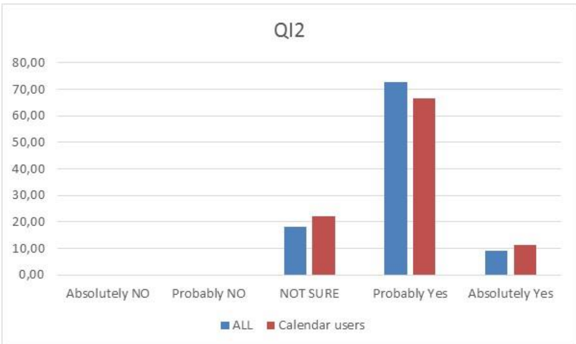

Fig. 10. Detailed results for question QI2, for all and calendar users separately

According to Fig. 10, 78% of calendar users are positive towards devoting time to describe the time requirements of their classes at a fine-grained level of detail, whereas this ratio remains practically the same (82%) for non-electronic calendar users ($p_a = 0.82, p_b = 0.04$).

Comparing the results from questions QI1 and QI2, we notice a significance increase in the willingness of the teachers to feed the system with the analytical temporal details of the educational objects they manage, compared to their estimate about whether they will be able to do so. This result is rather encouraging, since it demonstrates a positive attitude towards such technologies, which can be reinforced by more usable and intuitive systems.

QI3: Did you like the natural language interface to describe temporal aspects and other requirements for a class?

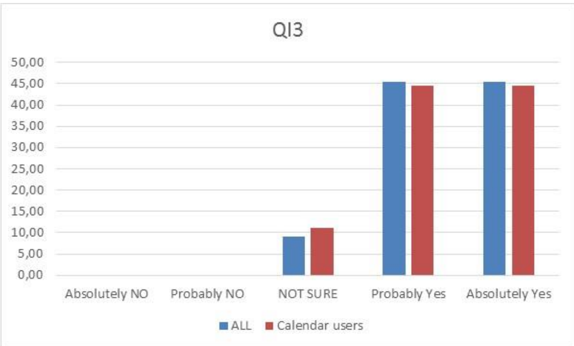

Fig. 11. Detailed results for question QI3, for all and calendar users separately

According to Fig. 11, 89% of calendar users and 91% of non-calendar users evaluated positively the natural language interface to describe periodicity rules and attendance prerequisites ($p_a = 0.88, p_b = 0.01$).

In the general comments that we received from instructor evaluators, among others we received suggestions about supporting multiple languages, including Greek; others suggest to provide the schedule of the day (however, this can be retrieved from the user's calendar). However, most of the comments were encouraging to continue to this line of research.

The first question for student evaluators (QS1) was whether they have enough free time during a semester. This is an important question, since students with a lot of free time do not need electronic calendars and tools like COURSR2 to organize their life. 15 students out of the 25 answered that they do not have enough free time. Furthermore, 11 students were answered that they both use electronic calendars and do not have enough free time. In the following bar charts we give statistics for all students (25 evaluations), as well for their subset that uses electronic calendars and has not enough free time (11 evaluations, referred as busy students).

QS2: Would you use an application like COURSR2 to organize your educational activities?

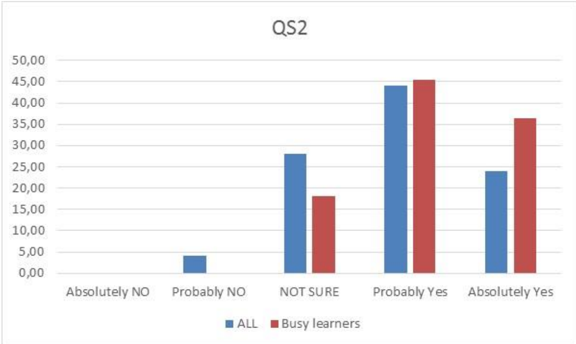

Fig. 12. Detailed results for question QS2, for all students and for busy only

According to Fig. 12, 68% of all students and 82% of busy students are positive towards using a system like COURSR2 to plan their educational activities ($p_a = 0.39, p_b < 0.01$).

QS3: Would you use the combination COURSR2 & SELFPLANNER to organize all your activities?

According to Fig. 13, 60% of all students and 64% of busy students are positive towards using a system like COURSR2 plus SELFPLANNER to plan all of their activities. Comparing the answers to QS2 and QS3, we can see that the percentage of students that are thinking positively about using intelligent applications to plan all of their activities has decreased considerably to the students who are positive in using intelligent systems to plan only their educational activities. This fall can be attributed to several factors, such as the negative feeling arising from having an intelligent system planning our life; the burden of continuously updating an application about our future plans; the fear of losing privacy; usability issues concerning the specific two systems, and several other factors ($p_a = 0.83, p_b < 0.02$).

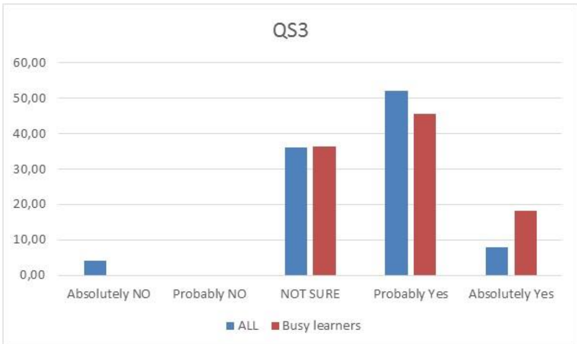

Fig. 13. Detailed results for question QS3, for all students and for busy only

QS4: Did you find the COURSR2 interface usable?

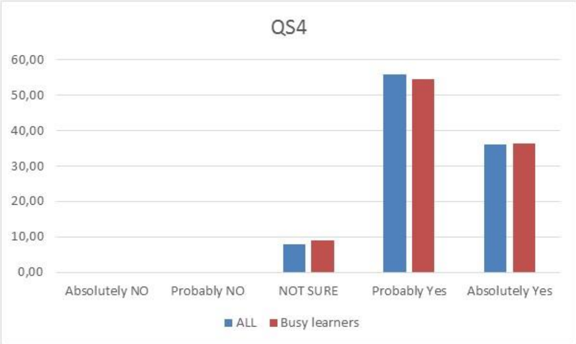

Fig. 14. Detailed results for question QS4, for all students and for busy only

According to Fig. 14, 92% of all students and 91% of busy students consider the user interface of COURSR2 usable ($p_a = 0.91, p_b < 0.01$).

QS5: Did you find the COURSR2 interface intuitive?

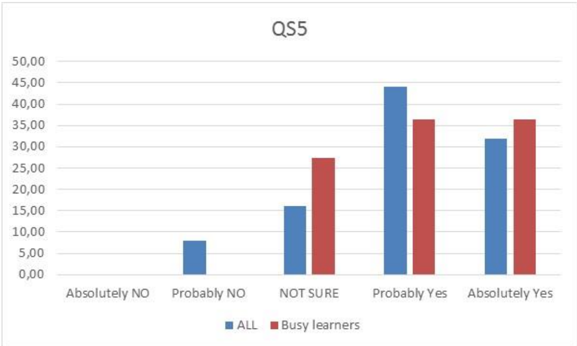

Fig. 15. Detailed results for question QS5, for all students and for busy only

According to Fig. 15, 76% of all students and 73% of the busy students consider the user interface of COURSR2 intuitive ($p_a = 0.83, p_b < 0.01$).

We received plenty of free text comments from the students. Besides the many encouraging messages, we received some interesting suggestions that concern several aspects of the systems, such as the following:

- More user friendly interface
- Student profiles that relate to the time a user needs for homework and exam preparation

- Notification of last minute changes
- Mobile application for smartphones

Taking into consideration that p_a-values of all question are larger than 0.05, we can safely conclude that the replies of the users are irrelevant of their group (instructors using or not using an e-calendar, and busy or not busy students). On the other side, most p_b-values are quite smaller than 0.05, indicating honest replies on behalf of the users, thus increasing the significance of the survey.

The results of the evaluation are very encouraging, both in terms of the overall attitude of users towards using intelligent systems to organize their student life, as well as particularly for COURSR2. Nevertheless, this is a first evaluation, with a small sample of users. More comprehensive results could be extracted after deploying the system and using it in real world, with real users and data.

5. Conclusions and Future Work

COURSR2 is a prototype innovative application aiming at helping learners to schedule their educational activities within their professional and personal activities. Its main target group comprises people that study or retrain at a higher age, thus their free time is limited. COURSR2 tries to solve the problem of taking informative decisions about whether the learner is able to successfully bring in fruition a new educational commitment or not. To manage this, COURSR2 possesses detailed information of learning objects, in terms of everyday duties (lectures, homework, exams, etc), considering their synchronous and asynchronous part, the need for physical presence, whether they are obligatory or not, etc.

In addition, COURSR2 cooperates with another system, called SELFPLANNER, which is a general scheduling system for everyday activities, so as to enable its users to schedule dynamically all their activities. The empirical evaluation of COURSR2 with real users, teachers and students, has shown that there is a positive attitude from the teachers to encode the temporal requirements of their classes in COURSR2, and from students to use intelligent systems to organize their educational activities, but not all of their activities.

For the future, we plan to extend COURSR2 in terms of its exposed functionality (e.g., a more detailed model to describe learning objects, a better interface, etc). Another goal is the enrichment of the system's database with more learning object descriptions. Ideally, this could be done with the adoption of COURSR2 by a large institution, e.g., a university, in which case we could also proceed in a large scale evaluation. Furthermore, the adoption of COURSR2 by a trusted institution would encourage users to grant access to their calendars, in order to exploit its full functionality.

In order to facilitate data gathering, we plan to develop a widget, which will allow viewing and editing the learning object information directly in the target web page of the learning object (e.g., the class web page). Finally, other potential extensions include integrating the underlying ontology with other known ontologies, such as the IEEE 1484.12.1 – 2002 ontology, as well as the whole system within a LMS, e.g., Moodle.

Acknowledgments

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II: Investing in knowledge society through the European Social Fund.

Co-financed by Greece and the European Union

References

1. K.Agnantis and I.Refanidis, 2015. COURSR: Scheduling Composite Educational Objects. *In Proceedings of the Scheduling and Planning Applications Workshop (SPARK-2015),* within the ICAPS-2015, Jerusalem, pp. 50-58.
2. W. Watson and S.L. Watson, 2007. An Argument for Clarity: What are Learning Management Systems, What are They Not and What Should They Become? *TechTrends*, Springer, vol. 51, no. 2, pp. 28-34.
3. W. Debbi, S. Christine and S.N. Chenicheri, 2008. Academic and Student Use of a Learning Management System: Implications for Quality. *Australian Journal of Educational Technology*, vol. 24, no.1.
4. L. Castillo, L. Morales, A. González-Ferrer, J. Fdez-Olivares, D. Borrajo, and E. Onaindía, 2010. Automatic generation of temporal planning domains for e-learning problems. *Journal of Scheduling*, *13*(4), 347-362.
5. A. Garrido, L. Morales, and I. Serina, 2012. Using AI Planning to Enhance E-Learning Processes. In *Proceedings of ICAPS-2012*.
6. A. Garrido, E. Onaindia, and O. Sapena, 2008. Planning and scheduling in an e-learning environment. A constraint-programming-based approach. *Engineering Applications of Artificial Intelligence*, *21*(5), 733-743.
7. C. Limongelli, M. Lombardi, A. Marani, F. Sciarrone, and M. Temperini, 2016. A recommendation module to help teachers build courses through the Moodle Learning Management System, New Review of Hypermedia and Multimedia.
8. C. Limongelli, F. Sciarrone, and G. Vaste, 2011. Personalized e-learning in moodle: The Moodle LS system. Journal of E-Learning and Knowledge Society, 7(1), 49–58.
9. I. Refanidis, 2007. Managing Personal Tasks with Time Constraints and Preferences. *17th International Conference on Automated Planning and Scheduling Systems (ICAPS)*, Providence, Rhode Island.

10. I. Refanidis and A. Alexiadis, 2011. Deployment and Evaluation of SELFPLANNER, an Automated Individual Task Management System. *Computational Intelligence*, 27(1), 41-59.
11. I. Refanidis and N. Yorke-Smith, 2010. A Constraint Based Programming Approach to Scheduling an Individual's Activities. *ACM Transactions on Intelligent Systems and Technologies*, vol. 1 (2), 2010.
12. D.E. Joslin and D.P. Clements, 1999. "Squeaky Wheel" Optimization. *Journal of Artificial Intelligence Research*, vol. 10, 375-397.
13. A. Alexiadis and I. Refanidis, 2013a. Post-Optimizing Individual Activity Plans through Local Search. *In Proceedings of ICAPS 2013 Workshop on Constraint Satisfaction Techniques for Planning and Scheduling Problems (COPLAS)*, pp. 7-15. Rome.
14. A. Alexiadis and I. Refanidis, 2015. Optimizing Individual Activity Personal Plans through Local Search. *AI Communications*, IOS Press, to appear.
15. A. Alexiadis and I. Refanidis, 2013b. Generating Alternative Plans for Scheduling Personal Activities. *In Proceedings of ICAPS 2013 Workshop on Scheduling and Planning Applications (SPARK)*, pp. 35-40. Rome.
16. L. Garrido, and K. Sycara, 1995. Multi-agent meeting scheduling: Preliminary experimental results. *1st International Conference on Multi-Agent Systems* (ICMAS).
17. N.R. Jennings and A.J. Jackson, 1995. Agent based meeting scheduling: A design and implementation. *IEE Electronic Letters*, 31(5):350-352.
18. S. Sen and E.H. Durfee, 1998. A formal study of distributed meeting scheduling. *Group Decision and Negotiation*, vol.7, pp. 265-289.
19. D. A. Smith, and H. Lieberman, 2009. Recognizing and using goals in event management. *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. ACM.
20. N. Yorke-Smith, S. Saadati, K. L. Myers, and D. N. Morley, 2012. The design of a proactive personal agent for task management. *International Journal on Artificial Intelligence Tools*, *21*(01), 1250004.
21. P. Berry, K. Conley, M. Gervasio, B. Peintner, T. Uribe and N. Yorke-Smith, 2006. Deploying a Personalized Time Management Agent. In Proceedings of *5th International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS), Industrial Track, pp. 1564-1571.
22. P. Berry, M. Gervasio, B. Peintner, and N. Yorke-Smith, 2011. PTIME: Personalized assistance for calendaring. *ACM Transactions on Intelligent Systems and Technology (TIST)*, *2*(4), 40.
23. Y. Gil, V. Ratnakar, T. Chklovski, P. Groth, and D. Vrandecic, 2012. Capturing Common Knowledge about Tasks: Intelligent Assistance for To-Do Lists. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, *2*(3), 15.
24. A. Alexiadis and I. Refanidis, 2009. Defining a Task's Temporal Domain for Intelligent Calendar Applications. *In Proceedings of the 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI)*, pp. 399-406, Springer. Thessaloniki, Greece.
25. D. Brickley, L. Miller, 2012. FOAF vocabulary specification 0.98. Namespace document 9. Latest version available at http://xmlns.com/foaf/spec/.
26. J. Broekstra, A. Kampman and F.V. Harmelen, 2002. Sesame: A generic architecture for storing and querying rdf and rdf schema. *The Semantic Web—ISWC*. Springer Berlin Heidelberg, pp. 54-68.