Assembling Learning Objects for Personalized Learning: An AI Planning Perspective

Antonio Garrido and Eva Onaindia, *Universitat Politècnica de València*

Some of the challenges of e-learning include selecting the proper learning objects, defining their relationships, and adapting their sequencing to the student's specific needs, objectives, and background.

The Internet offers many opportunities for promoting student learning: you can easily find terabytes of learning object (LO) repositories for course assembling, customization, and content packaging.[1–3] But despite this amount of data, institutions engaged in educational processes have traditionally opted for developing their own nonstandard solutions. The resulting incompatibility of LO formats, along with the tendency of regarding them in isolation, has contributed to a general lack of reusability and interoperability.

For effective interoperability, a LO must be a stand-alone, modular entity that incorporates its learning context, or semantic relationships, in its own metadata. Metadata labeling is a key issue for LOs' semantic annotation, encoding, exchange, and reuse, as it offers a successful way to catalogue and navigate content, context, usage, and structure.[2–6] This is valid for both adaptive courseware generation, where the goal is to ensure a student completes the required activities, and dynamic courseware generation, where the goal is to assist students in navigating a complex hypermedia space. Metadata labeling is also crucial for dealing with correct task adaptation in terms of educational aspects, such as the difficulty of using the LO and how it affects the learning process.[7–9] Intuitively, the ultimate goal is to create a personalized course, a student-centered solution in which the gathering of activities and their sequencing is tailored to each student's specific needs, objectives, and background.

This personalization perspective has been addressed by using different techniques, such as adjacency matrices, integer programming models, neural networks, and intelligent planning techniques.[9–12] In particular, researchers have successfully applied AI planning techniques to the construction of adapted courses as a means to bring the right content to the right person.[10,12] However, designing a course usually requires dealing with aspects such as group interaction, collaboration, and sharing of specific (and perhaps costly) resources. Thus, it isn't only about bringing the right content to the

right person but also at the right time and with the right resources, a missing aspect in traditional e-learning.

We explore here the potential of metadata enrichment and the promising avenue of planning technology as a step forward in developing reusable, interoperable LOs and pushing forward the agenda for innovative instructional engineering methods and joint tools.

A Motivating Example

Let's assume two students, John and Rebecca, are interested in a Java programming course organized in seven modules, according to the tutorial available at www.merlot.org/merlot/viewMaterial.htm?id=88853 (see Figure 1).

John is in his first year of BS in computing science and is interested in programming. Rebecca is a self-taught programmer with experience in object-oriented programming (OOP) and C++; she wants the Java certificate.

As the top of Figure 1 shows, we have different routes to achieve both learners' goals. If we focus on the LOs of the *Learning the Java Language* module (Figure 1, bottom), the number of routes is even higher. John will require most of the LOs here, whereas Rebecca will need just a few to learn the main features of Java, excluding the OOP-related LOs. This is content adaptation; students receive different sequences of LOs according to their profile, knowledge, and interests. Let's also imagine that the *An example* LO requires the use of a computer. Rebecca has a laptop, but John has no computer, so he needs to go to a lab. Picture now that the *Generics* LO is a lecture that requires in-person attendance on Tuesdays, from 1:00 to 3:00 p.m., and the classroom's maximum capacity is 20 people. This is context adaptation:

considering the real-world constraints (time, resource consumption, and group activity synchronization) to schedule a route or perhaps avoiding some LOs if another route is feasible. Thus, adaptation implies provision for dynamic learning content, an adaptive behavior to promote the quality of learning, and a flexible process that lets students adjust their schedules to the course's resources.

Requirements for Supporting Adaptation

Adaptation involves several technological issues, as depicted in Figure 2: use of common LO repositories and modeling tools, algorithms for students' information acquisition, application of solving techniques, and visualization of learning designs on learning management systems (LMSs). The role of course designers is to model a course by reusing or defining new LOs. The relationship of students with the system is established when setting their profiles and preferences as well as during navigation through their personalized learning designs. One route per student is generated (in an offline mode) before using the LMS, so the student's learning behavior doesn't require any particular change—technological aspects are transparent to students. This personalized learning encompasses five essential requirements.

Labeling Metadata

Metadata labeling is specified by the LO's creators, usually in an XML standard format, such as the IEEE Learning Technology Standards Committee's (LTSC) standard for Learning Object Metadata (LOM).[4] The purpose is to offer a unified way to label LOs for their eventual use or reuse as interoperable units (see Figure 3). There are many useful entries for pedagogical theories, including the

general descriptors, but only three aspects address personalization:

- The platform requirements for the LOs. The technical definition of a requirement is rather vague, such as *Unix operating system*, but in other cases, it's precisely regarded as a resource—that is, an entity of limited availability required by the LO.
- The information about the student's learning style (profile) and its suitability to the LO in terms of educational difficulty and typical learning time.
- Relationships as content dependencies among LOs. These relationships comprise hierarchical structures (*IsPartOf*, used for LOs aggregation) and three types of orderings to represent causal dependencies: *Requires*, *IsBasedOn*, and *References*, as conjunctive, disjunctive, and recommended preconditions, respectively. The first two relations represent hard preconditions, but the third denotes a soft precondition that can involve a kind of incentive or learning reward.

This information is sufficient in most situations, but more details are necessary for real scenarios. For instance, how many resources are available, at what time, and with which cost and capacity? This is important because a limited resource might not be available for all students simultaneously. Also, when an LO is adequate for a profile, does this mean that it's inadequate for other profiles? The same holds true for soft preconditions: How do you measure the incentive value when a recommendation holds?

These challenging questions require some expertise. Consequently, this metadata specification, sound from a pedagogical perspective, needs further extensions.
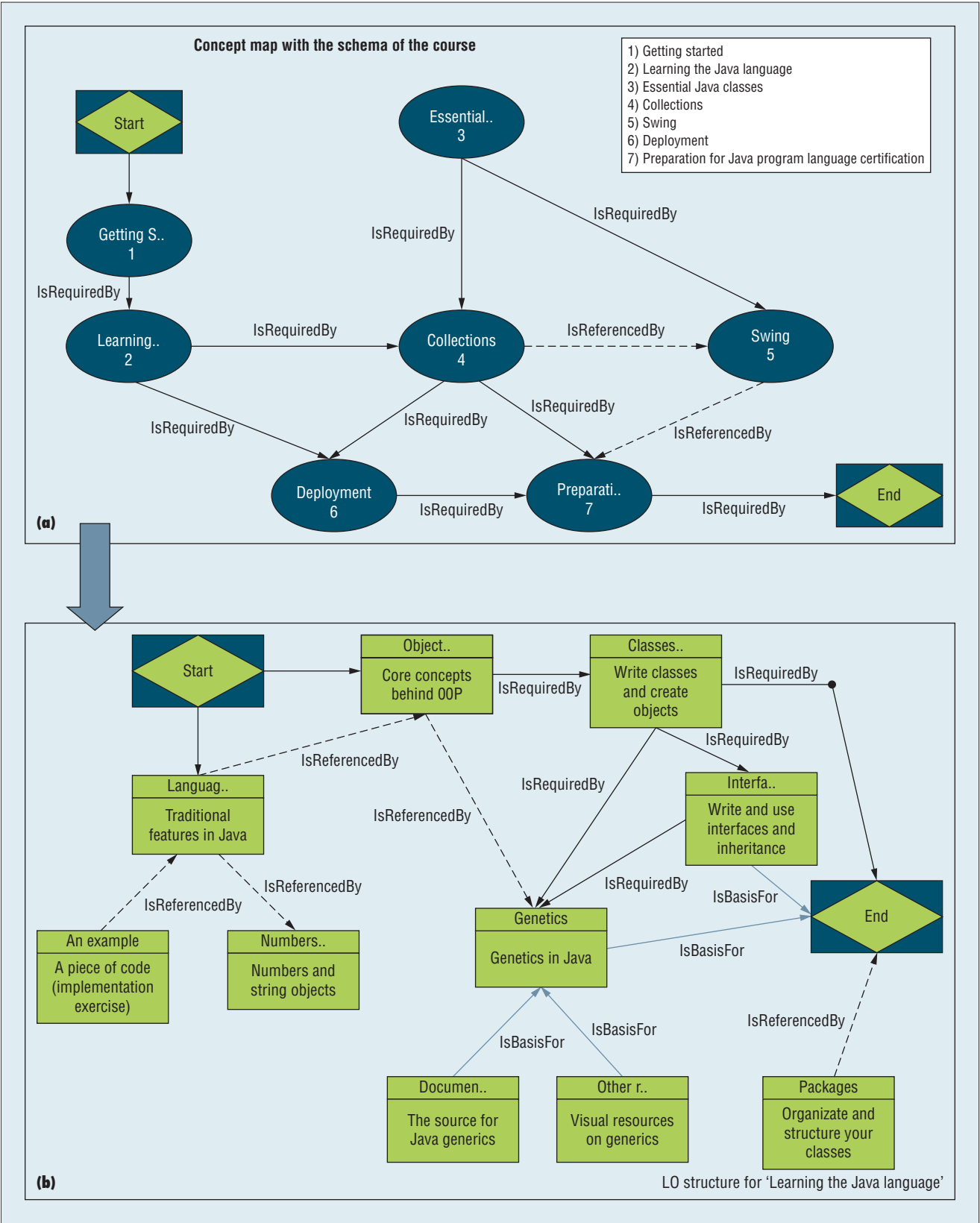
Figure 1. Example case. A basic Java course organized as (a) a seven-module conceptual map and (b) the detailed structure for module 2, *Learning the Java Language*.

www.computer.org/intelligent

Modeling the Course

Creating personalized courses is a hard task because LOs in themselves are insufficient for significant instruction. Hence, offering an incremental and friendly way to link LOs using pedagogical and instructional design theories is highly appreciated.[13] A course initially designed with the same LOs for all students (see Figure 1) can later be enriched by including collections of tailored LOs suitable to particular profiles. For instance, according to Richard Felder and Linda Silverman's learning style classification,[7] a lecture can be very recommendable for verbal students but not for visual ones, and just the opposite holds true for a diagram. This incremental process lets designers extend LO metadata records, thus improving their adaptation capabilities.
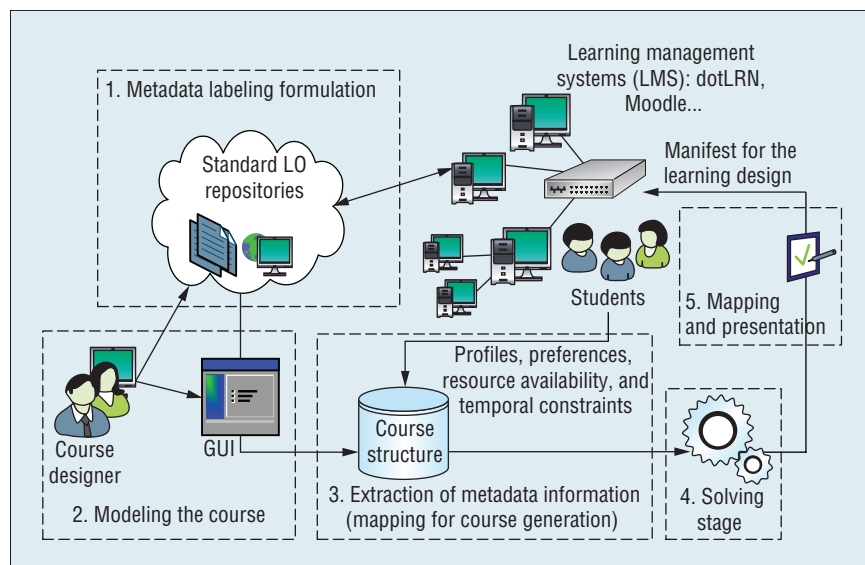

Figure 2. Five-step schema for personalized learning. It comprises: course design by probably using standard repositories with labeled learning objects (LOs), automatic extraction of metadata information, personalization of the learning routes and, finally, presentation of the routes in LMSs.

However, there are some limitations. First, not all LOs are fully compatible or freely able to be assembled. Second, LOs can only be combined under certain conditions, and dealing with coherent metadata information
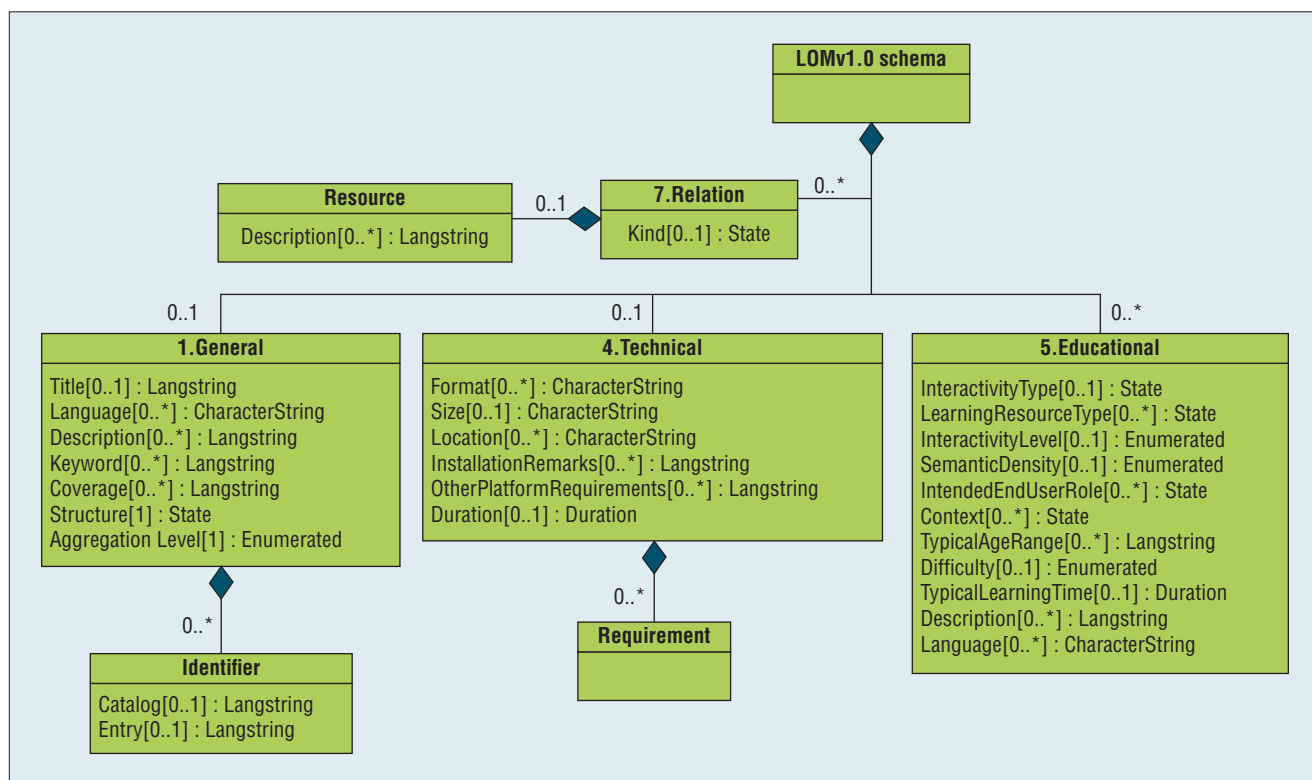

Figure 3. Simplified view (four out of the nine main elements) of the Learning Object Metadata (LOM) base schema (http://en.wikipedia.org/wiki/Learning_object_metadata). Image used under the terms of the GNU Free Documentation License.

is crucial: "LO1 *Requires* LO2" and "LO2 *Requires* LO1" would entail a contradiction, although it's automatically detected and avoided during modeling. Third, the course designer needs some experience with LOs in a given domain to accomplish a high degree of personalization.

Extracting Metadata Information

The mapping for course generation is a translation process that extracts LO metadata information and automatically builds the course structure through the causal dependencies between LOs and their adaptation aspects. Note that there isn't a unique mapping, as it would depend on the techniques used in the solving stage—for example, a set of formulas for mathematical models, an action-based formulation for planning, or a constraint satisfaction problem for constraint programming. The underlying idea in any mapping is to process the course's LOs and include student information (background, profile, learning goals, and temporal or resource constraints) to create a course structure that a particular solver can use as an input.

Solving Stage

The result of the solving stage, regardless of the technique used, is a tailored learning design: a sequence of LOs that suits each student's preferences and necessities. From a pedagogical perspective, a tailored design is simply a collection of LOs, but the temporal and resource constraints must be contemplated in a real scenario. Thus, some LOs might be associated with a time stamp and to a particular resource—for example, an LO that requires the use of a shared microscope is only available when the lab is open.

Mapping for Presentation

The presentation mapping step is another translation process to transform the calculated learning design into a standard language (manifest) for the LMS's use. Although current LMSs support different languages, such as IMS-LD for dotLRN or Moodle templates, the compilation algorithm is quite general. For each student, the process generates one or more documents that include the learning goals, the prerequisites and previous knowledge, the roles, the activity structure (which represents the LOs and their orderings), and the resources required by the LOs. After uploading these documents on an LMS, the student progressively navigates through the contents, avoiding exposure to all the course's LOs.

Achieving Full Adaptation via Planning Techniques

From an AI planning perspective, we can exploit the previous general adaptation requirements even further. The idea is to adapt profiles to LOs—and thus generate extremely flexible learning designs—by extending the modeling, metadata extraction, and planning techniques (steps 2, 3, and 4 of Figure 2).

Modeling (and Extending) the Course

Let's revisit the Java course and the two students we introduced earlier to explain the potential of planning for modeling adaptation. Observe the two LOs at the bottom of Figure 4, where LO1 (*An example*) requires students with Programming skill = {High, Medium}, whereas LO2 (*Documentation*) is adequate only for Felder's verbal learners. A flexible adaptation for profile-dependent scores or grades is also possible. In the case of LO1, if the student's programming skill is high, he or she will get the

max value (100 percent) of the competence level or score, but if the skill is medium, this value is reduced to 70 percent. This means that a more skillful student, such as Rebecca, will get a better outcome from this LO than a medium one, such as John.

Moving beyond, we can model a higher level of adaptation, as shown in LO3 (*Classes and objects*) in the top part of Figure 4. Given two levels for previous OOP knowledge—namely, high and low (Rebecca's and John's, respectively)—LO3 is valid for any type of student, but if the student's previous OOP knowledge is high, the prerequisite for this LO is lower than if it were low.

Additionally, the LO3 outcome is higher when the student's profile is high and smaller if it's low. This means that when using LO3, a more OOP-experienced student such as Rebecca will require less effort than John. After doing this LO, a student with more OOP experience will become more competent in Java classes than a novel student who has just started with OOP. Analogously, we can model temporal and resource constraints for context adaptation, such as, for instance, that LO1 requires a computer or that LO3 must be done in a collaboration group, which entails synchronization constraints among the students (and other constraints to arrange/attend a seminar).

Extracting Metadata Information for Planning

The metadata information extraction stage analyzes student information and iterates over the LOs to generate one standard Planning Domain Definition Language (PDDL) action per LO and student, as detailed in Figure 5's mapping. This compilation is highly efficient, as each action comprises four entries automatically extracted from the values of the LO

Figure 4. Exploiting planning for adaptation of profiles to LOs, requirements, and outcomes. By using planning, LOs can be extended to include more elaborate requirements and outcomes better adapted to the students' profiles.

metadata: name, with the LO name; duration, with the LO learning time; conditions, based on the profile's dependencies plus the relations defined in its metadata; and effects, based on the learning outcomes. This process also validates and matches preconditions and effects, thus detecting incorrect LO metadata and discarding unfeasible actions.

The PDDL action for Figure 4's LO3 adapted to Rebecca ("OOP previous knowledge = High" profile) is as follows:

```
(:durative-action LO3_Classes_
and_Objects_Rebecca
  :duration (= ?duration 50)
```

```
:condition (and ;; LO3 not
done yet and OOP competence
requirement
  (at start (= (LO3_Classes_
  and_Objects_Rebecca_done) 0))
  (at start (>= (OOP_
  Competence_Rebecca) 75)))
:effect (and ;; LO3 now done
and competence outcome
  (at end (increase (LO3_
  Classes_and_Objects_
  Rebecca_done) 1))
  (at end (increase
  (Classes_and_Objects_
  Competence_Rebecca) 100))))
```

Numeric values represent different competence levels and are useful to

model adaptation, costs, or learning rewards, which in turn lets us deal with multi-objective metrics.

The scheduling mapping is even simpler, as it directly maps student and LO constraints to a constraint satisfaction problem (see Figure 6), with variables representing each action's (LO's) start time and restrictions representing the dependency relations and temporal or resource constraints.[10] For instance, assuming that Rebecca and John perform LO3 in a collaborative group, the CSP asserts the following constraint: (LO3_ Classes_and_Objects_Rebecca = LO3_Classes_and_Objects_John). If LO3 needs a seminar that opens

Planning mapping		
LO metadata item	→	PDDL action entry per student
`general/title`		`action-name`
`educational/typicallearningtime`		`:duration` for temporal planning

dependency relations:
  type of relation:
    switch(`relation/kind`)
      case (*Requires*): conjunctive (and)
      case (*IsBasedOn*): disjunctive (or)
    the LO: `relation/resource/entry`
		`:condition`
		  `and (at start (= (action-name_done) 0))`, and
		if and-precondition: (`and …`
		else-if or-precondition: (`or …`
		for each `entry`
		  `(at start (≥ (action-competence) val))`
		  … other optional preconditions for profile adaptation taken from
		  `educational` item)

model that the action has been done, and increase the competence value, and increase a reward/utility (if any) due to:
  profile adaptation:
    `educational/learningresourcetype`
  additional adaptation:
    if `relation/kind` is *References*
		`:effect`
		`(and (at end (= (action-name_done) 1))`, and
		` (at end (increase (action-competence) val))`
		` (at end (increase (reward_student) val_LRT))`
		` (at end (increase (reward_student) val_Ref))`
		  … other optional rewards and/or costs)

Scheduling mapping		
LO information	→	Formulation in the CSP model (vars. & restricts.)
time stamp for the LO		variable `action-name` $\in [0, \infty)$, which represents the start time of the LO
ordering relations: `relation/kind` and others `relation/resource/entry`		`action-name`+duration(`action-name`) $\leq$ `action-name_entry`
temporal synchronization (collaboration) of LO among students `s1, s2 ... sₙ`		`action-name_s1=action-name_s2=...= action-name_sₙ`
capacity constraint for resource $R_i \in$ `technical/otherplatformrequirements`		capacity(R_i) $\leq \sum$ use(`action-name_sⱼ`, R_i), for all `action-name_sⱼ` that requires resource R_i

Figure 5. Planning and scheduling technical mapping for course generation—from LO metadata in Figure 3 to Planning Domain Definition Language (PDDL) planning and constraint satisfaction problem (CSP)-based scheduling.

from 1:00 to 3:00 p.m. (or from 780–900 minutes, if time is measured in minutes), the constraint is (`(LO3_Classes_and_Objects_Rebecca ≥ 780) AND (LO3_Classes_and_Objects_Rebecca + 50 ≤ 900))`, assuming 50 is the LO3 duration, which also restricts John's LO3 due to the previous equality. Note that this information is still uncommon in standard LOs and requires some extensions when designing the course, which can be easily modeled from a planning perspective.[13]

Solving Stage
Mapping the e-learning problem to a standard PDDL+CSP model facilitates the use of independent solvers and abstracts out the e-learning features from the planning and scheduling details. A whole description of

planning and scheduling technology is out of scope here, but more information about our solver appears elsewhere.[10] In short, the planner determines the best LOs, and the scheduler handles students' sharing temporal and resource constraints, thus deciding when and how to use such LOs.

Evaluation and Discussion
We can evaluate our approach from a quantitative perspective that measures system response and a qualitative perspective that measures the benefits and effort for course designers and students.

Quantitative
We tested our approach on four AI courses (short, short-medium, medium-long, and long, with approximately 10, 20, 40, and 80 LOs,

respectively) in a repository of 172 LOs for different learning styles. We targeted plan quality and system scalability, considering only planning and planning+scheduling (synchronization and resource consumption constraints, randomly generated, on LOs). We conducted two experiments in our solver to minimize makespan (shortest plan) and maximize the students' learning rewards based on the LOs that best fit them.[10] We defined problems with 1, 2, 4, 8, ... 256 fictitious students with different profiles and ran the experiments on a 2.33-GHz Intel Core 2 Duo CPU with 3.23 Gbytes of RAM.

Figure 6 shows the results. Obviously, the maximization problem involves longer plans, which degrades performance. Including the scheduling constraints (+SC) increases the
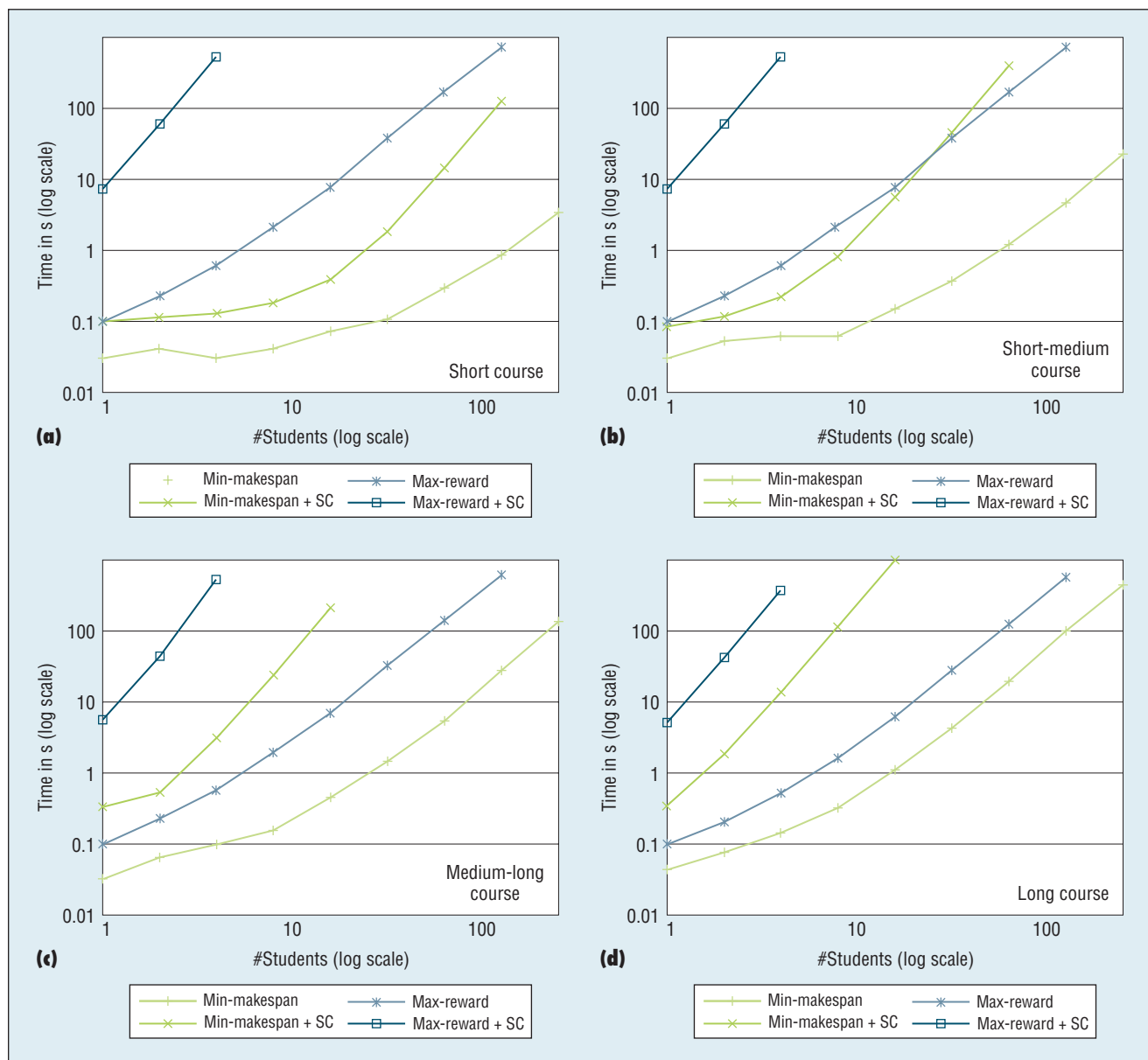
Figure 6. Experimental results for two different metrics and two options, planning and planning+scheduling constraints (+SC). All tests were censored after 360 seconds of running time. Performance shows scalable for 100 students when doing (only) planning and up to 10–20 students when doing planning+scheduling.

complexity of the course and makes the problem more difficult to solve, which means an expectable negative impact on performance. In this situation, we can only solve problems with fewer students (mainly in longer courses). This general issue was due to CSP solver complexity—our embedded scheduler handled up to 2,000 variables for the LOs in a reasonable time, but anything extra exhausted the allocated time.

Ultimately, scheduling reduced scalability by approximately one order of magnitude. However, the performance was still reasonable and let us manage scheduling constraints in groups of 10 to 20 students, even for long courses.

Qualitative

Personalization approaches are challenging, and the horizon is still unclear.[3] Adapting LOs to learning styles is a big shift from the conventional

way of teaching: some lecturers were reluctant to redesign LOs and courses to fit different profiles. Moreover, assembling LOs for fully adapted instructional courses requires experience and training, which sometimes handicaps IT-illiterate lecturers. The demand for self-learning and IT approaches continues to grow, but the application of planning technology can ease the incremental construction of tailored courses.[13]

Table 1. Questionnaire for a qualitative evaluation of an AI course.*

Question	Very little	Little	Neutral	Much	Very much
Q1				3	7
Q2	2	5	3		
Q3		2	3	4	1
Q4			2	2	6
Q5	2	3	3	2	
Q6	4	5	1		
Q7			2	6	2
Q8			3	5	2
Q9				4	6
Q10				2	8
Q11	4	4	2		
Q12				3	7

Q1. Is the sequence of LOs consistent with the objectives of the course?
Q2. Is the size (number of LOs) of the course appropriate?
Q3. Is the duration of the course appropriate?
Q4. Do you think the learning route and contents are adapted to the student's profile?
Q5. How much experience do you need to deal with these LOs?
Q6. How much planning background is necessary?
Q7. Do you consider this approach useful?
Q8. Would you recommend this approach to other lecturers?
Q9. Do you find e-learning as positive and motivating as traditional teaching?
Q10. To which extent did the course fit your needs and constraints?
Q11. Would you suggest some changes in the course structure?
Q12. Would you recommend this approach to other students?

* The first block evaluates the course contents and structure, the second block the lecturers' opinion on using LOs, and the third the students' opinions on using this course.

To assess lecturers' opinions on course contents and adaptation to student profiles, we designed a questionnaire for a long AI course; see Table 1. Ten lecturers who regularly teach AI to graduate students answered our survey. We also gathered the opinions of 10 students for this course. Generally speaking, the lecturers agreed with the flow automatically generated (by the planner) of the learning designs. However, lecturers missed the particular, pedagogical organization of LOs that they manually include in their way of teaching in the form of advice or recommendations to students. Although they generally agreed with the course composition and causal dependencies among LOs, they felt that the course's overall structure was too open to students.

As for students, the experience was highly positive. They found the provided LOs to be a helpful resource for catching up on the background required for the course and grasping key ideas as well as an ideal mechanism for self-assessment.[1] Student motivation also stems from the ease of signing up for the course. No previous training was required; they just had to classify themselves in one or more learning styles (www.engr.ncsu.edu/learningstyles/ilsweb.html) and define their background, preferences, learning outcomes, and, optionally, their temporal or resource constraints. The most outstanding result among students was the degree of satisfaction with the course's self-organizing activities. This approach let each student have a personalized learning route that was especially designed for their profile, which makes it easier to fit the student's personal and temporal restrictions.

We can affirm that students generally seem more enthusiastic about e-learning than lecturers. From our experience, the use of planning technology proved to be very successful in promoting adaptation. But it's important to note that the success of this approach can't be directly assessed through student grades: following a fully adapted path doesn't necessarily mean a better score. However, it does require students to study effectively and quickly, which provides a higher motivation for using LOs that fit their preferences and learning styles.

So although planning technology is highly appreciated by students and less popular among lecturers, who are somewhat reluctant to give up their traditional role of course planners, reality shows that in the context of Web libraries, it's difficult for a lecturer to create fully personalized plans for students that meet their personal constraints, too. Both students and lecturers can agree that the flexible application of planning techniques can provide the right content to the right person at the right time.

References

1. E. Guzman, R. Conejo, and J.L. Perez de la Cruz, "Improving Student Performance Using Self-Assessment Tests," *IEEE Intelligent Systems*, vol. 22, no. 4, 2007, pp. 46–52.
2. L.P. Santacruz-Valencia et al., "Comparison of Knowledge during the Assembly Process of Learning Objects," *J. Intelligent Information Systems*, vol. 35, no. 1, 2009, pp. 51–74.
3. T. Tiropanis et al., "Semantic Technologies for Learning and Teaching in the

Web 2.0 Era," *IEEE Intelligent Systems*, vol. 24, no. 6, 2009, pp. 49–53.

4. "Draft Standard for Learning Object Metadata," rev. 16, IEEE, Feb. 2005; http://ltsc.ieee.org/wg12/files/IEEE_1484_12_03_d8_submitted.pdf.

5. M. Schindler and D. Vrandeccic, "Introducing New Features to Wikipedia: Case Studies for Web Science," *IEEE Intelligent Systems*, vol. 26, no. 1, 2011, pp. 56–61.

6. V. Tamma, "Semantic Web Support for Intelligent Search and Retrieval of Business Knowledge," *IEEE Intelligent Systems*, vol. 25, no. 1, 2010, pp. 84–88.

7. R.M. Felder and L.K. Silverman, "Learning and Teaching Styles in Engineering Education," *Eng. Education*, vol. 78, no. 7, 1988, pp. 674–681.

8. P. Mika et al., "Learning to Tag and Tagging to Learn: A Case Study on Wikipedia," *IEEE Intelligent Systems*, vol. 23, no. 5, 2008, pp. 26–33.

9. C. Ullrich and E. Melis, "Complex Course Generation Adapted to Pedagogical Scenarios and Its Evaluation," *J. Educational Technology and Society*, vol. 13, no. 2, 2010, pp. 102–115.

10. A. Garrido and E. Onaindia, "On the Application of Planning and Scheduling Techniques to e-Learning," *Proc. 23rd Int'l Conf. Industrial Eng. and Other Applications of Applied Intelligent Systems* (IEA-AIE 2010), LNAI 6096, Springer, 2010, pp. 244–253.

11. N. Idris, N. Yusof, and P. Saad, "Adaptive Course Sequencing for Personalization of Learning Path Using Neural Networks," *Int'l J. Advances in Soft Computing and Its Applications*, vol. 1, no. 1, 2009, pp. 49–61.

12. E. Kontopoulos et al., "An Ontology-Based Planning System for e-Course Generation," *Expert Systems with Applications*, vol. 35, nos. 1–2, 2008, pp. 398–406.

13. A. Garrido et al., "Modeling e-Learning Activities in Automated Planning," *Proc. 3rd Int'l Competition on Knowledge Eng. for Planning and Scheduling* (ICKEPS-ICAPS 2009), 2009, pp. 18–27.

THE AUTHORS

Antonio Garrido is an associate professor of computer science at the Universitat Politecnica de Valencia in Spain. His research interests include planning, scheduling, constraint satisfaction, search, and agreement technologies in AI, and their application to engineering and education fields. Garrido has a PhD in computer science from the Universitat Politecnica de Valencia. Contact him at agarridot@dsic.upv.es.

Eva Onaindia is an associate professor of computer science at the Universitat Politecnica de Valencia in Spain. She currently leads the Reasoning on Planning and Scheduling (http://users.dsic.upv.es/grupos/grps/) group, conducting research in temporal and classical planning, development of integrated techniques of planning and scheduling, and multiagent planning. Onaindia has a PhD in computer science from the Universitat Politecnica Valencia. Contact her at onaindia@dsic.upv.es.