On supporting the process of learning design through planners

Miguel-Angel Sicilia, Salvador Sánchez-Alonso and Elena García-Barriocanal

Information Engineering Research Unit, Computer Science Department,
University of Alcalá – Ctra. Barcelona km. 33.6
28871 Alcalá de Henares (Madrid), Spain
{msicilia, salvador.sanchez, elena.garciab}@uah.es
http://www.cc.uah.es/ie

Abstract: Recent standardization in learning technology has resulted in a model of activity-based learning designs called IMS LD that provides a common framework for expressing any kind of activity-based learning program. This emphasis on designing learning programs based on assembling activities and resources in turn allows the application of Artificial Intelligence (AI) techniques to help in the process of design. However, learning design is a problem of open rationality, and the applicability of computational techniques is a controversial matter in itself. This paper explores the applicability of common planners to the (partial) automation of learning design, and provides the general guidelines for the design of pedagogical designer agents. Even though such designers can not provide unique or deterministic solutions – due to inherent characteristics of the problem – they can be equipped with different "rationalities" about human learning, eventually leading to new insights in the conceptions of learning, gained through the observation of computational models that embody the main principles and action guidelines of pedagogical design approaches.

Keywords: Learning design, planners, learning objects.

1. Introduction.

Recent research related to the concept of *learning design* emphasizes 'inductive learning design' models and approaches (Koper, 2004), in which past accumulated experience is used as a source for guidelines or rules in the design of learning programs. This emphasizes the elaboration of models that codify past pedagogic design experience. Such models include patterns (often defined as 'proven solutions to recurring problems') that are inductively abstracted from real practice, i.e. by examining and finding commonalities among (successful) learning designs. In addition, this kind of design approach complements the representation of different pedagogical ontologies (Sicilia and Lytras, 2005) as a mean for the codification of design guidelines. In both cases, the underlying idea is that of representing some

F. J. García et al. (Eds.),
Virtual Campus 2006 Post-proceedings. Selected and Extended Papers, 81-89

pedagogical knowledge that informs the subsequent process of design of learning experiences.

Such knowledge-based approaches to design reveal a renewed interest in the use of knowledge representations for the intellectual process of learning design, and *Artificial Intelligence* (AI) techniques become thus candidates for the realization of tools that help in the process of design. This could be considered as an instance of the class of *expert systems*, but with the broad problem category of "designing learning activities".

The automated design of learning sequences is not a novel idea, but several authors have approached tools that aggregate contents into higher levels of instruction (Vassileva and Deters, 1998), use past activities to improve new ones (Elorriaga and Fernández-Castro, 2000) or that consider planning of learning activities (MacMillan and Sleeman, 1987). However, the sort of *lingua franca* for the results of design provided by IMS LD[1] provides new opportunities to engineer solutions that are not restricted to concrete domains or constrained applications, but rather promote the sharing, comparison and competition of different models. This is accomplished by providing a shared common language with a concrete interpretation (Koper, 2004).

The IMS LD provides a powerful language for the expression of learning designs targeted at the realization of activities. An *activity* is considered as a piece of interaction among a number of specified *roles* played by persons that produce a tangible *outcome* by using a concrete environment made up of learning objects and services (facilities available at runtime). Activities can be further decomposed in sub-activities, and they are aggregated into *methods*, that specify the conditions for application of the learning design, along with the planned objectives that will eventually match the outcomes of the activities. Methods can be structured around concurrent *plays* and these in turn can be structured in sequential *acts*, the latter allowing the specification of execution conditions. This schematic description of LD gives an idea of the flexibility the specification provides in describing activity-based learning programs. The practical use of LD-based tools would then allow for the definition of the activities resulting from a process of instructional design that takes as point of departure a concrete perspective about learning that drives the crafting of the activities.

Planning in AI (Russell and Norvig, 2002) has been defined as the process of search and arrangement of a sequence of actions that allow the attainment of a concrete objective. From this definition, we can consider learning design as a planning problem, in which the objectives are some expected learning outcomes, and the actions are the possible learning activities.

But the concept of *learning design* mentioned so far should be understood in terms of the concept of "expandable rationality" as described by Hatchuel (2002), integrating creativity and unexpected expansions of the original requirements. This precludes ontological definitions in which the problem space is completely bounded *a priori*. In consequence, a degree of openness is necessary to integrate different kinds of detail in description, from fully described ones to others with shallower semantics, e.g. some providing only references to generic assumptions. This entails that the use of planning algorithms for the problem is fairly different from the classical bounded-

[1] http://www.imsglobal.org/learningdesign/

space problems for which they were designed. This paper explores the application of existing planning technology as an aid for the learning design process, as conceptualized by the IMS LD model.

The rest of this paper is structured as follows. Section 2 discusses general problems and alternatives of the application of planners to produce learning designs. Then, the principal overall representation issues are approached in Section 3. Then Section 4 illustrates the differences between diverging theoretical positions on learning when used as built-in knowledge inside a classical planning problem.

2. Problems and alternatives in the learning design context.

The environments of classic planning problems are completely observable, determinist, finite, static and discrete. In our case, the objective of planning is producing knowledge in the target learners or participants (in a general sense, possibly including competencies, creating attitudes or developing social ties). However, the problem of learning design diverges from classic settings in several aspects:

- Learning through predefined activity arrangements is never *determinist*, since the background, attitudes and even the personal matters of learners at a given time impact the outcomes of the learning experience.
- Some learning theories consider learning as a process of “construction” which makes learning a *non-discrete* process. This is since according to such views, the combination of the knowledge acquired may produce qualitative changes in the individual when combined with others.
- The environment of a learner is never completely *static* or *observable*, since the other activities of the individual may impact the learning process in unexpected ways.
- The space of possible activity arrangements can not be considered static in practical terms, since the possibilities of combining materials, activities, feedbacks and other elements admits a large wealth of variation, since the materials for the design today include resources available through global networks as the Web, in which new resources appear at a quick pace.

Activity programming introduces time in the consideration of planning, but the issues just described remain a problematic issue with that extension.

The conclusion from the above is that planning algorithms could never result in perfect learning design. This is a problem of an epistemological nature, since a complete environment definition would entail the knowing of the “mind” of the learners, which is a well-known philosophical controversy (Brook and Stainton, 2000) and it is in any case a hard problem for knowledge engineering. The conclusion therefore could be that planning algorithms are not suited for the problem described. However, the problem lies in that the problem addressed is to date an “expandable rationality” problem, for which humans only have some guidelines, heuristics or general intuitions.

Then, the emphasis must be shifted from the design of (definitive) solutions to the contrast of the solutions provided by different reasoning paradigms. This leads to the concept of *pedagogical design algorithm*. Different planning algorithms can be devised that work with different hypothesis, principles or guidelines on human learning (Sicilia and Lytras, 2005). For example, a “socio-cultural” agent would try to assemble learning activities that emphasize and require collaboration and role playing.

On the contrary, a “content-oriented” agent would give primary consideration to the quality of the contents and their prerequisite, perhaps ignoring issues related to learner interaction.

In consequence, the importance of the application of planning models to learning design relies in the possibility of contrasting different design outcomes that are the results of the application of a concrete collection of hypotheses, guidelines or assumptions about learning. Such contrast could only come from a disciplined inquiry approach with the following main aspects:

- The representation in formal terms of the different action approaches that are a consequence of different pedagogical standpoints.
- The connections of such action plans to actual learning designs.
- The *a posteriori* examination of the learning outcomes and difficulties of learning designs that were created following the different alternative standpoints.

This represents both a device for generating alternative designs – so that human designers can then compare and assess the different options – and also a vehicle for inquiry on the consequences of existing theoretical standpoints have in actual designs. More on the representation of pedagogical approaches as applied to IMS LD is described in (Sicilia, 2006).

3. Representing resources, participants: and overall approach?

Declarative representations allow for the generic modeling of design problems. AI planners are a candidate for learning design, and here we concentrate on the classical family of hierarchical planners. Hierarchical task network planning (HTN), is a efficient planning technique that offers a relatively straight-forward way for representing human expert knowledge. It incorporates heuristic knowledge in the form of the decomposition rules: A planning problem is represented by sets of tasks, methods decompose non-primitive tasks into sub-tasks until a level of primitive tasks is reached, which can be solved by operators. For each task, there may be more than one applicable method, and thus more than one way to decompose the task into subtasks. The rest of this paper provides examples and use the terminology specific to JSHOP, a well-known planner (Nau et al., 1999) that has been used in several applications.

Our approach diverges from other uses of planning in that the tasks to be arranged are those that *generate* an IMS LD instance and not tasks that represent *actual* teaching or learning activities. This clearly shifts the focus of the designer to solving the problem of generating an appropriate design for the given objectives and requirements. The following JShop2 fragment can be used to illustrate the main issues of such approach. The idea is that the methods of the planner create during the selection of the tasks to be done a representation of an IMS LD structure – such structure will eventually be executed in a conventional IMS LD engine in a later phase, after the examination of the outcomes of the planning process by the human decision maker – tutor, instructor, designer or the like.

```
(:method  ; head
        (learning-design ?goal ?learner)
        ; preconditions
        (
          (learning-goal ?goal) (learner ?learner)
        )
        ; subtasks
        (
         ; Create the initial definition for the LD
          (!create-LD-structure ongoingLD)
           ; start the creation of activities.
          (create-activity ongoingLD ?goal ?learner)
         )
      )
```

The `ongoingLD` represents the LD structure being created for the given goal(s) and learner(s). This generic design left open the representation of goals and the different ways in which the design could be created. In other words, different "design agents" will provide different methods to elaborate `create-activity` (this activity creation can be extended to creating *acts* or *plays*, according to the IMS LD model and structure briefly sketched in the introduction of this paper).

The representation of objectives can be stated in terms of logical functions as `matchs(ongoingLD, goal)`, which can be formulated in terms of checking that there exist activities associated to the created LD structure so that their objectives cover all the goals previously stated.

Representing resources

Resources inside designs in IMS LD are learning objects and services. The latter stand for any run-time service as a chat room that can be used during activities and the former can be interpreted as learning contents in general.

One important element in representing resources is that such resources in the Web are assumed in the learning object paradigm to reside in repositories. This entails that these elements are not bounded *a priori* in the problem, so that there is a need to modify the matching functions of the planning algorithm to allow the search outside the closed world of the problem domain. This in JSHOP by implementing the interface `Calculate`. The newly implemented function calls are able to do any kind of search, e.g. calling search services of distributed learning object repositories.

An alternative solution could be that of leaving learning objects "unbounded" till the runtime of the learning design. However, this is not consistent with the present approach, in which the planner is expected to obtain concrete designs that will be assessed by human designers.

A basic operator `deliver-lo` can be used as a basic action of delivering a learning object to given learners. However, it should be noted that such objects could in fact be of a considerable complexity, including interactive features and navigation. The LO delivered must be determined at design time, i.e. when the planning is getting resolved. Thus, the process of design has available the LO that are in the repositories selected at the moment of planning execution.

The description of learning objects can be represented as straightforward logic-based translations of the LOM standard or other models, e.g:

```
(LOMidentifier lid1)
       (catalog     lid1   URI)
       (entry lid1   http://www.dnaftb.org/dnaftb/)
(LOMtitle lt1  DNA-from-the-Beginning)
(LOMLanguage llang1)
...
(LEARNING-OBJECT lo1)
       (has-identifier     lo1    lid1)
       (has-title          lo1    lt1)
        ...;
```

Even though this appears as a simple translation of the LOM Standard, the semantics of the representations needs to be normative as described by Sánchez-Alonso and Sicilia (2005).

Representing participants

The representation of participants is based on learner profiles. Instances of `learner` represent profiles, and `learner-pools` can be used to represent indefinite number of potential learners with the same profile. The latter could be used to enhance the planner algorithm with the computation of “optimal” student group sizes.

The representation of the characteristics of learners is pedagogical agent-specific. This entails that different conceptions of learning will end up with different notions of change (Sicilia and Lytras, 2005) that require different models for the learners, and in some cases, for groups of learners understood as social units.

The typical conditions for the selection of learner profiles can be simply represented as functions as the following:

```
(:- (hasPrerrequisites ?learner ?pre)
        (   (knows ?learner ?pre)   )
    )
```

4. Contrasting a content-oriented planner with a socio-cultural planner.

As an example of the different possible pedagogical reasoning agents, in this section, two diverging cases are sketched. The examples are not intended as definitive typical cases, but only as illustrations of positions that can be found in research reports.

A “content-oriented” planner would essentially follow the same approach taken by common educational adaptive hypermedia systems that are based on selecting contents based on matching hierarchies of concepts to user models that represent the knowledge and objectives of individuals, e.g. (Brusilovsky, 2003). The following method sketches an example of such kind of reasoning.

```
(:method   ; head
          (create-activity ?ld ?goal ?learner)
          ; preconditions
          ; there should be a learning object that has
          ; as post-condition the concept specified in the goal
          ( (learning-object ?lo) (concept ?c)
           (postcondition ?lo ?c) (outcome ?goal ?c))
          ; subtasks
          ((!create-LD-Activity ?ld ?lo))
 )
```

To accomplish the goal of creating an activity (and adding it to the ongoing LD structure) for a given goal, a contract-based approach can be used that search for the appropriate `learning-object`. As discussed above, some mechanism of external search is required for this. Goals can be stated in terms of concept hierarchies as it is common in many adaptive hypermedia approaches. The hierarchies can be organized in several dimensions as illustrated in the following example.

```
(concept basic-programming)
    (subconcept oo-programming basic-programming)
    (subconcept procedural-programming basic-programming)
    (prerrequisite procedural-programming oo-programming)
```

This way, prerequisites could lead to two different courses of action: (a) checking id the learner profile fulfills such prerequisite; and (b) trying to assemble learning objects to cover such prerequisites. However, (b) could lead to designs of unreasonable extension due to the chaining of learning objects to cover prerequisites.
Competencies instead of concept hierarchies could also be used (Sicilia, 2005), but this again will end up with a match of goals to contents (learning objects). However, many other issues are to be considered. If we consider the framework of Conole et al. (2004), the Social-Individual axis could give pre-eminence to activities in which the “content objects” are not of special relevance. For example, a socio-cultural reasoning procedure could be sketched as follows.

```
(:method   ; head
          (create-activity ?ld ?goal ?learner)
```

```
        ; preconditions
        ; the target learner is actually a collection of roles
        (
           (isGroup ?learner)
        )
        ; subtasks
        (
          (find-services ?goal ? learner ?srv)
          (!create-LD-Activity ?ld ?srv ?act)
          (!addUserRoles ?act ?learner)
        )
)
```

The principal difference with this second approach is that it is not driven by concepts but rather by creating group activities in which: (a) appropriate services are selected, according to the goals and the profile of the learners, (b) activities are multi-role, and the concrete role of the participants could be subject of configuration also.
Both kinds of approaches can be combined but in that case it is important to clearly give the structure of the methods some kind of “metric” that help in deciding which of the approaches is considered more important. Immediate tasks in JSHOP could be used for that purpose, but some richer mechanisms of priority would be preferable in complex cases.

5. Conclusions and future work.

The standardization of activity-based learning program templates achieved by the IMS LD model provides renewed opportunities for the application of AI technology to aid in the intellectual process of learning design. This paper has explored the main issues of applicability of classical AI planners to that problem, under the departure assumption that learning design is not a classical planning problem but one of expandable rationality. Thus, the emphasis is on contrasting the design-reasoning of different pedagogical agents. Such contrast enables the creation of alternative designs that can be considered by human designers, and they also serve as models of the reasoning procedures for different conceptions on learning.

The work ahead is that of completing the specification of several archetypical pedagogic agents, and the construction of a base of problems that can be used for a meaningful and significant contrast of the different positions.

Acknowledgements.

This work is supported by project ELSEM, UAH-PI2005-070 – University of Alcalá.

Referentes.

Brook, A. and Stainton, R.J. (2000). *Knowledge and Mind*. Cambridge, MA: The MIT Press.

Brusilovsky, P. (2003). Adaptive navigation support in educational hypermedia: The role of student knowledge level and the case for meta-adaptation. *British Journal of Educational Technology*, 34 (4), pp. 487-497.

Conole, G., Dyke, M., Oliver, M. and Seale, J. (2004). Mapping pedagogy and tools for effective learning design. *Computers & Education*, 43(1-2), pp. 17-33.

Elorriaga, J. and Fernández-Castro, I (2000). Using case-based reasoning in instructional planning. towards a hybrid self-improving instructional planner. *International Journal of Artificial Intelligence in Education* 11, pp. 416–449.

Hatchuel, A. (2002). Towards Design Theory and expandable rationality : The unfinished program of Herbert Simon, *Journal of Management and Governance*, 5(3-4).

Koper, R. (2004). Use of the Semantic Web to Solve Some Basic Problems in Education: Increase Flexible, Distributed Lifelong Learning, Decrease Teacher's Workload. *Journal of Interactive Media in Education*, 2004 (6).

MacMillan, S.A and Sleeman, D.H. (1987) An architecture for a self-improving instructional planner for intelligent tutoring systems. *Computational Intelligence*, (3), pp.17–27

Nau, D., Cao, Y., Lotem, A. and Muñoz-Avila, H. (1999) SHOP: Simple Hierarchical Ordered Planner. In *Proceedings of IJCAI-99*, pp. 968-973.

Russell, S. and Norvig, P. (2002). *Artificial Intelligence: A Modern Approach*. Second Ed. Prentice Hall.

Sánchez-Alonso, S. and Sicilia, M. A. (2005). Normative Specifications of Learning Objects and Learning Processes: Towards Higher Levels of Automation in Standardized e-Learning. *International Journal of Instructional Technology and Distance Learning*, 2(3), pp. 3-12.

Sicilia, M. A. (2005). Ontology-based competency management: Infrastructures for the knowledge-intensive learning organization. In M. D. Lytras and A.Naeve (Eds.), *Intelligent learning infrastructures in knowledge intensive organizations: A semantic web perspective*, pp. 302-324, Hershey, PA: Idea Group.

Sicilia, M. A. and Lytras, M. (2005). On the representation of change according to different ontologies of learning. *International Journal of Learning and Change*, 1(1), pp. 66-79.

Sicilia, M.A. (2006). Semantic learning designs: recording assumptions and guidelines. *British Journal of Educational Technology*, 37(3), pp. 331-350.

Vassileva J. and Deters, R.(1998) Dynamic Courseware Generation on the WWW. *British Journal of Educational Technology*, 29(1), pp. 5-14(10).