Abstract

Designing a computer-supported learning scenario involving a constructivist approach of learning lays on a paradox. On the one hand, learning flows must be precisely described – including role distribution, required resources, tools, and scaffolds – to be realized in computer-supported environments. On the other hand, a fine-grained formalization of learning flows diverges from the constructivist notion of learning that learners are responsible for their knowledge (co-)construction. This paper draws upon the fundamentals of a new concept so called “Learning Activity Space” (LAS) aiming at realizing some necessary flexibility when designing learning scenarios. LAS is the basis of a graphical scenario modeling language intelligible for non-computer scientists but still rich enough in detail to describe a large set of computer-supported learning scenarios.

1. Context of LAS’ emergence

Design of computer-supported learning environments is supposed to build on pedagogical theories [1] as well as take advantage of technological potentials like automation, adaptability and reusability. Various Educational Modeling Languages (EML) have been developed to support design of pedagogically sound and technologically advanced computer-supported learning environments [2].

Comparison of EMLs [3], [4] reveals a number of vital issues and challenges:

- **Comprehensibility**: how can EMLs be made usable for educational practitioners [5], [6]?
- **Pedagogical neutrality**: how can an EML realize one unified, pedagogical neutral notation for supporting a large variety of pedagogically sound scenarios [7], [8]?
- **Flexibility**: how can EMLs support design of well-supported, but flexible environments [9], [10]?
- **Interoperability**: how can EMLs build on existing learning platforms and contents [11]?

In addition to this non-exhaustive list of currently debated issues, an under-investigated, but crucial aspect remains without practicable answer among the existing technology enhanced learning solutions and systems: how to specify and further manage at run-time learners’ artifacts (data, hypotheses, arguments, documents, etc.) and the evolutions of these artifacts throughout the learning process?

We claim the de facto standard modeling language IMS LD and its potential alternatives require considerable expertise to use and fail to satisfy other of the aforementioned issues (e.g., pedagogically sound scenarios, flexibility). Some efforts have been made to facilitate design for non-specialists like, for example, with COLLAGE, a high-level IMS-LD compliant authoring tool that is specialized for computer-supported collaborative learning [12]. The use of customizable patterns [12] effectively fosters the design process. However, even if IMS LD is hidden to teachers – as in the case of COLLAGE – this EML remains the constraining backbone of such LDs.

To move beyond IMS LD, the next main challenge for improving EMLs is to provide an intuitive notation...
to support educational practitioners to not only understand, but also describe a large number of flexible scenarios themselves. Another challenge is to utilize the scenario representations produced with an EML to support teachers becoming aware of and facilitating current learning process.

The concept of Learning Activity Space (LAS) [13], and its graphical representation, which were refined and developed in the context of an interdisciplinary European project named SCY (Science Created by You), are first attempts to meet these challenges.

We define a LAS as a coherent and intuitive set of activities supported with specific tools and scaffolds. The input and output of a LAS are described in terms of a set of artifacts created by students that we consider as “Emerging Learning Objects” [14].

Thus, the general outline of a learning scenario can be defined by a collection of LASs and the various learning paths between them.

This paper presents a graphical modeling language based on LAS, which meets the aforementioned challenges at the design phase of computer-supported learning scenarios: (1) embedding all core elements of learning scenarios including the learners’ artifacts, (2) guaranteeing a sufficient level of flexibility without sacrificing comprehensibility, and (3) allowing designers and practitioners to visualize, adapt, or build from scratch the desired scenarios and plans without detailed technical knowledge. To make this flexible and expressive LAS-based EML easily comprehensible for non-computer scientists, LAS and its components are represented graphically. A specific editor - described in part 3 – has been developed, allowing practitioners and designers to initially design, exchange and compare pedagogical scenarios and LASs.

2. A Graphical Modeling Language for computer-supported learning scenarios

Starting from the idea that learning involves cognitive activities, the design of learning environments can be built on a specified selection of learning activities, which constitute the respective LASs. This would allow building various pedagogical scenarios with a limited set of LAS and help linking learners’ activities with the desired learning outcomes [15]. The learners’ productions through these activities can be shared with co-learners and serve to assess learning processes and outcomes.

The LAS main components (see Figures 1 and 3) are activities, emerging learning objects, tools, and scaffolds. The grammar of the language describes the rules of how these components are connected and specified. Each activity requires an input that could be an emerging learning object developed previously in another LAS by the learners or a given learning object pre-defined by the content developers. The activities in a LAS result in at least one output emerging learning object, but in many cases there can be several of these. Additionally, the activities are supported by tools that may or may not have scaffolding characteristics, e.g., a simulation tool that – adapted to the learners’ changing needs – provides clues for what variables to look for in the simulation.

To arrive at a limited set of learning activities and LASs, we have selected a number of scenarios for science learning. Examples of scenarios are inquiry-based learning, design-based learning, argumentative knowledge construction, etc. Such an abstract scenario becomes what we call a “mission” when adding specific content and a pedagogical plan. Ten scenarios were described in terms of learner activities and, after homogenization, led to a list of 52 activities, such as “build consensus” or “run an experiment”. Next, we created a set of LASs at a grain size larger than these activities, which implied that in a LAS the activities are clustered into a coherent set of activities together with an appropriate set of tools and scaffolds. To adequately describe the ten scenarios we identified a total of only thirteen LASs, named: analysis, conceptualization, construction, debate, design, experiment, evaluation, information, management, orientation, reflection, regulation, and reporting.

The graphical language models scenarios on three levels of abstraction. Firstly, the language gives an overview of each LAS mostly defined by activities, emerging learning objects and tools. Figure 1 shows, as an example, the LAS called Design with its basic components.

![Figure 1: LAS Design.](image-url)
Secondly, a limited number of LASs are selected for a scenario and pedagogically relevant relations are made between these. Inter-LAS relations can figure the possible learning paths in respect with a specific pedagogical scenario.

Figure 2 shows how the scenario “design-based learning” (with the LAS design from Figure 1 included) is represented. This helps a practitioner (teacher, content developer) to understand the core ideas and aims of any particular scenario.

Thirdly, as the LASs are arranged to form a specific scenario, they are further specified by adding components that characterize the learning setting, namely, the learning arrangement, e.g., individual or collaborative, the location, e.g., home or classroom, and the foreseen support through technology, e.g., desktop computers or mobile devices. It enables the users to understand which skills will be developed according to the learning plan, and what resources, tools, materials, and equipment is needed for applying this scenario to a mission. Figure 3 shows how the LAS Design is specified for the scenario Design-based learning.

Let us describe how a designer might work with LASs to build a scenario. In the specific outline of a scenario, a designer has to (1) choose a specific set of LAS from the 13 available LASs, and then (2) specify each LAS. In step (1), choosing the right set of LAS is done with different constraints. Activities to be performed help selecting a set of LASs. The focus of the scenario can lead to choose certain LASs and put others aside. For instance, if the main focus of the scenario is project-based learning, a LAS like management is useful while it may be not essential for a shorter scenario. In step (2), each LAS can be further specified with respect to its setting (arrangement, technology support, and location). Thus, each LAS can have different instantiations, i.e. single activities can be left out or added without changing the character of the LAS (Figure 3 compared to Figure 1). Moreover, learning arrangement, location and technology are specified at this point (figure 3). The next step would be to adapt this scenario to a mission, adding specific content and a pedagogical plan to this scenario.

Pedagogical scenarios as they are substantiated in concrete missions can be thought of as a sequence of LASs. A crucial point here is that the sequencing of activities is not prescribed in a linear fashion; instead, LASs offer a number of resources, namely tools and scaffolds, which are available for students anytime they enter a LAS in order to make up their learning path. The option of choosing among alternative routes is in this sense the key to advance a constructivist mode of learning. Flexibility is facilitated in two dimensions, that is, an intra-LAS dimension and an inter-LAS one. The intra-LAS dimension can be promoted by choosing alternative routes in the sequencing of activities. Further, intra-LAS flexibility is reinforced by activities coming with alternative assignments. Although these might be brought under the same learning goal, they are different from the learners’ perspective. The inter-LAS dimension is facilitated by the increased variability that is provided for sequencing LASs during a mission.

3. Scenario and LAS editor

Based on the conceptualization and the suggested graphical language for modelling educational scenarios, a software application has been developed to support the creation of the above mentioned pedagogical scenarios and LAS in a graph-based modelling environment. This application has been built as an extension of the FreeStyler multi-purpose modelling tool [16]. FreeStyler had previously been extended with a somewhat similar plug-in for learning process modelling based on the IMS-LD approach [17]. The resulting MoCoLADe editor and simulation environment [18] allows for specifying particularly the process aspects of learning scenarios, including role taking and group formation processes. However, it is
less explicit about tool use and learning objects. Since the focus in this approach is on emerging learning objects, their relations and evolution and on the design and support of flexible collaborative learning processes a new tool was developed to edit educational scenarios and Learning Activity Spaces. An additional focus of the current development is the provision of similarity based recommendations for an exchange of scenario/LAS specification in a community of educational designers.

The so-called SCY Scenario Editor (SCY-SE, Figure 4) supports comfortable, graph-based and consistent creation of Learning Activity Spaces and pedagogical scenarios, allowing practitioners and researchers to initially design, exchange and compare new scenarios and LASs.

SCY-SE is able to retrieve and infer information from an ontology that is being built in the context of the SCY project. In this ontology, a set of pre-defined template scenarios and LASs (as listed above) are stored along with their relations to activities, tools, emerging learning objects, scaffolds, etc. Thus, SCY-SE assists the designer in adhering to constraints (e.g., an “interpret data” activity necessarily needs a tool to display data) and it is able to compare new or edited scenarios and activity spaces to existing ones. Similar scenarios or activity spaces may be joined or discussion between their authors may be initiated. SCY-SE provides two different views (with different objects and tools provided in a “palette”) for LAS specification (shown in Figure 4) and for the combination of LASs into an integrated scenario.

At the moment, the SCY-SE explicitly addresses the design time aspects of learning process modelling and specification. A future goal is to make use of the LAS descriptions and the Scenario Editor to configure and adapt the learning environment at runtime and to provide means to monitor the learners’ progress.

A machine-readable representation of LAS and pedagogical scenarios (as provided by the Scenario Editor) can be used to dynamically adapt and configure computer supported learning environments at runtime. Tools to be used might be proposed to the learner, collaborative sessions initiated, the user interface may be changed to the current needs etc. As done similarly in [20], together with a monitoring component, a visual representation of LASs and scenarios can be used to display the learners’ progress and current state in the course of a mission. This information may be useful for a learner, teacher, and/or moderator.

4. Conclusion

The notion of ‘Learning Activity Spaces’ (LAS) enables a structured and expressive description of learning scenarios and at the same time it allows for a high variability in interrelating activities within and between LASs. In this regard, our approach offers an opportunity to overcome the contradiction between prescribed routes and flexible designs. By means of the graphical language and the SCY Scenario Editor (SCY SE) we seek to involve learners, designers, practitioners, and software developers in the orchestration of activities in computer-supported learning environments.
5. References


Disclaimer: The SCY project is partially supported by the European Community under the Information and Communication Technologies (ICT) theme of the 7th Framework Program for R&D (FP7). This article does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.