Bridging the gap between instructional design and double loop learning

Howard Spoelstra[*,1]**, Maristella Matera**[2]**, Ellen Rusman**[1]**, Jan van Bruggen**[1] **and Rob Koper**[1]

[1] OTEC, Open University of the Netherlands, Valkenburgerweg 177, Heerlen, the Netherlands
[2] Politecnico di Milano, Via Ponzio, 34/5 20133, Milano, Italy

Keywords: e-learning; instructional design; double loop learning; workflow; atomic actions

There seems to be a gap between instructional design, as implemented in workflow-like e-learning systems, and double loop learning, because the former predefines educational processes and the latter assumes these processes can be reflected upon and can be modified by the learners. Instructional designs implemented in workflow-like e-learning systems that use modelling standards enable portability of the designs. While maintaining this advantage of portability, the COOPER environment aims to make room for learning reflection and influence as well. We will present a pedagogical scenario called "Virtual Company", which uses double loop learning extensively, to demonstrate the feasibility of our proposed solution. We provide learners with "atomic actions", that can be used to create and revise processes "on the fly", thus enabling double-loop learning.

1. Introduction

The challenge of the COOPER project (Collaborative Open Environment for Project Centered Learning) [1] is to provide virtual teams, whose members are geographically dispersed and have different backgrounds and competencies, with an environment in which to work *and* learn together in projects to solve complex problems. An important design aspect of our environment is the *portability* of the COOPER environment, including the pedagogical designs, to other (educational) institutions. In this paper we will focus on the implementation of the Virtual Company design [2, 3]. The COOPER-environment is a web-based working and learning environment created with Webratio [4], a WebML [5] and Business Process Modeling Notation (BPMN) [6] tool. The COOPER environment is created in *design-time* while users work in *run-time*. To support double loop learning [7] we will propose our solution of "atomic actions" that can be used as independent building blocks with which students can build, modify or re-arrange work processes in *run-time*.

2. Instructional design in e-learning systems

Supporting process oriented collaborative work in instructional design based e-learning systems is an important challenge today. Instructional design implemented in e-learning systems is often developed in analogy with workflow systems. Workflow management systems might seem very suitable candidates for modeling these teamwork processes. However, most of the resulting applications are characterized by predictability and repetitiveness. Furthermore, the application of workflow systems in domains that show highly dynamic processes e.g., solving of ill-structured problems, is still largely unexplored. These dynamic processes can hardly be completely predefined and/or exhibit an explosive number of alternatives, thus escaping the ability to being fully modeled. [8]

Besides that, these processes should be adaptable to support double loop learning. Unfortunately, adaptations in *run-time* still seem to pose an unsolvable problem for almost all existing workflow based e-learning systems. For usually, process changes can only be accomplished by modifying a corresponding workflow schema in *design-time*. However, as argued, it is important that such changes can be conducted inside the *run-time* instance, without causing inconsistencies and errors because of unfulfilled dependencies between process steps. [9]

3. Implementing double-loop learning

The notion of double-loop learning stems from organizational management theory. In double loop learning a student evaluates fundamental aspects of the organization [10] and initiates changes, like work processes and the definition of organizational roles. To enable double loop learning, we need to provide an environment that supports the evaluation of working and learning processes of individuals, their team and the organization in which they operate. The results of these evaluations need to be fed back into the different levels of the organization, changing work processes and organization aspects. The Virtual Company (VC) educational design implements these requirements in that students have an organizational role to fulfill in the processes in the virtual company. In that role, they perform ill-structured tasks from real clients in a real, but virtualized, company in

[*] Corresponding author: email: howard.spoelstra@ou.nl, Phone: +31 455762391

order to expand their (collective) expertise in a professional setting. In doing so, they gain expertise in the form of personal learning, team learning, organizational learning, knowledge management and the development of organizational competencies. [3, 11]

In the VC design, we discern three double loop learning cycles: the personal development cycle, the team development cycle and the company development cycle. The personal and team development cycles are depicted in Figure 1. (A similar development cycle is available for the organizational level.)

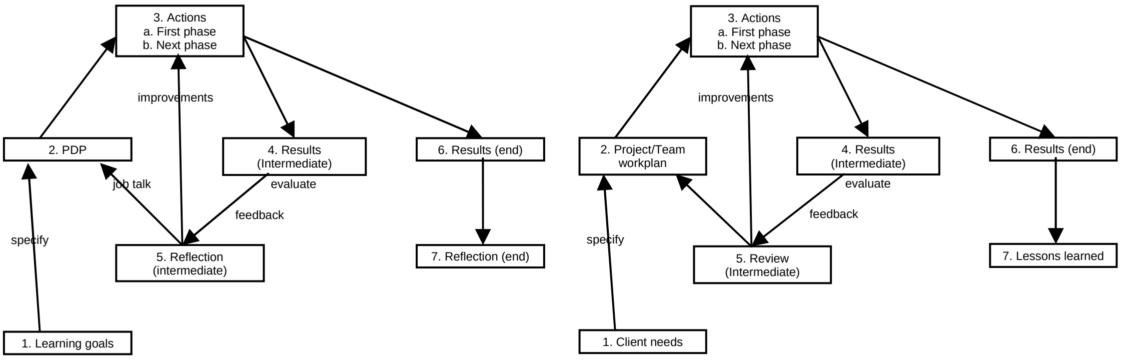


Figure 1: The personal and team development cycles in the Virtual Company design

Activities depicted in figure 1 are:

1. Learning goals are defined, also based on possible learning opportunities in the client needs
2. The personal development plan is made, in conjunction with the team work plan
3. Actions are performed, based on the initial personal development plan and team project plan.
4. Results are delivered to the VC and the customer.
5. Reflection on both personal development and team development can lead to adjustments in the personal development and team work plan, thus changing the actions to be performed.
6. The project end result is delivered.
7. Final personal and team reflections on product and process are performed; lessons learned are made available for other project teams and the company to reflect on.

It is in the cycle of steps 2, 3, 4 and 5 that double-loop learning takes place and students can decide to adjust e.g., personal development plan, project plans or assessment criteria, thus leading to changes in their working and learning environment.

To facilitate these adjustments in work flow-like e-learning environments, there should be a limited dependency between activities defined in the working process. The COOPER environment provides the learner with the means to define and change working processes without some of the caveats, pointed out above, other (educational) workflow systems exhibit.

4. The Cooper environment

The Cooper environment is developed using a model-driven, easily extensible development environment, which is based on the adoption of a conceptual model for web application design, WebML (Web Modeling Language), and Webratio, a tool that assists the WebML-based visual modeling of applications and enables the automatic generation of code starting from the visual schemas generated during design. Recently, WebML and Webratio have been extended and now can also cope with the design of *static workflows* [12], i.e., processes that are specified at design-time, and are then delivered to the enrolled users by means of a Web application supporting the execution of the planned process activities. Such extension in particular allows designers to specify visually workflows at a high level of abstraction by using the BPMN notation. It also provides a set of model transformations from BPMN workflow diagrams to WebML hypertext diagrams that allow fast generation of site skeletons implementing the specified business process [13].

However, static workflows cannot fully support the Virtual Company educational design, in which students need to be presented with a collaboration environment that is adaptable in run-time. In fact, once the application supporting the static workflow execution is produced and deployed, it becomes difficult (or even impossible) to modify the process. Therefore, we envisioned a more flexible mechanism, to allow students to define and/or adapt at *run-time* their *dynamic cooperation processes*. First we analyzed project development for reoccurring activities. Secondly, from these reoccurring activities, we defined the notion of *Atomic Actions* to guide the development of an Atomic Actions library. Atomic Actions:

1. are performed on a regular basis
2. may involve individual or group activities and may be started by an individual or a group
3. have a clear starting and ending point, serving a (very) small goal in the project process
4. use one or more of the services that are integrated in the COOPER platform
5. can easily be composed into dynamic processes supporting the completion of cooperative tasks involving several actors

The Atomic Actions library consists of atomic actions directly aimed at the process of running the project and of atomic actions that support the communication processes used to collaborate virtually.
They may be seen as (very) small pieces of workflow that can be "stitched" together at will, while retaining the changeability of the so constructed process.
The Atomic Actions are then modeled in the environment. This enables students, when analyzing the tasks in their project, to use these "Atomic Actions" as building blocks to model their own working and communication processes, and change these if reflection on the processes so requires.

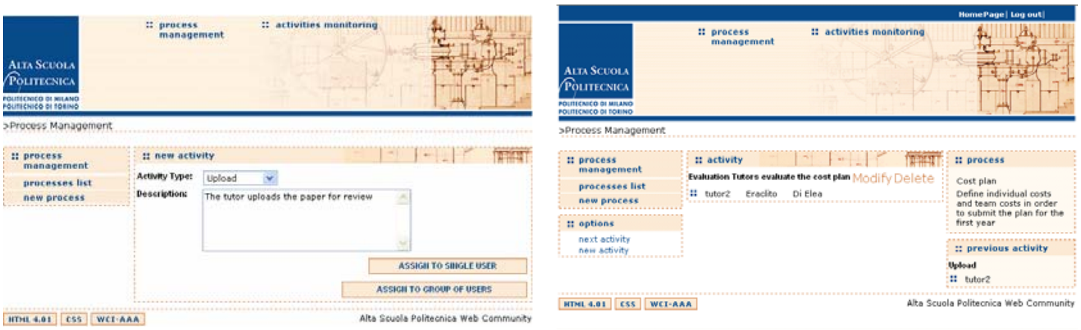
Although our work also involves modeling the entire company environment, our current list of atomic actions consists of actions to be used only in the project development cycle. A provisional list of these atomic actions is presented in Table 1.

Table 1: Atomic Actions classified by the collaborative activity they relate to.

Project Activities	**Atomic Actions**	
Define project method	Define a task	
	Assign a task	
	Define a project milestone	
	Define a deliverable	
Organize review	Create a review report	
	Assign a resource to (a) reviewer(s)	
	Submit a review on a resource	
Organize assessment	Define an assessment criterion	
	Define a performance indicator	
	Plan an assessment	
Manage resources	Upload a document	
	Publish a document	
	Route a document	
Project support activities		
Communication	Open a VoIP* meeting	Open a co-browsing activity
	Join a VoIP meeting	Moderate a co-browsing activity
	Make a VoIP call	Join a co-browsing activity
	Create a chat room	Open a forum
	Open a chat session	Open a forum thread
	Moderate a chat session	Moderate a forum thread
	Join a chat session	Send a forum thread message
	Define a voting question	Define a Wiki
	Submit a vote	Modify a Wiki page
	Summarize voting results	
	Open a video conference	
	Moderate a video conference	
	Join a videoconference	

*: Voice over IP

An example of how a working process is created and can be changed is shown below:



Activity Type: Upload

Description: The tutor uploads the paper for review

Evaluation Tutors evaluate the cost plan

tutor2 Eraclito Di Elea

Cost plan

Define individual costs and team costs in order to submit the plan for the first year

Upload

tutor2

Figs 2a and 2b. Student defines and alters processes

In the COOPER environment, a working process can be considered a sequence of phases, in which each phase is limited by some synchronization points that establish possible constraints for controlling the process evolution. The definition of a working process therefore proceeds in phases, and for each phase the selection of one or more atomic actions is required (corresponding to the phase activities)

Figure 2a shows the page where a student in *run-time* defines a process by selection an atomic action (e.g., upload of a document) to be included in a process phase, and describes the purpose of such activity by entering a short textual description. The activity is then assigned to the team member(s) that should accomplish it. Activities can be assigned to single users or to a group of users. In the last case, the activity definition also requires the student to specify the type of parallelism governing the execution of the parallel activity. It is possible to choose whether all the team members are asked to execute the activity, or whether at least one of them should execute it. Finally, the definition of a single activity may also require the association of resources - if any - for managing possible document flows, as it often occurs in cooperation processes. After a process has been defined, it is possible to revise its definition, as is shown in figure 2b, by modifying or deleting its activities or the assignment of activities to users. If the process is not running, any activity can be modified. Once the process is running, modifications are only allowed on the activities not yet started.
The Cooper environment also makes it possible to create process templates, i.e., process models, which are independent from specific project teams and specific actors that are executing the process activities. The template only defines the temporal sequence of activities and possible activity synchronization constraints, omitting the assignment of activities to users. Once defined, these templates can be used for starting new processes that only require the addition of actors for each activity in the template.
This enables us to present students either with a “clean slate” on which to model and adapt their own processes, or with a predefined set of templates of activities. However, both offerings retain the flexibility to make “on the fly” changes, thus bridging the gap between ID and double loop learning.

5. Discussion

Instructional design as implemented in work flow systems and double loop learning can be brought together using Atomic Actions. The notion of Atomic Actions is not limited to the COOPER environment alone. An implementation in e.g., IMS LD is also to be considered. As Atomic Actions do not yet completely free double loop learners from work flow dependencies, further analysis of these dependencies and ways to avoid them is required. A broader list of Atomic Actions and the modelling of more portable educational scenarios in the Cooper environment are also subject to further research.

Acknowledgements: This article is sponsored by the COOPER Project that is funded by the European Commission's 6th Framework Programme, priority 2 IST. Contract no.: 027073 (www.cooper-project.org)
The authors are further indebted to Dr. Peter B. Sloep for his kind comments.

References

[1] The COOPER project: http://www.cooper-project.org
[2] M. Bitter, P. Sloep and D. Jansen (2003) Learning to change: The Virtual Business Learning approach to professional workplace learning. Educational Technology & Society 6 (1), 2003.
[3] W. Westera and P.B. Sloep (1998) The Virtual Company: Towards a Self-Directed, Competence-Based Learning Environment. Educational Technology, the magazine for managers of change in education, 38 (1), 32-38.
[4] Webratio: http://www.webratio.com/page18.do
[5] WebML: http://www.webml.org/webml/page1.do
[6] BPMI. Business Process Management Language, http://www.bpmi.org.

[7] C. Argyris and D.A. Schön (1996). Organizational learning II: theory, method, and practice, Reading, MA: Addison Wesley.

[8] P. Mangan and S. Sadiq. On Building Workflow Models for Flexible Processes. ACM International Conference Proceeding Series; Vol. 18, 2002.

[9] S. Rinderle, M. Reichert and P. Dadam. Flexible Support of Team Processes by Adaptive Workflow Systems. Distributed and Parallel Databases Volume 16, Number 1 / July, 2004 pp. 91-116

[10] C. Argyris and D.A. Schön (1974). Theory in Practice, San Francisco: Jossey-Bass.

[11]W. Westera, P.B. Sloep and J. Gerrissen (2000). The Design of the Virtual Company; Synergism of Learning and Working in a Networked Environment. Innovations in Education and Training International 37 (1), 24-33.

[12] M. Brambilla, S. Ceri, P. Fraternali, I. Manolescu. Process Modeling in Web Applications. ACM TOSEM (to appear).

[13] M. Brambilla. Generation of WebML Web Application Models from Business Process Specifications.
Demo at the 6th International Conference on Web Engineering (ICWE2006), July 2006, Palo Alto, CA, USA.