

Tele-education Process Modelling supported by The ODP Enterprise Viewpoint Language

Cees Volman, Val Jones, Marten van Sinderen, Ing Widya, Eddie Michiels
Centre for Telematics and Information Technology
University of Twente, PO Box 217, 7500 AE Enschede, The Netherlands
{volman, jones, sinderen, widya, michiels}@cs.utwente.nl

Abstract

This paper reports on applying the ODP enterprise viewpoint in the domain of tele-education. The work is conducted as part of a research activity¹ that aims at designing a tele-education system to support planning, execution and evaluation of dynamic distributed educational processes. We explore the ODP enterprise viewpoint as a basis for communication and co-operation between educational scientists and ODP systems designers involved in the design process. Our application of the enterprise viewpoint involves four main steps. First, an educational language is proposed to describe educational processes in generic terms. Second, a set of appropriate enterprise language concepts is selected. Third, a relationship is established between the educational language and enterprise language concepts. Fourth, an educational process is modelled in terms of an entity-oriented model and a behaviour-oriented model. It is hoped that the experience gained in this exercise will provide useful feedback to both the educational and ODP communities.

Keywords: ODP enterprise viewpoint, tele-education, didactic models, telematics systems, telematics infrastructure, modelling concepts.

1 Introduction

Educational theorists and practitioners are increasingly exploring the opportunities offered by advanced information and communication technology (ICT) to realise tele-education courses. Tele-education courses involve educational actors communicating with each other and utilising information resources to accomplish goals (see e.g. [2,3,4,11,13]) while eliminating the need for

the actors or resources to be present at the same time and/or place.

In tele-education, educational processes² may range from ‘closed’ to ‘open’ mode processes. Closed processes are based on plans that are pre-determined by the teacher and the plans are not expected to be changed while a process is in progress. Closed mode processes are applied for rote learning, practice and drill or tutorial learning for example [13]. Open processes, in contrast, are pre-planned only to some extent by the teacher, and students possibly together with the teacher are expected to plan the process further while it is in progress. Open mode processes are usually applied in cases of learning-to-learn, problem-based learning or project-based education [1,5,13]. The two process modes can also occur in combination in educational processes.

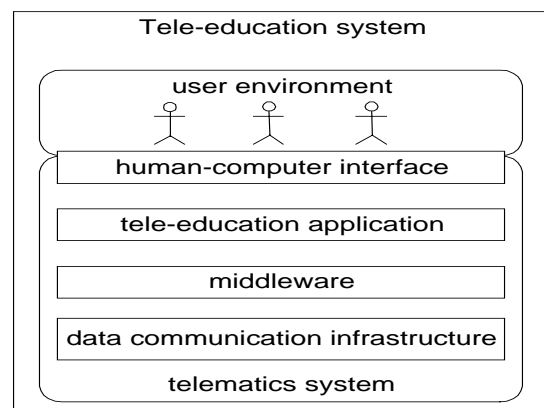


Figure 1: A Tele-Education System

In our view, tele-education is promoted by a tele-education system that enables flexible planning, execution and evaluation of distributed, educational processes of both closed and open modes. A tele-education system comprises a user environment interacting with a telematics

¹ This is in turn part of a broader ‘Tele-learning’ program, a multi-disciplinary project of the Centre of Telematics and Information Technology, which is supported by a research grant from the University of Twente.

² Educational processes are interpreted as patterns of educational activities.

system (ODP system). The telematics system involves an embedded architecture comprising of a data communication infrastructure, middleware, a tele-education application and a human-computer interface. The tele-education system is depicted in Figure 1.

The design of a telematics system involves, amongst others, collaboration between the disciplines of educational science and computer science, therefore establishing a common understanding across the two domains is crucial. Computer scientists need to fully understand the requirements of the users from the education domain; the educational scientists, on the other hand, need to be aware of the kinds of support which can be expected from computer based distributed systems. A design approach, with appropriate (sets of) modelling concepts which spans the two disciplines, is needed to bridge the gap between the two disciplines.

In this paper we investigate whether the design of this tele-matics system benefits from the ODP approach. We focus on the modelling of tele-education processes in terms of roles and activities which are identified as being essential from a *didactic* point of view, and, in accordance with the ODP approach, we start with the enterprise viewpoint. A relatively simple example from tele-education practice is used to show the application of the enterprise language concepts. The example is simplified, but representative enough to serve two purposes. First, to discover whether enterprise viewpoint modelling gives leverage on (some of) the concepts and problems of the tele-education domain; and second, to discover whether the enterprise language concepts are sufficient and appropriate when applied in specific application domain of tele-education.

The remainder of the paper is organised as follows. Section 2 briefly describes the relevance of the ODP-RM to designing a telematics system. Section 3 introduces selected concepts from the tele-education domain. Section 4 introduces the ODP enterprise language concepts which are considered necessary in the tele-education domain. Section 5 presents the mapping of the tele-education language concepts onto the ODP enterprise language concepts. These concepts are then used in Section 6 to describe an educational process involving both a closed and an open mode sub-process. Based on this example, Section 7 evaluates the applicability of the ODP enterprise viewpoint to the tele-education domain. Section 8, finally, indicates some directions for future work.

2 Relevance of the ODP-RM

Since ICT plays an important role in accomplishing successful tele-education practice, the tele-matics system should be designed in such a way as to empower and enable educational practitioners to work with the full

complexity and flexibility of educational practice. We investigate how the ODP-RM provides a framework which permits such a design approach. Underlying the ODP framework is the recognition that different abstractions (viewpoints) and multiple levels of detail are necessary to control the complexity of the overall design process [15]. In this paper, we focus on the enterprise viewpoint but here we indicate the relationship between ODP viewpoints and the different elements of the tele-education system.

2.1 The design of tele -education processes

Figure 2 depicts a tele-education system in compliance with the ODP Viewpoints.

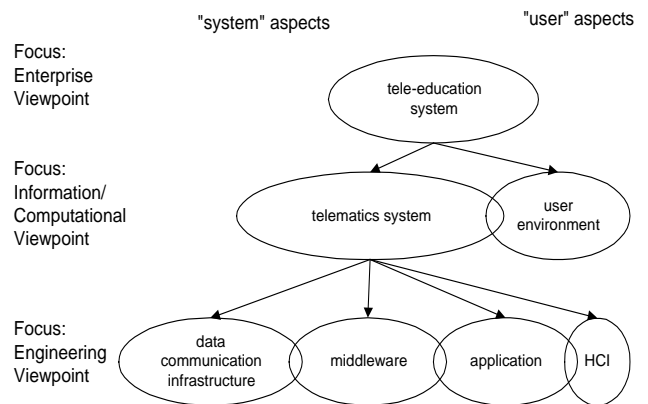


Figure 2: Tele-education system

From the enterprise viewpoint, the telematics system is designed in terms of the roles, policies and activities associated with the enterprise which are identified as being essential from educational or didactic point of view. In this design, no distinction is made between 'system' roles and activities (which can be automated) and the 'user' roles and activities (which have to be performed manually).

From the information/computational viewpoint, the tele-education process is considered as a user environment to be supported by a telematics system. The user environment model is concerned with tele-education activities assigned to the individual human users or organisational entities with access to the telematics system. The telematics system model is concerned with the (tele-education) services provided to the users while it abstracts away from the internal structure of the telematics system.

From the engineering viewpoint, an internal structure for the telematics system is introduced. The internal structure may consist, for example, of a standard communication service, several standard application protocols and application protocols which are specifically designed to provide the required tele-education services,

the tele-application and a human-computer interface to support the abstract user-system interactions also represented in the information/computational model.

In the sequel of this paper we will investigate the use of the ODP enterprise viewpoint as a first step in the design of the telematics system. The enterprise viewpoint “focuses on the expression of the purpose of an open distributed system” [8]. In order to represent a design from an enterprise viewpoint, it is necessary to “define a structured set of concepts in terms of which the representation (or specification) can be expressed” [9]. This ‘structured set of concepts’ is called the enterprise language.

2.2 Application of modelling concepts

Since the enterprise viewpoint is independent of any specific application domain, the concepts of the enterprise language are necessarily general purpose (with respect to different application domains). One can imagine that a specific application domain requires specific concepts that permit a more direct representation of design aspects in that domain, and facilitate communication of the design amongst people working in that domain.

Figure 3 depicts the relationship between tele-education language concepts and the enterprise language concepts. It also shows the distinction between modelling (architectural or design) concepts and their representation. Modelling concepts are the basic building blocks with which a model (design) can be expressed. A model can only be communicated if it has some representation.

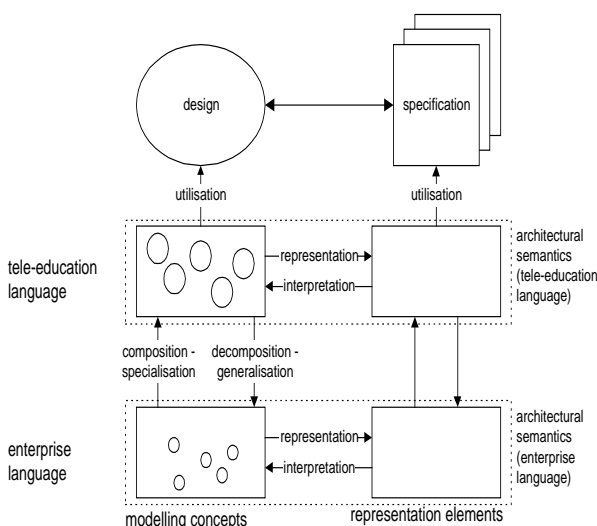


Figure 3: Relationship between tele-education language concepts and the enterprise language concepts

Hence, each modelling concept must have a representation; for this, representation elements are used;

these are the basic building blocks of a specification language. In principle, any specification language may be used to represent the same set of modelling concepts. In practice, however, some specification languages may be more suitable than others, since each language has limited expressive power as determined by its syntax and semantics. The relationship between modelling concepts and their representation in terms of specification language elements determines an architectural semantics.

The definition of an architectural semantics for the enterprise language is one of the work items carried out in the context of ODP-RM [10]. Here we try to establish appropriate mappings of tele-education language concepts onto enterprise language concepts with the ultimate goal of establishing an architectural semantics for the tele-education language via these mappings and the architectural semantics defined for the enterprise language.

3 The tele-education language concepts

To establish a mapping between educational and ODP enterprise language concepts, as discussed in Section 2, we obviously need a set of educational language concepts. The discipline of education does not currently have an agreed standard language for describing educational processes. In this section, we suggest a simple educational language that supports the description of educational processes.

In educational practice, educational processes are realised within a didactic organisation that can be described in terms of *didactic model components*. Typical components shared by didactic models are ‘educational objectives’, ‘entry conditions’, ‘educational activities’, ‘learning materials’, ‘media and tools’, ‘group schedules’, ‘time schedules’, and ‘outcomes’ (see e.g. [1,2,12,13]). For this reason, we propose to use these terms as our initial set of educational language concepts. This gives us the ‘what’ of educational processes.

As for *who* is involved in educational processes, we can immediately arrive at the concepts teachers and students. We expect from teachers that they control and manage the educational process. In open mode processes, however, teachers may also put students in the position of controlling and managing educational processes in which they are involved [1,5,13].

Educational processes are subsumed to plan-do-evaluate cycles [1,13]. The activities related to the stages of the cycle are considered *meta-activities* in contrast to activities that constitute an educational process. The educational activity ‘write a program’ for example can be planned, performed or evaluated. In the ‘plan’ stage, a didactic model guides the specification of the content for each of the components in the model. In the ‘do’ stage, the

didactic model guides teachers and students working in compliance with a plan. In the 'evaluate' stage, finally, a didactic model supports teachers or students in comparing the actual outcomes of the educational process with the planned outcomes in order to establish the quality of an educational process.

To discriminate closed from open mode processes, we finally need to indicate *when* (components of) a plan-do-evaluate cycle can occur. In a closed mode process,

planning activities take place before an educational process occurs. In an open mode process, however, planning activities may also occur once the educational process has started. Taking the participation of students in the educational process as the point of reference, we propose the distinction between 'before education', 'during education' and 'after education' to clarify this issue.

Table 1: A set of educational language concepts

Educational Concepts	Meaning
DM components:	
Educational objectives	Expected outcome of, or deliverable produced by, an educational process
Entry conditions	Prerequisite skills and knowledge of educational consumers
Activities	Educational activities facilitating achievement of educational objectives
Learning content	non-human, passive information resources
Group schedules	configuration of educational actors
Time schedules	periods in which (teaching and) learning activities are expected to occur
Activity schedules	ordering relationships between educational activities
Media & Tools	non-human resources for production, storage, retrieval or transfer of information
Outcomes	tangible resources resulting from educational activities
Who:	
Teacher	principal educational provider
Student	educational consumer
Meta-activity:	
(Re-)plan	(re-)establishing a plan for an educational process
Execute	executing an educational process according to a plan
Evaluate	evaluation of educational process and process results during or after its completion
When:	
During education	phase of education during which students may be involved
Before education	phase preceding education during which students may be involved
After education	phase succeeding education during which students may be involved

Table 2: A description of a closed and open mode educational process

Closed educational process	Open educational process
<i>Before education, the teacher plans the complete educational process in terms of the educational components.</i>	<i>Before education, the teacher plans parts of the educational process.</i>
<i>During education, students possibly together with the teacher perform activities in compliance with the plan.</i>	<i>During education, students possibly together with the teacher plan the remaining parts of the educational process, and execute activities in compliance with the plan. During education, students possibly together with the teacher may evaluate the execution of the educational process and they may also re-plan the process.</i>
<i>After education, the teacher evaluates the results of the educational process and the teacher may re-plan the educational process.</i>	<i>After education, the teacher evaluates the results of executing the plan and may re-plan the process.</i>

In Table 1 we have summarised the set of educational language concepts, together with an informal description of the concepts. The concepts are applied according to a rule as expressed by the question: *when* (e.g. before

education, during education, after education) does *who* (teachers and students) perform *which meta-activity* (plan,

execute, evaluate)³ with respect to *which DM component(s)*. The question will be asked repeatedly during the specification of a complete educational process, i.e. for each of the sub-processes involved.

With the educational language concepts and the rule proposed we can describe two extreme modes of operation of educational processes, namely the closed and open mode. The two modes are characterised in Table 2.

In real education, both closed and open types of processes may occur as parts of a larger process. In Figure 4, this is illustrated by using a flowchart notation commonly applied in educational systems design (see e.g. [13,14]).

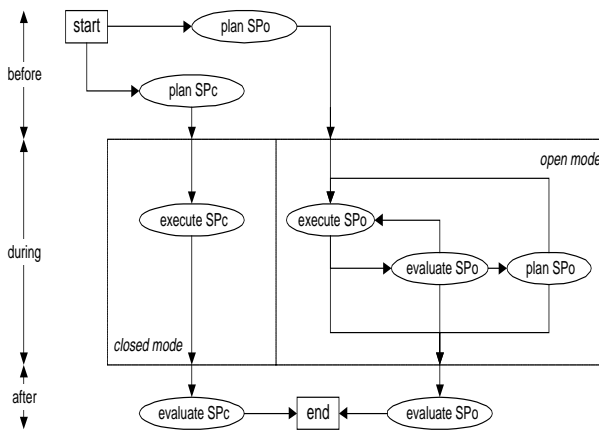


Figure 4: An educational process sharing a combination of open and closed modes

In Figure 4 EP, SP_c and SP_o denote a complete educational process comprising of a closed mode sub-process and an open mode sub-process, respectively. The labels before, during, and after denote the stages *relative* to the students being allowed to participate in the EP.

The sub-process SP_o is an example of a self-organising or self-referencing process [6]. That is, **execute** a plan may involve obeying directives to **evaluate** or **re-plan** the educational process while it is in progress. The directives to **evaluate** or **re-plan** take the form of specifications for the activity component in the didactic model. We believe that it is important to exercise the ODP methodology on self-modifying processes since any realistic educational model will involve some degree of self-modification in the during stage of an educational process [5,13].

³ The activities (re-)plan, execute and evaluate are set a different font to indicate that these activities are meta-activities compared to activities as specified in the activity component of the didactic model.

4 The ODP enterprise language concepts

The enterprise language is defined in Part 3 of the ODP-RM [9]. In addition to the concepts used from ODP-RM Part 2 [8], two language concepts are defined that explicitly relate to the enterprise viewpoint, namely community and federation⁴. From this set of concepts, we selected those that seemed appropriate in order to describe the educational 'enterprise'. The resulting set of enterprise language concepts is presented in Table 3. To indicate the meaning of the selected concepts, we have paraphrased most of the ODP definitions as stated in Part 2 and Part 3 of the ODP-RM [8,9].

Table 3: Enterprise Language concepts

ODP-RM Concepts	Meaning
Environment	The part of the model which is not part of the object under consideration
Contract	Agreement that governs activities in which multiple objects participate
Action	Something which happens
Location in time/space	Attribute of an action, constraining the time/place of the action
Activity	Set of related actions
Role	Activity pattern
Object	An entity; a carrier of activity
Group	Set of objects sharing a characteristic relationship
Epoch	A time period in which an object shows behaviour

5 A mapping between tele-education and ODP concepts

To be able to model educational practices from the enterprise viewpoint we first need to establish the relation between the educational concepts presented in Table 1 and the enterprise language concepts of Table 3. Table 4 presents relationship between the educational concepts and the enterprise language concepts as indicated in Fig. 2.2.

⁴ In this paper, we only pursue the concept of community for reasons of simplicity. The concept of federation would be relevant, however, if for example students are collaboratively working on information resources but where the results are finally evaluated and graded in the context of different curricula by teachers from different faculties.

Table 4: Relation between educational and enterprise language concepts

Educational concepts	ODP-RM concepts	Educational concepts	ODP-RM concepts
Before education	Epoch	Educational objectives	Contract
During education	Epoch	Entry conditions	Policy
After education	Epoch	Educational activity	Action/Activity
Teacher	Role	Learning contents	Object
Student	Role	Group	Group
Planner	Role	Time dependencies	Location in time
Executor	Role	Activity dependencies	Policy / Contract
Evaluator	Role	Media & Tools	Object
(Re-)Plan	Action/Activity		
Execute	Action/Activity		
Evaluate	Action/Activity		

6 An example of modelling a tele-education process

We present the modelling of a simplified educational process as an example to illustrate the essentials of the design and specification of complex educational processes and to study the applicability of the ODP enterprise viewpoint within the tele-education domain. The example is based on a course on Courseware Engineering and the Human Computer Interface (HCI) taught at the University of Twente. The example educational process (EP) consists of a closed mode sub-process (SP_c) and an open mode sub-process (SP_o) as depicted in Figure 4. The following steps are taken to present the example:

1. The example is introduced by informally describing the scenario for SP_o (Section 6.1).
2. The objects and their shared activities are identified, leading to a first model in terms of ODP enterprise language concepts (Section 6.2).
3. The actions and their (temporal) relationships are identified corresponding to the activities of step 2, leading to a second model in terms of ODP enterprise language concepts (Section 6.3).
4. It will be shown that SP_c is a special case of SP_o .

The first model emphasises the entity aspects whereas the second model focuses on the behaviour aspects of the educational process. The entity-oriented model is needed to explicitly describe the structural aspects of educational processes. The behaviour-oriented model, however, gives leverage on the temporal ordering of the activities involved in education. We consider the two representations to be complementary in the process of designing a tele-education system. In both cases we adopt an ad-hoc graphical notation, which is hopefully intuitive

and needs little explanation to provide comprehensive representations.

6.1 The educational scenario for an educational process EP

The educational process in *closed* mode, sub-process SP_c is fully planned by a teacher before it is started. In our example, the educational objective is that students have to make a plan for the design and implementation of a Human Computer Interface (HCI). The entry condition for the students is undefined. According to the activity specification in the SP_c -plan, students have to make a plan for the development trajectory leading to realisation of an HCI prototype. The teacher evaluates the plan. Based on the evaluation of the teacher, the students either have to adjust the plan until it meets the evaluation criteria or they may proceed with the next activity: making a flow chart for the HCI. Again the teacher evaluates the results. If satisfactory, the students may start implementing the HCI, otherwise they have to adjust the design for the HCI. The students will work on the prototype until the teacher gives a positive assessment of the prototype. After an indicated time period, the teacher must evaluate SP_c by examining how well SP_c has led to the accomplishment of the educational objective.

The educational process in *open* mode, sub-process SP_o , is sparsely planned by a teacher before it is started. The educational objective is that students will be able to plan, execute and evaluate projects, e.g. concerning a field test for a HCI. The entry condition for the students is defined in terms of having passed a sub-process SP_c , i.e. having developed an HCI prototype. According to the activity specification in the SP_o -plan, students have to plan the educational process by further instantiating educational components for SP_o . The instantiation should state how students go about to realise a project concerning a field test for a HCI. In addition, the students will

execute their plan, meaning that students accomplish the activities they have specified in their plan. Finally, they have to evaluate their plans, i.e. establish whether they have learned from planning, executing and evaluating a project plan. If students decide to (re-)plan their project, the students are in fact (iteratively) modifying the SP_o plan. The teacher may participate in the activities performed by the students. After an indicated time period, the teacher must evaluate SP_o by examining how well SP_o has led to the accomplishment of the educational objective.

6.2 The entity aspects of EP: the identification of objects and shared activities

The objects involved in SP_o are the ‘environment’ (e.g. a university), the teacher, the students, an educational plan, electronic information resources, tools and tangible results as process outputs (see also [3,10,14]). The activities between objects in SP_o are plan, execute, evaluate, and the activities as specified in the educational plan. (Since we are interested in the open versus closed mode of educational processes, the object ‘environment’ abstracts from entities in the course environment. All activities are governed by explicit or implicit contracts, for example the process plan.

The objects and their shared activities are depicted in Figure 5. In Figure 5, the following conventions are used: objects and contracts are represented by different shapes, activities between objects are represented by bi-directional arrows, constraints on activities imposed by contracts are represented by connecting lines, and the establishment of contracts by objects is represented by dotted arrows.

The objects *teacher* and *student* represent the human actors in the educational process who assume particular roles. The object labelled *plan* indicates an educational plan (for SP_c or SP_o). The object labelled *~plan* refers to all other resources (e.g. programming environment, word processor and flow charter). The object labelled *results* indicates the tangible learning outcomes available during or after execution of the course. The activities plan, execute and evaluate are ruled by contracts. The contract *open or closed* indicates that a teacher may choose to implement a closed or open mode course plan. The contract *before* indicates that the teacher has to provide an initial plan before the course is started. The contract *after* indicates that the teacher is obliged to evaluate the course after completion of the course. The contract *iff EP = open* indicates the disabling or enabling of plan and evaluate activities by students in case of a ‘closed’ or ‘open’ course mode respectively. The contract *during* is enforced while the teacher performs the plan activity to set up a *plan* for an educational process. This

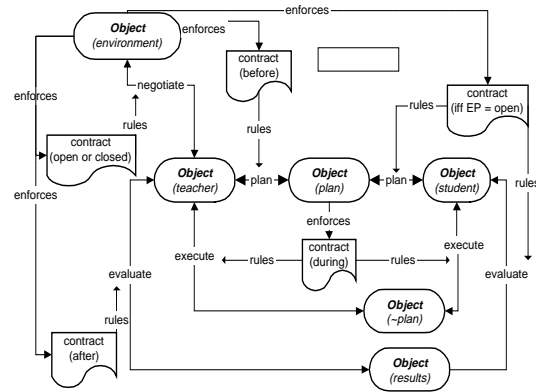


Figure 5: The educational process

contract actually rules the activities of the objects during the activity execute.

In terms of the model presented in Figure 6.1, the difference between SP_c and SP_o follows from the educational plan as embodied by the object *plan*. The activities specified in the plan form the directives characterising the process as an open or closed mode process. This, in turn, determines the contents of the contract *during*, which rules the activity execute, and the applicability of the contract *iff EP = open*, which rules the activities plan and evaluate.

Entity oriented models like the one in Figure 5 are used to refine the model presented in Figure 4 by emphasising the structural aspects of educational processes. The structural aspects involve the dimensions of Table 1. In Figure 5 the dimension ‘DM components’ is reflected by the objects *plan*, *~plan* and *results*. The ‘who’ dimension is represented by the objects *teacher* and *student*. The ‘meta-activities’ are shown as the arrows labelled plan, execute and evaluate. The ‘when’ dimension, finally, is shown as the contracts before, during and after. Note also that the dynamics of the educational process shown in Figure 4 become less visible.

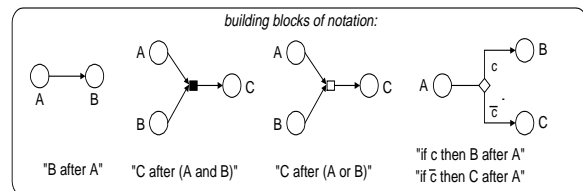
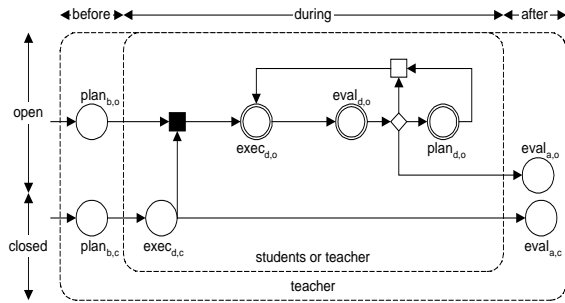


Figure 6: Basic building blocks of behaviour notation

6.3 The behaviour aspects of EP: Actions and their relationships

To describe the behaviour aspects of educational processes, we use the notation as presented in Figure 6. The notation used is based upon [16]; actions are represented by circles, and relationships (causality relations) between actions are represented by arrows. The composition in terms of actions of the main activities identified in the previous section, i.e. plan, evaluate and execute (see Figure 7), depends on whether the ‘open’ or ‘closed’ mode applies. We will first describe the ‘open’ mode SP_o process. Then we will show that the ‘closed’ process SP_c forms a special case of the open mode process.



b = before, d = during, a = after, o = open, c = closed
Figure 7: Actions and relationships between actions in the educational process EP

In Figure 7 we have used the following additional conventions. The action subscripts indicate a particular phase and mode, respectively. The notation $plan_{b,o}$ means that an open sub-process (o) is planned before (b) it has actually started for example. The doubled circles indicate that an action may be invoked at subsequent points in time. The grouping of actions according to the educational phases is represented by the double headed arrows labelled before, during, and after. The double headed arrows labelled open and closed group the actions according to the open or closed mode process. The double headed arrows, marking different domains of specification, are not part of the notation; they are added as annotations to support the accompanying text. Grouped actions in the rounded rectangle labelled teacher means that only the teacher can be involved in those actions. Grouped actions in the rounded rectangle labelled students or teacher means that the students or the teacher can be involved in those actions.

Based on Figure 7 we can now describe the difference between a ‘closed’ and open mode sub-process. The closed mode process SP_c is considered a special case of the of the open mode process. In a closed mode process, only one action can be invoked, namely $exec_{d,c}$. In case of

the scenario for SP_c as presented in Section 6.1, $exec_{d,c}$ can still involve planning, execution and evaluation activities closed mode process SP_c for setting up a plan for the design and implementation and documentation of an HCI but these steps are fully pre-specified by a teacher before SP_c has started. Non of the planning, execution or evaluation activities in the plan refer to the plan for the SP_c itself.

In the next section we will illustrate that the behaviour notation is also suitable to model the actions and the relations between the actions by looking into the level of executing the SP_c .

6.4 Zooming in on the behaviour aspects of SP_c : Actions and their relationships

In Figure 8, we have used the same notation as in Figure 7. The difference is that the activities mentioned in the SP_c scenario as presented in Section 6.1 are added

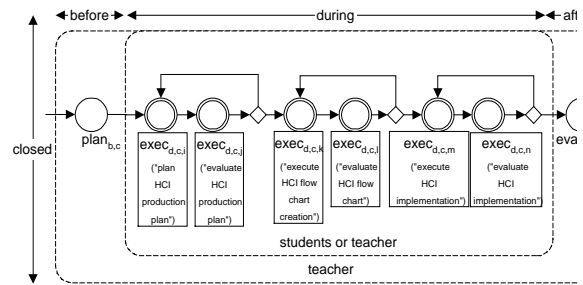


Figure 8: Actions and action relationships within the closed mode process SP_c

between brackets. The use of brackets is not part of the notation, but is intended to easily establish the conceptual relationship with Figure 7.

Note that the activities plan, execute and evaluate between the brackets are not the same as the activities $plan_{b,c}$, $exec_{d,c}$, $eval_{d,c}$ and $eval_{a,c}$, since the former do not refer to the plan for SP_c . If this would be the case, then the process in Figure 8 would be an open mode process. This is this an important difference in tele-education processes that we want to carefully address in the design of telematics systems, together with the self-regulation mechanism present in open mode processes.

Behaviour oriented models, like the one shown in Figure 7 and Figure 8 are used to refine the model presented in Figure 4 in the sense that the behaviour oriented model allows for a more detailed and accurate description of transitions between activities in the educational process. Note, however, that in the model displayed in Figure 4 and in the behaviour oriented model

of Figure 7 and Figure 8, the structural aspects of the educational process are less visible.

7 Evaluation

The models presented in Section 6 were primarily used to illustrate the essentials of the design and specification of tele-education processes in compliance with the enterprise viewpoint. As such, the models represent organizational aspects of tele-education processes in such a way that it becomes clear how educational actors can plan, execute, and evaluate arbitrary processes that aim at the construction and transfer of knowledge and skills. The main purpose of the models is thus to provide a high-level specification which helps to understand a possible logical structure and dynamics underlying tele-education processes. In the sequel of this section, the resulting model will be discussed and the potential use of the models will be indicated.

7.1 Discussion of the models

The application of the ODP enterprise viewpoint has shown that the enterprise language concepts can be applied to model both open and closed modes of educational processes. To fully capture structural and dynamic aspects of educational processes, *both* the entity and behaviour approaches described in Sections 6.2, 6.3 and 6.4 must be used. Both models together provide powerful means for a top-down approach in order to refine process models like the one presented in Figure 4 in a concise and manner. This is important since the same type of activities, i.e. planning, execution and evaluation, can occur at both an 'institute' level and a 'classroom' level to realise an unambiguous semantics for telematics systems design. Furthermore, standardised notations, as proposed in [16] and applied in Section 6.3 and Section 6.4 for example, would support an increase in common understanding across disciplines involved in designing telematics systems.

Modelling educational processes in compliance with the enterprise viewpoint has also revealed the need to discriminate between actors or entities and roles again. In general, educational scientists and educational practitioners are used to talking about persons, identified as teachers or students, who are involved in the educational process. The teachers and students are also implicitly associated with typical or traditional activity patterns. By linking activities to teachers and students as persons or actors, concepts like 'teacher' and 'student' or 'teaching activities' and 'learning activities' will become ambiguous, most notably in the case of modelling open mode processes. In open mode processes, students as

actors may perform activities that are traditionally associated with teachers, for example the planning and evaluation of education. Consequently, the concepts of teacher and student can no longer be used to uniquely identify different activity patterns. Such ambiguity hinders a concise and formal modelling of educational processes.

Open mode tele-education processes are typically self-organising processes that can evolve over time. Describing self-organisation in an unambiguous way is often difficult. The application of the ODP enterprise viewpoint has given leverage to this problem. First, by applying the ODP enterprise viewpoint, educational processes are described in an explicit way. This explicit description also enables the modelling of self-organisation in a way that reduces ambiguity. One way of modelling self-organisation is to proceed as follows. Meta-activities are considered as activities that have (components of) a didactic model as their object of interest, e.g. $\text{plan}(\text{DM}_{\text{activity}})$, $\text{exec}(\text{DM}_{\text{activity}})$, and $\text{eval}(\text{DM}_{\text{activity}})$. The meta-activities are discriminated from the activities that are specified in the activity component of an educational plan, e.g. $\text{DM}_{\text{activity}}$ 'study manual' or $\text{DM}_{\text{activity}}$ 'implement an HCI'. Self-organisation can then easily be modelled by accepting meta-activities as specifications for the activity component of the DM, e.g. $\text{exec}(\text{DM}_{\text{activity}} \text{ 'plan(EP)'}$).

Finally, the exercise has shown that modelling a tele-education system requires educational processes to be described rather explicitly. The communication between the education and ODP community could be enhanced if the educational domain had its own standardised 'educational reference model'. Such a reference model would also propel the modelling of the tele-education system in terms of the ODP enterprise viewpoint because the ODP-RM supports a formal way of establishing a relationship between the educational language and the ODP enterprise language concepts.

7.2 The potential use of the models

The models in Section 6 provide a starting point for design and specification of the tele-education process within the other ODP RM viewpoints (see Figure 2). Each of these viewpoints is concerned with the services of distributed systems (e.g. tele-education services) that must be provided. The information viewpoint is concerned with the information sources and destinations, and the flows of information between them. From the computational viewpoint, the application components and their interactions must be designed and specified. In the engineering viewpoint, the implementation of the services in terms of a distributed system configuration is to be considered.

The transformation of models from one viewpoint into models from other viewpoints is a topic of current

research. Concerning the model presented in subsection 6.3, some experience has been gathered regarding high-level specification in the area of business processes re-engineering [17]. This type of experience is considered useful input to this research.

The models presented in Section 6 can also be used for analysis. In the tele-education application domain, the analysis aims at answering the main question, namely whether educational actors are enabled to effectively and efficiently plan, execute, and evaluate arbitrary distributed educational processes. It is expected that the results of the analysis will cause the models presented in Section 6 to evolve into more refined models until they cover the user requirements of the target audience of the intended ODP system.

At present, analysis, as well as validation of correctness and verification of consistency of the models, must be done manually. No dedicated tools are currently available that support the design and specification from the enterprise viewpoint. Some alternatives are slowly emerging, however. First, tools are under development that support modelling and re-design of business processes at the enterprise viewpoint ([17]). It is likely that these tools can be modified and used for the purpose of designing processes in the domain of tele-education. A second option is to study the possibility of using existing tools in the area of object-oriented software engineering that are based on a clearly defined language (e.g. [18]).

8 Future work

In future work we intend to:

- study the relationship between tele-education and enterprise language concepts in more detail
- extend the work to other ODP viewpoints
- achieve designs that more closely approximate the reality of educational practice
- incorporate developments in the area of object-oriented modelling and architectural semantics in the ODP community aimed at formal specification of tele-education systems.

Acknowledgements

We would like to thank I. De Diana for his valuable contributions regarding the educational process models and R. Huis in 't Veld for the inspiring discussions on the use of the ODP framework.

References

- [1] Borsum, W., H-G. Posern, and K. Schittko, *Einführung in die Didaktik. Didaktische Modelle, Grundprobleme, Praxishilfen*. Urban und Schwarzenberg, München, 1982.
- [2] De Corte, E., C.T. Geerlings, N.A.J. Lagerweij, J.J. Peters, and R. Vandenberghe, *Beknopte didaxologie*, Groningen: Wolters-Noordhoff, Groningen, 1981.
- [3] I.P.F. De Diana, and T.N. White, "Towards an educational SuperInterface", *Journal of Computer Assisted Learning*, **10**, 1994, 93-103.
- [4] De Diana, I.P.F., P.W. Verhagen, and E. Heeren, Educational use of multimedia databases in networked learning environments, in: S. Dijkstra, F. Schott, N. Seel, and R. Tennyson (Eds.) *Instructional Design, Solving instructional design problems*. Chapter 6. Hillsdale: NJ: Lawrence Erlbaum, (In Press).
- [5] Frey, K., *Die Projekt Methode*, Beltz Verlag, Basel, 1995.
- [6] Gomez, P. *Modelle und Methoden des systemorientierten Managements*. Verlag Paul Haupt, Stuttgart, 1981.
- [7] ISO/IEC (1995) Open Distributed Processing - Reference Model: Part 1: Overview and Guide to Use, International Standard 10746-1.
- [8] ISO/IEC (1995) Open Distributed Processing - Reference Model: Part 2: Foundations, International Standard 10746-2.
- [9] ISO/IEC (1995) Open Distributed Processing - Reference Model: Part 3: Architecture, International Standard 10746-3.
- [10] ISO/IEC Open Distributed Processing - Reference Model: Part 4: Architectural Semantics, Draft International Standard 10746-4.
- [11] Ladhani A, I. De Diana, and I. Widya, "Combining Desktop tele-classroom and Distributed Multimedia Databases for Tele-tutoring", *First International Distributed Conference on High Performance Networking for Teleteaching (IDC '95)*, November 16-17, 1995, Madeira, Portugal.
- [12] Peters, O., *Die Didaktik des Fernstudiums. Erfahrungen und Diskussionsstand in Nationaler und Internationaler Sicht*. ZIFF-Papiere 100. Zentrales Institut fuer Fernstudienforschung, FernUniversitaet - Gesamthochschule - Hagen, Hagen, 1995.
- [13] Romizowski, A.J., *Designing instructional systems. Decision making in course planning and curriculum design*, Kogan Page, London, 1981.
- [14] Romizowski, A.J., *Developing auto-instructional materials. From programmed texts to CAL and interactive video*, Kogan Page, London, 1986.
- [15] Sinderen. M., van, L. Ferreira Pires, C.A. Vissers, and J-P. Katoen, "A design model for open distributed processing systems", *Computer Network and ISDN systems*, **27**, 1995, 1263-1285.
- [16] Vissers, C.A., & M. van Sinderen, "Understanding Architectural Semantics", in: J. De Meer, V. Heymer, and R. Roth (Eds.) *Open Distributed Processing*.

Elsevier Science Publishers B.V., North-Holland, 1994, 367-368.

- [17] Franken, H.M. and M.K. de Weger, A modelling framework for capturing business process dynamics. Accepted for publication in: *Journal of Business Change and Re-Engineering*.
- [18] Rational Software Corporation, "*UML Semantics*", version 1.0, Santa Clara, CA, 13 January 1997.