Modeling Units of Assessment for Sharing Assessment Process Information: towards an Assessment Process Specification

Modeling Units of Assessment for Sharing Assessment Process Information: towards an Assessment Process Specification

Yongwu Miao, Peter Sloep, and Rob Koper

Educational Technology Expertise Center,
Open University of The Netherlands,
{Yongwu.Miao, Peter.Sloep, Rob.Koper}@ou.nl

Abstract. IMS Question and Test Interoperability (QTI) is an e-learning standard supporting interoperability and reusability of assessment tests/items. However, it has insufficient expressiveness to specify various assessment processes, especially, the new forms of assessment. In order to capture current educational practices in online assessment from the perspectives of assessment process management, we extend QTI and IMS Learning Design (LD) with an additional layer that describes assessment processes in an interoperable, abstract, and efficient way. Our aim is an assessment process specification that can be used to model both classic and new forms of assessment, and to align assessment with learning and teaching activities. In this paper, the development of the assessment process specification and its benefits and requirements are described. A conceptual model, the core of the assessment process specification is presented. The proposed conceptual model has been subject to a first validation, which is also described.

Keywords: e-learning standard, IMS QTI, IMS LD, assessment process specification, and new forms of assessment.

1 Introduction

IMS Question and Test Interoperability (QTI) [20] is an open technical e-learning standard which was developed to support the interoperability of systems and reusability of assessment resources. QTI addresses those assessment types for which an unambiguous definition in technical terms can be specified such as multiple-choice and filling-in-blank. In addition, QTI provides sufficient flexibility to grow into the advanced constructed-response items and interactive tasks we envisage as the future of assessment [1]. Recently, many QTI-compatible systems and assessment items have been developed (e.g., APIS [2], AQuRate [3], QuestionMark [21], and R2Q2 [22]). The development and application of QTI-compatible systems will promote and accelerate the exchange and sharing of assessment resources across platforms.

However, QTI provides no means to support the design and management of assessment processes. Specifically, it ignores who will be involved and what roles they will play, what kinds of activities should be performed by whom and in which sequence, what assessment resources will be produced and used in an assessment process, and what dynamic changes may happen and under which conditions. In short, it provides insufficient support for the representation and execution of an assessment plan. Furthermore, QTI does not sufficiently emphasize the support for 1) the integration of assessment with learning, and 2) competence assessment.

Integration of assessment with learning: according to Biggs [4], teaching, learning and assessment interact in modern learning, and this requires that curriculum objectives, teaching and learning activities and assessment tasks are aligned. Many researchers (e.g., Boud [6], Bransford et. al. [8], Brown & Knight, [10]) have emphasized the importance of formative assessment in student learning. As Black and Wiliam [5] pointed out, formative assessment that precisely indicates student strengths and weaknesses and provides frequent constructive and individualized feedback leads to significant learning gains if compared to a traditional summative assessment. However, QTI is just a specification about question definitions and response processing, and has nothing to do with teaching and learning activities. Conversely, IMS Learning Design (LD) [16] is used to support teaching-learning processes, but cannot explicitly support assessment.

Competence assessment: there is a marked tendency to place ever more emphasis on general competences in education and, therefore, in assessment too. Information gathering for the assessment of competences is increasingly based on qualitative, descriptive and narrative information, in addition to quantitative, numerical data. Such qualitative information cannot be judged against a simple, pre-set standard. Although classic forms of assessment still can be used for competence assessment, they do not suffice. Competence assessment relies mainly on new forms of assessment. Examples of new forms of assessment are self- and peer assessment, 360 degree feedback, progress testing, and portfolio assessment. These innovative forms of assessment address complex traits of students and foster deep learning [7], [13], [25]. However, these innovative forms of assessment are process-based and involve multiple persons in multiple roles. As already argued, they cannot be expressed using QTI alone.

Several software tools that support various forms of assessment have been developed, such as SPARK [11], Peer Grader [12], and eSPRAT [17]. However, these tools cannot support interoperability, reusability, and integration with learning activities, because each tool has its own data structure. In order to orchestrate various assessment-relevant activities performed by multiple roles/participants and, in particular, to address the problems described above, we have set out to extend QTI and LD with an additional layer that describes assessment processes in an interoperable, abstract, and efficient way. The aim is an assessment process specification (APS) that should facilitate experts and practitioners to share assessment process information. It is expected that APS can provide the means for defining assessment processes, as an internal part of the design process of a unit of learning (UoL), by combining new types of assessment with the ones already included in QTI specification [24]. As a first step towards APS, we developed a conceptual model, the core of APS. In this paper, we identify the requirements for the APS. Then we present the conceptual model, which represents the main concepts and their relations. This conceptual model has been validated by using literature and case studies. We conclude the paper with some indications of future work.

2 Objectives, Approach, Benefits, and Requirements

In practice, there are many different assessment process models (sometimes described as assessment plans and scenarios) and new models will be developed at all time. In order to support online assessment planning and execution, developing a software tool for each separate assessment process model would be inefficient. Based on our experience with the development of the IMS Learning Design specification (LD), a standard educational modeling language used to specify a wide range of pedagogical approaches/strategies, we set out to develop an abstract notation based on various assessment process models. We expect that the abstract notation can be used to specify a wide range of assessment approaches/strategies if not all. In a way analogous to extending IMS Meta-Data and IMS Content Package (CP) to LD, we extended QTI by applying the framework of LD to APS: from a content-based specification to an activity-centric and process-oriented specification. And similar to the term learning design in LD, the term assessment design refers to the formal description of an assessment approach/strategy. Also, similar to the unit of learning (UoL) in LD, a unit of assessment (UoA) in APS is a package of an assessment design and associated assessment resources (e.g., QTI assessment items/tests) using IMS CP.

As proposed in [18], an assessment process can be formally modeled through a combined use of LD and QTI. However, by adopting this approach, the user has to model assessment-specific concepts (e.g., trait, responding, and comment) using generic concepts (e.g., outcome variable, learning-activity, and property). The user must deal with all the complexity of integrating QTI resources into LD, binding LD properties to QTI outcome variables, and so on. In comparison with typical software development approaches, such a process modeling and execution approach is efficient and flexible for technical experts. However, for practitioners it is very difficult if not impossible to work at this abstraction level [18]. Therefore, APS should be abstracted at an appropriate level. For APS to be useful, on the one hand, the notation should be sufficiently general to represent various characteristics found in different assessment process models. On the other hand, it should be sufficiently specific to have expressiveness for modeling assessment processes stronger than provided by LD and QTI. To achieve this goal, we applied a domain-specific modeling approach with the intent to raise the level of abstraction beyond QTI and LD; we did so by choosing the vocabularies used in the domain of assessment. These vocabularies provide natural concepts that describe assessment in ways that practitioners already understand. They do not need to think of solutions in coding terms or/and generic concepts [19]. Once practitioners have specified a solution in terms of the vocabularies, an interpreter will automatically transform the solution represented in the high-level process modeling language into a formal model represented in LD and QTI. That is, a UoA will be translated into a UoL with QTI resources, which then can be instantiated and executed in existing integrated LD and QTI compatible run-time environments.

Based on APS, it is possible that practitioners can develop UoAs. The benefits of the UoA are:

1. A UoA, as a description of a use case represented in a standard language, can facilitate understanding, communication, and reuse of a variety of assessment practices.

2. A UoA provides a base for analyzing and evaluating an assessment plan by using formal techniques (e.g., validation and simulation) for a deeper understanding, comparison, and improvement.
3. An executing UoA can scaffold learners, tutors, and other stakeholders to perform the tasks suggested by providing guidance and awareness information, such as current status, suggested next steps, available resources, and decisions (e.g., terminating activities and initiating a service).
4. An executing UoA can enforce learner, tutors, and other stakeholders to strictly follow a plan by configuring a workspace for carrying out prescriptive tasks (e.g., doing an examination with a QTI tool and demonstrating skills with a simulator), by controlling and changing the sequence of activities based on the execution state and circumstantial information, and by orchestrating the efforts made by different roles/participants.

For all these benefits to materialize, APS has to match the following requirements (derived from [14, 15]):

1. Completeness: The APS must be able to fully describe the whole assessment process, which consists of various types of activities performed by various roles that use a variety of assessment resources.
2. Flexibility: The APS must be able to express the assessment meaning and the functionality of the different data elements within the context of a UoA. It must be sufficiently flexible to describe a wide range of assessment strategies/approaches.
3. Adaptability: The APS must be able to describe adaptation aspects within a UoA, so that the assessment resources and assessment activities within a UoA can be adapted to the preferences, portfolio, educational needs, performances, assessment results and situational circumstances of users.
4. Compatibility: The APS must be able to match and integrate available standards and specifications, such as the IMS (imsglobal.org) and IEEE LTSC (ltsc.ieee.org). In particular, it should be compatible with existing relevant standards such as QTI and LD.

APS, following common IMS practice, should consist of: (a) a conceptual model, (b) an information model, (c) XML Schemas binding, (d) a Best Practices and Implementation Guide. Among these, the conceptual model is the core of the specification. This paper focuses on the conceptual model. Admittedly, reusability, formalization, and reproducibility are also requirements of a specification. Because these requirements deal with technical issues in respect to the formal representation and run-time execution, they will not be discussed in this paper.

3. The Conceptual Model of APS

The conceptual model of the APS represents main concepts and their relations. In this section, we will express it as a semantic aggregation model, a conceptual structure model, and a process structure model.

3.1 Semantic Aggregation Model

Fig. 1 represents the conceptual model of the semantic aggregation levels in APS. The model shows the levels of semantic aggregation. The semantically highest level is *assessment design*, which aggregates a collection of *components* and a *method*. A *component* can be one of five types: *role*, *artifact*, *service facility*, *information resource*, and *property*. More detailed categories of each component are also depicted in Fig. 1. They will be familiar to those who know LD, as will be several aspects to be discussed subsequently. A *method* consists of one or more assessment *scenarios* and a set of *rules*. An assessment *scenario* consists of several sequential *stages*. Each *stage* consists of a set of *activities* and/or *activity-structures*. Each *activity-structure* consists of a set of sequential, selectable, concurrent, or alternative activities/activity-structures. A *rule* consists of a set of *conditional expressions* and a set of *actions* in a structured if-then-else/else-if format. The sub-types of each concept are illustrated in Fig. 1 as well. Because of the limited space available, this paper only briefly describes the semantics of the important vocabularies and attributes.

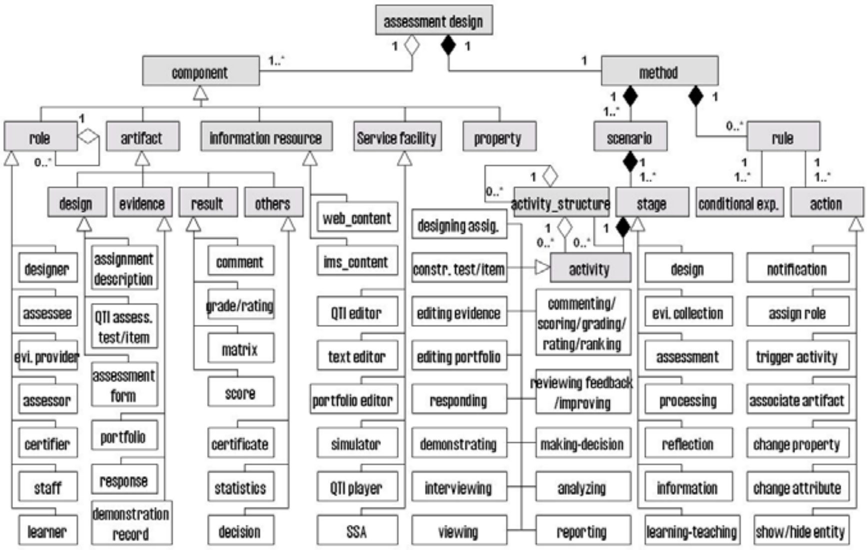


Fig. 1. Semantics Aggregation Model

Assessment design is a description of an assessment method that yields the appropriate evidence of assessees' competences and produces assessment results through following some rules. It has attributes such as *identifier, title, description, assessment objectives, assessment types*, etc. The *identifier, title,* and *description* are trivial attributes for presenting semantics and will not be mentioned any more when presenting other vocabularies. *Assessment-objective* is used to describe the intended outcome of the assessment in terms of information resources or competence proficiencies. *Assessment-type* is used to define a way to yield and evaluate evidence. The possible choices are *classic test*, *self-/peer assessment*, *portfolio assessment*, *360 degree feedback*, etc. Each choice will provide additional restrictions to the conceptual model. For example, in a peer assessment many concepts will be excluded. A detailed example of a peer assessment is given in [19].

Role is used to distinguish different types of participants in an assessment process. Several roles have been pre-defined such as *designer, assessee, evidence provider, assessor, certifier, learner,* and *staff.* Each role can be refined or customized further, for example, *candidate* and *assessment-taker* to *assessee* and *reviewer, rater,* and *evaluator* to *assessor*. Note that a user may be able to have several roles at the same time and that many users can play the same role. Two important attributes of a role are *role-property* and *role-member-property*. A declaration of a *role-property* is just instantiated once in an execution to present a characteristic and a state of the whole role, for instance, whether all assessors have finished commenting. A declaration of a *role-member-property* will be instantiated for every user who has this role, for instance, a *trait* is a pre-defined *role-member-property* for *assessee*. A *role-member-property* of the root role can be declared locally or globally.

Stage is used to distinguish different focuses within the whole assessment process, and *activity* is a logical unit of task performed individually or collaboratively within a *stage*. As shown in Fig. 1, APS has seven pre-defined types of stages and fourteen types of activities, which have more assessment-specific semantics than the generic terms such as act and activity in LD. However, the constraints about the aggregation relations between the stage types and activity types have not been illustrated in Fig. 1 for reasons of readability. In fact, in each type of stage only some types of activities are allowed. For example, *constructing QTI items/test* and *designing demonstration assignment* can only been specified in the design stage. In the *evidence collection stage* only *responding QTI test/item, editing portfolio, editing evidence,* and *demonstrating* are allowed. Note that *learning-activity* and *support-activity* (not shown in Fig. 1) are defined to be similar to those in LD; they can be performed in the learning-teaching stage. In addition, more than one activity can be performed within the same stage. A set of *activities* can be grouped as an *activity-structure*. Four types of activity-structures are specified: *sequence-structure* (all activities will be performed in a prescribed sequence), *selection-structure* (a given number of activities selected from a set of candidate activities will be performed in any order), *concurrent-structure* (a set of activities are performed concurrently), and *alternative-structure* (one or more activities selected from a set of candidate activities according to prescribed conditional expressions will be performed). A *stage,* an *activity-structure,* and an *activity* have common attributes such as *completion-condition* (e.g., user-choice, time-over, artifact-submitted, and even user-defined conditions) and *post-completion-actions* (e.g., show/hide information/activity).

Artifact is used to represent the information object created, introduced, and shared within and/or across activities as an intermediate product and/or a final outcome. As Fig. 1 shows, a particular type of artifact will fall into one of four categories: *design, evidence, assessment result,* and *others*. Each type of concrete *artifact* has a specific data-type and will be handled using appropriate *services*. For example, a *comment* is an information object created by using a QTI player as a response to an extended-text-interaction or an output of a text editor. Some attributes of an artifact can be used to capture generic information such as *status, size,* and *media-type* (e.g., a MIME-type). For example, an *evidence* or *demonstration* may be in the form of Text, XML, URL, an image, or a video. Other attributes are used to describe association information such as *source-activity, destination-activities,* and *default-service-type*. *Information resource* differs from *artifact* because it is available and keeps unchanged during the whole assessment process.

Service is used to specify the type of “service” for handling certain types of *artifacts* (e.g., QTI player and portfolio editor) or/and for facilitating communication and collaboration (discussion forum and text editor). As shown in Fig. 1, the APS extends LD built-in services by including several assessment-specific services and some general-purpose services which can be used for assessment. It is allowed to introduce new types of services when modeling and executing a UoA.

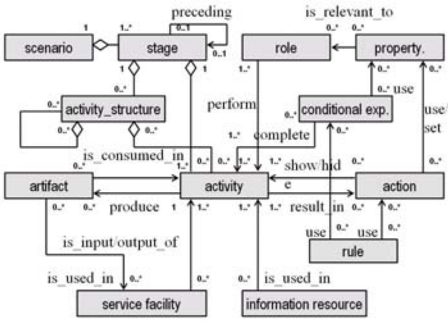
Property is designed for capturing any information relevant to the process or to certain roles. The role relevant property has been discussed above. A process relevant property will be instantiated once for each execution of a UoA or for all executions, depending on whether it is declared by the user as a local property or a global one. Examples of the process relevant properties are a process status, a decision, etc.

Rule consists of *conditional expressions* and a set of *actions* and/or embedded rules in a form of If (conditional expression) Then (actions) Else (actions/rules). A *conditional expression* is a logical expression on the attributes (e.g., assessment-type, activity-status, user-in-role, role-in-activity, artifact-default-service, and etc.) and properties. An *action* is an operation performed by the system. As shown in Fig. 1, exemplar actions are *change attribute* (assigning a value to an attribute), *associating artifact* (assigning an artifact as an input/output of an activity), and *show/hide entity* (making a scenario/activity/information visible for the user), etc. Thus, a *rule* can be used to model dynamic features and support adaptive assessment.

3.2 Conceptual Structure Model

Fig. 2 illustrates the main structural relations between the concepts. By design, APS is an activity-centric model. The core idea is: following certain *rules* people with various *roles* perform *activities/activity-structures* allocated to them; they do so in *stages* using *service facilities* and *information resources* in order to consume and produce *artifacts*. When presenting the semantics of each concept above, we have mentioned some structural relations. In this sub-section, we focus on discussing the structural relations around the activity.

The important attributes of an *activity* are *roles involved, input and output artifacts, services needed, information resources referred to, completion-conditions*, and *post-completion-actions*. For each particular type of activity, APS specifies a few particular structural relations with certain types of roles, artifacts, and services. For example, a *responding* activity is associated with an *assessee*, a *QTI test/item*, a *QTI player*, and a *response*. The structural relations between these components are pre-defined in APS. Therefore, in design-time, after an activity with a certain type has been created, the associated components (e.g., roles involved, input and out artifacts, and services needed) will be created automatically and the values of some attributes of these components (for specifying types and association relations) can be assigned automatically. Another example is *improving* activity, which can be specified according to the definition of the activity specified in the *evidence collection* stage. For instance, if the type of activity arranged in the *evidence collection* is *responding* (e.g., answering a list of multiple-choice questions or writing an essay), the *improving* activity will be configured in such a way that it associates the *improving* activity with a *QTI player*, the original *QTI test/item*, and the *response* of the user. Obviously, we cannot detail here all pre-defined structural relations between all types of roles, activities, artifact, and services. Please note, though, that a user-defined *rule* can be used to specify and change the pre-defined structural relations by the user. For example, the type of the input artifact used for the *commenting* activity is pre-defined in APS as an extended text interaction of QTI. The user can change the definition of a given *commenting* activity by assigning a value (e.g., Text) of the input artifact type. Then the default service (a text editor in this case) for handling this artifact type will be arranged accordingly. Thus, the structural relation specified in the rule can help the run-time system pass the text-based document as an input artifact of the activity when invoking a text editor.



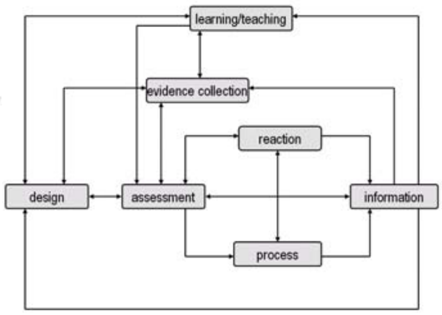


Fig. 2. Conceptual Structure Model **Fig. 3.** Process Structure Model

3.3 Process Structure Model

Fig. 3 illustrates the process structure relations between the seven stages (cf. Fig. 1). Usually both the start point and end point of an integrated learning and assessment scenario are the *learning/teaching stage*. A complete process may consist of all types of stages in a sequence of *learning/teaching, design, evidence collection, assessment, reflection, process, information,* and *learning/teaching*. Sometimes one or more stages can be excluded. For example, the *design stage* may be excluded if the method for collecting evidence and the assessment form/criterion have been designed before the start of the execution and will be available during the execution. In a particular case, a teacher may grade students based on memory and then an *evidence collection stage* can be excluded. In contrast, some stages may be repeated several times. For example, further evidence may need to be gathered after an initial assessment; and even a *design stage* may be needed for creating additional assessment items according to the user's response at run-time. Sometimes a peer assessment can be designed in a way that enables the *assessee* to review the feedback and request for elaboration. The assessor may provide further comments and detailed explanations. In some complicated cases, multiple loops may be defined within a scenario. Therefore, many concrete assessment process models can be derived from this generic process structure model. In particular, these concrete assessment process models can be designed differently at the component (e.g., role, activity, artifact, and service) level.

4. An Initial Validation of the Conceptual Model

Validation studies have been conducted to test if the conceptual model would meet the requirements described in section 2. In this section, we present the results of these initial validation studies.

Completeness: The OUNL/CITO model [9] is an extensible educational model for assessment, which provides a broad basis for interoperability specifications for the whole assessment process from design to decision-making. The OUNL/CITO model was validated against Stiggins' [23] guidelines for performance assessments and the four-process framework of Almond et al. [1]. In addition, the model's expressiveness was investigated through describing a performance assessment in teacher education using OUNL/CITO model terminologies. Brinke et. al. [9] reported that the OUNL/CITO model met the requirement of completeness. This paper bases the APS validation study of completeness on the OUNL/CITO model. Indeed, the conceptual model of APS is based on the OUNL/CITO model. However, like QTI, the OUNL/CITO model is a document-centric one. The concepts of stage and corresponding activities are not explicitly included in the model although they are conceptually used to develop and organize the model. As a consequence, an assessment description based on the OUNL/CITO model cannot be executed by a process enactment service, because important information about control flow and artifact flow from one activity/role to another is missing in the OUNL/CITO model. Nevertheless, APS extracts almost all concepts represented explicitly and implicitly in the OUNL/CITO model. We reformulated these concepts from a perspective of process support. APS explicitly formalizes concepts such as stage, activity, artifact, service, and rule, and re-organizes them around the activity. As already mentioned, like LD, APS is an activity-centric and process-based model. We removed some run-time concepts such as *assessment-take* and *assessment-session* from the OUNL/CITO model, because they are related to the execution of the model. Moreover, because some concepts such as *assessment policy*, *assessment population*, and *assessment function* are complicated for ordinary teachers and instruction designers, APS does not explicitly include them. If need be, the attribute *description of the assessment design* in APS can be used to represent these concepts implicitly. In addition, terms such as *assessment plan* and *decision rule* are replaced by other terms such as *UoA* (in fact, an instance of a UoA) and *rule*, which are expressed in a technically operational manner. We conclude that all concepts in the OUNL/CITO model can be mapped to APS. Furthermore, in order to model formative assessments, APS integrates the *learning/teaching stage* and the activities specified in LD. Thus APS meets the basic requirements of completeness.

Flexibility: As mentioned when we presented the process structure model in section 3.3, APS enables users to specify various assessment process models by tailoring the generic process structure model and by making different detailed designs at the component (e.g., role, activity, artifact, and service) level. We tested the flexibility by conducting several case studies. In order to explain how to model a case based on APS, we present a simple peer assessment model. As shown in Fig. 4, this three-stage model involves two learners. In the first stage, each learner writes a different article and sends it to the peer learner. Then each learner reviews the article received and sends a comment with a grade back to the peer learner. Finally, each learner reads the received feedback. In the same way, we have tested three more complicated peer assessment models, a 360 degree feedback model, and a programmed instruction model. For lack of the space, a detailed description of these case studies is omitted. All validation studies, however, reveal that APS is sufficiently expressive to describe these various forms of assessment. Thus APS supports flexibility to at least some extent.

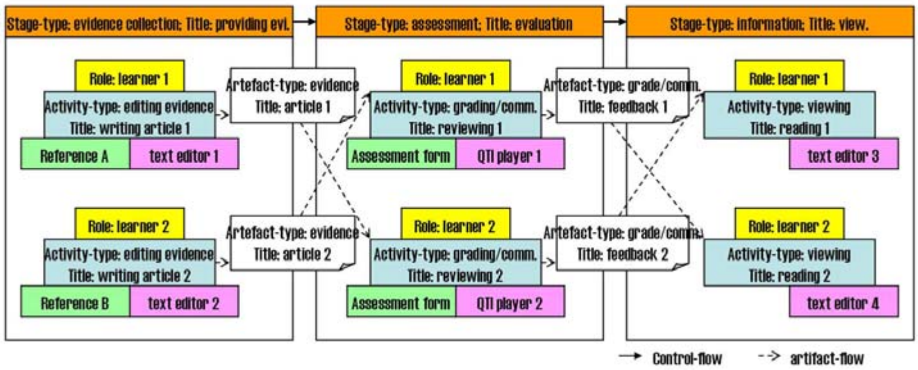


Fig. 4. A Simple Peer Assessment Model

Adaptability: Adaptation can be supported in APS at two levels. The first is at the assessment task level. As we know, QTI can support adaptation by adjusting assessment item/test (e.g., questions, choices, and feedback) to the responses of the user. APS, however, supports adaptation at task level much more broadly. According to an assessee's personal characteristics, learning goals/needs, response/performance, and circumstantial information, an assessment-specific activity can be adapted by adjusting the input/output artifact, service needed, completion-condition, post-completion-actions, and even the attributes of these associated components. For example, a rule could be: if (learning_goal:competenceA.proficiency_level >= 5) then (a test with a simulator) else (a test with a questionnaire). The second level is the assessment process level. APS supports adaptation of assessment strategies and approaches by changing the process structure through showing/hiding scenarios, changing the sequence of stages, showing/hiding activities/activity-structure. The adaptation is expressed as rules in APS. An example of such a rule is: if (learning within a group) then (peer assessment) else (interview with a teacher).

Compatibility: The domain of application of APS overlaps with those of both LD and QTI. However, they operate at different levels of abstraction. LD and QTI provide a wealth of capabilities for modeling assessment process models, but the code can become lengthy and complex. For this reason, we developed APS at a higher level of abstraction by providing assessment-specific concepts. These built-in constructs provide shortcuts for many of the tasks that are time-consuming if one uses LD and QTI to model them. However, APS is built on the top of LD and QTI, and the assessment-specific concepts are specializations of the generic concepts in LD and QTI. For example, concepts such as *constructing assessment item* and *commenting* in APS are specializations of the generic concept *support-activity* in LD. An assessment process model based on APS can be transformed into an executable model represented in LD and QTI. Thus, we should be able to use an integrated LD and QTI run-time environment to execute various forms of assessment based on APS. In addition, APS will be organized using the IMS Content Package specification. It can use IEEE Learning Object Metadata (LOM) to describe the meta-data of elements in APS. Moreover, the IMS Reusable Definition of Competency or Educational Objectives can be used to specify traits and assessment objectives. The IMS ePortfolio can be used to model portfolios (coupled with artifacts in APS) and integrate a portfolio editor. The IMS Learner Information Profile can be used to import global properties from a run-time environment and export them to it. IMS Enterprise can be used for mapping roles when instantiating a UoA. Therefore, APS is compatible with most existing, relevant e-learning technical specifications.

5. Conclusions and Future Work

This paper addressed the problems one faces when attempting to use QTI and LD to support the management of assessment processes, in particular, formative assessment and competence assessment. In order to support the sharing of assessment process information in an interoperable, abstract, and efficient way, we developed APS as a high-level assessment-specific process modeling language. We have developed the conceptual model of APS by adopting a domain-specific modeling approach. The conceptual model has been described through detailing the semantics aggregation model, the conceptual structure model, and the process structure model. The first validation study has been conducted through investigating whether the conceptual model of APS meets the requirements of completeness, flexibility, adaptability, and compatibility. The results suggest that the model does indeed do so.

APS should meet additional requirements (e.g., reproducibility, formalization, and reusability), which we intend to investigate after the development of the information model and XML Schemas binding. In order to enable practitioners to easily design and customize their own assessment process models, an authoring tool for modeling assessment processes with APS will be developed in the near future. In order to execute an instantiated model in existing LD and QTI compatible run-time environments, transformation functions have to be developed as well. Then we will carry out experiments to investigate the feasibility and usability of APS and the corresponding authoring tool. Finally, we will propose APS as a candidate, new open e-learning technical standard.

Acknowledgments. The work described in this paper has been fully supported by the European Commission under the TENCompetence project [project No: IST-2004-02787].

References

1. Almond, R. G., Steinberg, L. & Mislevy, R. J.: A sample assessment using the four process framework. CSE Report 543. Center for study of evaluation. Los Angeles, University of California (2001)
2. APIS: http://www.elearning.ac.uk/resources/1apisoverview.

3. AQuRate: http://aqurate.kingston.ac.uk/index.htm
4. Biggs, J. B.: Teaching for Quality Learning at University. Buckingham, Society for Research in Higher Education & Open University Press (1999)
5. Black, P. & Wiliam, D.: Assessment and classroom learning. Assessment in Education, 5(1), 7-74, (1998)
6. Boud, D.: Enhancing Learning through Self-Assessment. Routledge (1995)
7. Boud, D., R. Cohen, et al.: Peer Learning and Assessment. Assessment and Evaluation in Higher Education, 24(4): 413-426 (1999)
8. Bransford, J., Brown, A., & Cocking, R.: How People Learn: Mind, Brain, Experience and School, Expanded Edition. National Academy Press, Washington, DC (2000)
9. Brinke, D.J., Van Bruggen, J., Hermans, H., Latour, I., Burgers, J., Giesbers, B. & Koper, R.: Modeling assessment for re-use of traditional and new types of assessment, Computers in Human Behavior, 23, 2721-2741 (2007)
10. Brown, S., & Knight, P.: Assessing Learners in Higher Education. Kogan Page, London (1994)
11. Freeman, M. & McKenzie, J.: Implementing and evaluating SPARK, a confidential web-based template for self and peer assessment of student teamwork: benefits of evaluating across different subjects. British Journal of Educational Technology, 33(5): 553-572 (2002)
12. Gehringer, E. F.: Electronic peer review and peer grading in computer-science courses. In: Proceedings of the 32nd ACM SIGCSE Technical Symposium on Computer Science Education, Charlotte, North Carolina (2001)
13. Gipps, C.: Socio-cultural perspective on assessment. Review of Research in Education, 24: 355-392 (1999)
14. Koper, E.J.R.: Modelling Units of Study from a Pedagogical Perspective: the Pedagogical Meta-model behind EML, (provided as input for the IMS Learning Design), Educational Technology Expertise Centre, Open University of the Netherlands (2001), Available at: http://hdl.handle.net/1820/36.
15. Koper, R. & B. Olivier.: Representing the Learning Design of Units of Learning. Journal of Educational Technology & Society, 7(3): 97-111 (2004)
16. LD: http://www.imsglobal.org/learningdesign/index.cfm.
17. Lockyer, J.: Multisource feedback in the assessment of physician competencies. Journal Contin Educ. Health Prof., 23(1): 4-12 (2003)
18. Miao, Y. & Koper, R.: An Efficient and Flexible Technical Approach to Develop and Deliver Online Peer Assessment, In: Proceedings of CSCL 2007, 502-511, New Jersey, USA, (2007)
19. Miao, Y. & Koper, R.: A Domain-specific Modeling Approach to the Development of Online Peer Assessment. In T. Navarette, J. Blat & R. Koper (Eds.), In: Proceedings of the 3rd TENCompetence Open Workshop on Current Research on IMS Learning Design and Lifelong Competence Development Infrastructures, 81-88. Barcelona, Spain. (2007), Available online at: http://hdl.handle.net/1820/1098.
20. QTI: http://www.imsglobal.org/question/index.html.
21. QuestionMark: http://www.questionmark.com/uk/index.aspx
22. Wills, G., Davis, H., Chennupati, S., Gilbert, L., Howard, Y., Jam, E. R., Jeyes, S., Millard, D., Sherratt, R. and Willingham, G.: R2Q2: Rendering and Reponses Processing for QTIv2 Question Types. In Danson, M (ed) Proceedings of the 10th International Computer Assisted Assessment Conference, pp.515-522. Loughborough University, UK (2006)
23. Stiggins, R.J.: Het ontwerpen en ontwikkelen van performance-assessment toetsen. [Design and development of performance assessments]. In J.W.M. Kessels & C.A. Smit (Eds.) Opleiders in organisaties/Capita Selecta (afl. 10, pp. 75-91). Deventer: Kluwer (1992)
24. TENCompetence project: http://www.tencompetence.org
25. Topping, K. J.: Peer assessment between students in colleges and universities. Review of Educational Research, 68: 249-276 (1998)