A Catalogue of Primitive Scenario-Types. The First Step to the Automation of Learning Scenarios

A. Rius Gavidia [*,1], M.A. Sicilia Urbán[2],E. García Barriocanal[2]

[1] UOC Open University of Catalonia, Department of Computer Science, Multimedia and Telecomunication, Rambla Poblenou, 156, 08018 Barcelona, Spain. mriusg@uoc.edu

[2] UAH University of Alcalá, Department of Computer Science, Ctra. Barcelona km 33,6, 28871 Alcalá de Henares, Madrid, Spain. {msicilia@uah.es, elena.garciab@uah.es}

Abstract: Issues such as the standardization of Learning Objects (LO) and their reusability have been the target of numerous and important contributions by experts in the area. Other issues also related to LMS (Learning Management Systems), and on which there is not much knowledge is that of the automation of processes that usually occur in these kinds of systems and their boundaries. In this sense and with the aim of achieving executable process specifications, this paper presents a catalogue of basic or primitive scenarios that will be the basis of a wide and more complete catalogue. In order to achieve a reduced and sufficient set of primitive scenarios, we have taken as a reference some existing specifications (IMS DRI, LORI, IEEE LTSC). Later, we will present an example of a more complex scenario using those basic scenarios and showing the rules for composing them and creating another one.

Introduction

The importance of cataloguing scenarios consists in the fact of having descriptions for each of the most usual functions related to LMS and their boundaries independently of the system's platform, the repositories to be accessed, the heterogeneity of content and format and the kind of languages to request resources and learning processes. Besides, the construction of an interface according to the learning environment and the interactions between components of LMS promotes interoperability and portability among systems and, furthermore, it facilitates the extensibility of the repertory of LMS functions. In order to automate basic scenarios through executable specification of processes, first of all, it is necessary to obtain a cataloguing of the same.

This paper focuses on the obtaining of the catalogue of those scenarios that every LMS needs to achieve their most elementary functions. With reference to scenarios and their specification, there is some previous work in that field. The concepts scenario and scenario-type and the use of the LO metadata as the infrastructure to support the LMS functionality has already been suggested in (Sicilia et al. 05). Another proposal, which is more centred in scenario specification is that presented in (Sicilia et al. 04). It proposes to apply the Design by Contract Technique to the process specification; an idea that gives rise to the Semantic Conformance Profiles (SCP). The SCP, as a mechanism of specification at high level produces automation-oriented specifications, by means of the pre-conditions, the post-conditions and the restrictions for each scenario. And having these profiles as a starting point, (Rius, 2006) proposes the use of the Business Process Modelling Notation (BPMN) to describe scenarios, identifying sub-processes and the interactions among processes. Furthermore, it suggests using an executable specifying language to obtain an executable process specification from the BPM diagram.

Currently, no cataloguing of scenarios exists for LMS, although we have some specifications that shed light on the matter. IMS DRI (IMS-Digital Repository Interoperability) (IMS, 03) proposes a reference model for the communication among the service providers and the learning resources through LMS and it proposes a set of basic functions for repositories focused on the interchange functions. In contrast, LORI (CEN-Learning Object Repository Interoperability) (CEN, 05) is centred on an API, which allows a simple query interface, to access the learning resources provided by LMS. It presents two scenario types of inquiry and establishes communication between entities and it considers the possibility that both are repositories at the same time. The Draft Standard for Learning

[*] Ângels Rius Gavidia: e-mail: mriusg@uoc.edu, Phone: +34 933263740

Technology – Learning Technology Systems Architecture (LTSA) (LTSC, 01) provides a framework with the aim to facilitate the comprehension of systems such as LMS, their subsystems and the interaction with other related systems and it also proposes an architecture.

Consequently, these specifications have been used as a starting point to determine the basic scenarios that can be considered primitive in order to construct the catalogue that we want to create. The paper is organized in six sections. After this introduction, the second section introduces the subject matter; the third compares the scenario-types proposed by other types of implementation and standards. The next two sections, based on those, propose a catalogue of primitive scenario-types to cover basic functionalities in the boundaries of LMS and how it is possible to join them to create new scenario-types. The last section presents some conclusions and future work.

Basic concepts related to scenarios

Fist of all, we have to define what we understand by scenario, scenario-type and other related concepts. A scenario-type or a specification of scenario is consistent description of the behaviour of the entities involved in a process and the different states of the system in order to achieve an objective. Thus, a scenario would be an instance of scenario-type and it could be concreted by a use case. Several entities take part in a scenario, each of them has a role and they interact with each other. These entities will be called actors and the instance of an actor in a concrete scenario from now on will be called actor-role.

As the LMS environment is a distributed environment, communication among the actors will be established using flow messages. A message-type can be considered as the description of an operation in which the kind of communication among actors, the structure of the message that will be sent and the direction of the shipment are established. The message, in consequence, will be an instance of the message-type.

Since several actors take part in a scenario and all of them collaborate in the achievement of the objective, there are a set of processes where these actors take part and a protocol to define the way in which the message interchange is going to be carried out. If there is a one only process that is in charge of dictating the rules that guide the communication process among all actors, we say that the orchestration specification is necessary, while if the responsibility is shared among several processes, a choreography specification is required.

Scenario-Types in other specifications and standards

In this section, the contributions of those specifications mentioned before are exposed, since through comparing them, a more global vision is produced and it has helped us to obtain a minimum set of scenarios type to build our catalogue.

In IMS DRI (IMS, 03) four actors are proposed (Creator, Learner, Infoseaker, Agent) and some scenario type (Aggregator, Federator, Translator) in order to resolve the problem of multiple metadata representation and different access methods. It also proposes pairs of functions (Search/Expose, Gather/Expose, Submit/Store) to search, access, store and retrieve metadata or data in the repository. In contrast, LORI (CEN, 05) only considers two actors; the source (SQI source) and the target (SQI target) and a query scenario that can be implemented in an asynchronous or synchronous way. Furthermore, LORI focuses on queries more than on interchange of data among repositories. It seems that LORI and IMS DRI are complementing them. On the other hand, the IEEE LTSC specification (LTSC, 01) offers a wider perspective. It considers not only repository issues, since it treats other issues related to the learning process. Thus, apart from the Learner actor, it considers other actors such as Delivery, Evaluation and Coach to support the learning process, transferring learning resources, evaluating and the guiding of the learning process respectively. At the scenario-type level, it presents a reduced set of scenario type (bidirectional and unidirectional communication, store and retrieve), but it proposes a more complete list of use cases that involve the analysing of all interactions among actors and resources in the learning processes.

The main Primitive Scenario-Types

In order to obtain a minimum set of primitive scenarios which define the basic functions that usually an LMS supports in a very clear way, we have identified the main basic scenarios that the previous specifications and standards mentioned have already taken into account. The criteria have been to isolate the fundamental functions in order to facilitate their future specifications and the extension of scenario-types cataloguing.

Name of primitive scenario	IMS DRI	LORI	LTSC
Send	Submit/Store		Assertion
Receive			Retrieve (if the source is a repository)
Send-Receive		Query	
Translation	Translator		
Aggregation	Aggregator (only for metadata)	As part of a query	
Disaggregation	As part of Federator function		
Distribution	As part of Federator function (information restricted to a query and including the management of responses)	As part of a query	
Save_in_repository	Submit/Store		Store
Search_repository	Search/Expose (the result restricted to metadata)		Retrieve
Search_repositories	Gather/Expose ((the result restricted to metadata and new metadata is obtained by aggregation)	Query	
Retrieve_from_respository	Request/Deliver		Retrieve
Erase_LR			
Update_LR			
Acquire_LR			
Assembly_LR			
Desassembly_LR			
Detect_paltform			
Publish_LR			
Evaluate_LR			

Table 1: Correspondence among primitive scenario proposed and other recognized proposals

Tab. 1 presents a correspondence between each suggested primitive scenario and those proposed by IMS LDRI (IMS, 03), LORI (CEN, 05) and LTSC (LTSC, 01), if it exists. Besides, it is possible to identify four categories of scenarios-type:

Transfer scenarios – Their mission is to make the transfer of information across the LMS among actors and/or repositories. The Send, Receive and Send-Receive scenarios assume that the information is to be sent or received in an appropriate format and the destination available. They are based on the IMS DRI (Submit function), LORI (Synchronous scenario) and LTSC (Assertion/Inquiry and Retrieve functions).

Preparing transfer scenarios – Four basic scenarios prepare the information to be understandable by the targets (actors or repositories): The Translation scenario facilitates the interchange formats, the Aggregation compose complex information from different and more simple information, the Disaggregation makes the inverse function than the Aggregation and the Distribution distribute a set of information into several targets. All these scenarios are inspired on IMS DRI proposal (Translator, Aggregator and Federator functions) and the query scenario, but simplifying them at the most (e.g.. the Federator function in IMS DRI disaggregate, distribute and aggregate in order to achieve the federated search).

Repository management scenarios – It includes the most basic functions on the repositories so that the saving, locating, updating and retrieving the learning resource is made possible The Save_in_repository and Retrieve_from_repository scenarios are proposed for a one only repository but they can be extrapoled to more repositories (eg.. it is possible to invoke Save in_repository scenario for each repository if previously the initial learning resource is disaggregated and distributed). The location of a learning resource is considered through two scenarios, one for the search in a concrete scenario and the other for the federated search. These scenarios are included in the mentioned proposals but here the search condition search can be a metadata set as occurs in IMS DR (MS, 03) or a generic query as LORI (CEN, 05). Furthermore, only the location is provided in our search scenarios and this does not includes the aggregation of metadata as Gather/Expose in IMS DRI. Other scenarios such as Erase_LR and Update_LR make possible the learning resource updating, if the rights to are access provided.

Support to learning process scenarios – These scenarios offer support to the learning process and they are centred in LO. Scenarios for LO acquisition, the LO assembling or desassembling to personalise them according to the learner needs, the LO publication taking into account the platform and the evaluation use of LO with respect to the learning objectives. These scenarios were pointed out in (Sicilia et al. 04). This group has to grow once the learning process is going to be defined; however it is now out of the scope of this paper.

Tab. 2 shows the SCP of primitive scenario-type proposed in order to present an automation-oriented specification of them. If we look through Tab. 2, we will see that some scenarios-type are related to the fundamental ones proposed by the specifications and standards mentioned before and some others are included because they are related to the learning process and we consider them as essential.

Primitive scenario	Pre-conditions	Restrictions	Post-conditions
Send	Target identifier Information to be sent	Target located Information in the target format	The information is sent to the target
Receive	Source waiting response	Information to be received in a comprehensive source format	The response is received by the source
Send-Receive	Target identifier Query to be sent Source identifier	Target and source localized and available Source and target using the same format	The query is sent to the target and the response is returned to the source
Translation	Information to be translated Source format Target format	Mapping between formats available	The information is translated into target format
Aggregation	Information to be aggregated Aggregation mode	Aggregated information is consistent and coherent	The integrated information available
Disaggregation	Information to be disaggregated Disaggregation mode	Disaggregated information is consistent and coherent	The disaggregated information is available
Distribution	Information to be distributed Distribution mode Target identifiers	Targets formats known	Information distributed among targets before being used
Save_in_repository	Resource to be saved Repository identifier	Repository available Writing rights	The resource is saved in the repository
Search_repository	Repository identifier Request_condition	Repository available Reading access rights	The requested resource is located
Search_repositories	List of repository identifiers Request_condition	Repositories available Coherent and comprehensive condition	A list of resources are located
Retrieve_from_repository	Target identifier Repository identifier	Repository available Reading access rights	The requested resource is obtained
Erase_LR	Resource identifier Repository identifier	Repository available Updating access rights	The resource is eliminated
Update_LR	Resource identifier Updating information Repository identifier	Repository available Updating access rights	The resource is modified
Acquire_LR	Resource Identifier Buying conditions Economic transaction data	Vendor system available Copyright and other rights Buying conditions satisfied	The resource is acquired
Assembly_LR	Resoruce_1 identifier Resource_2 identifier composition mode Repository_1, Repository_2	Repositories available Resources located Consistence and semantic coherence in the composition	New resource (includes metadata composition)

Primitive scenario	Pre-conditions	Restrictions	Post-conditions
Desassembly_LR	Resource identifier Repository identifier Decomposition mode	Repository available Resource located Decomposition produces independent resources	Resource_1 Resource_2
Detect_platform	Technical characteristics	Detailed list of device capabilities	Device self-description
Publish_LR	Resource identifier Presentation mode Publication site	Supported by platform target	The resource is published
Evaluate_LR	Resource identifier Evaluation criteria	Information about the resource use is available	Evaluation of resource is done

Table 2: Semantic conformance profiles of primitive scenarios

Composing Primitive Scenario-Types

Having the primitive scenario-types, we are going to illustrate how to create a complex scenario-type from primitive scenario-types. The scenario-type to be constructed is proposed to be called Save_composed_LR. It intends to locate two learning resources that once composed will create a new learning resource, which has to be saved in a repository. Briefly, the Save_composed_LR scenario-type is specified using its semantic conformance profiles which is presented in Tab 3.

New scenario	Pre-conditions	Restrictions	Post-conditions
Save_composed_LR	L_resource_identifier_1, Repository identifier_1, L_resoruce_identifier_2, Repository identifier_2, Composition_mode L_resource Repository Identifier	Repositories available Semantic coherence composition	The learning resource is composed in the repository Saving justification

Table 3: Semantic conformance profiles of a complex scenario

In order to construct this scenario, we propose to select the primitive scenario-types to be used and then decide the rules that combine them.

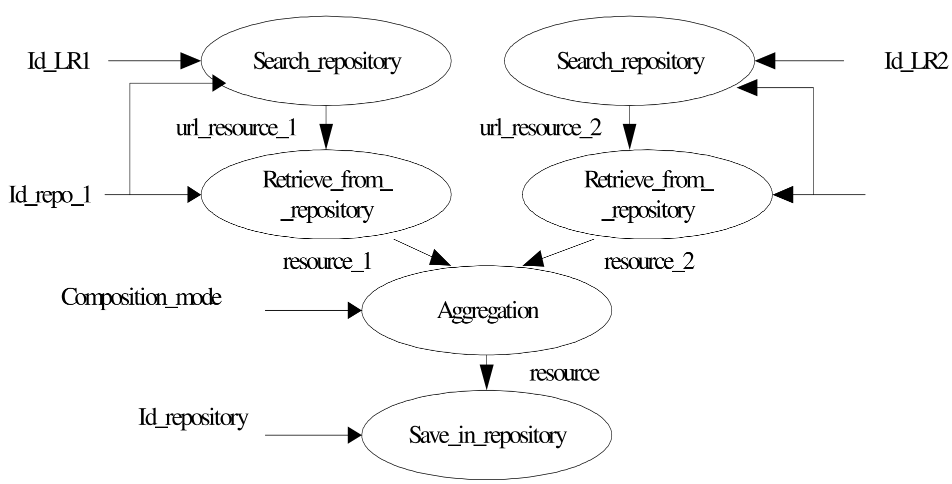

Figure 1: Save_composed_LR as a composition of primitive scenario-type sequence

Fig.1 graphically depicts this composition of the new scenario-type connecting the primitive scenarios.type according to the goal to be achieved. Firstly, the two source learning resources have to be located; hence we need the primitive scenario Search_repository to locate each of them. After that, the learning resources have to be retrieved, thus the primitive scenario-type Retrieve_from_repository is required twice to obtain both learning resources. When we have found them, they have to be composed according the composition mode and their metadata created, then the Aggregation scenario-type is required. Finally, if the composition is achieved successfully, the scenario-type Save_in_repository permits to save the resulting learning resource in the repository.

Conclusions and future work

There is not a cataloguing of learning scenarios available, therefore in order to achieve this, a set of primitive scenarios is presented in this paper which is defined as simple as possible with the aim of facilitating the composition of new scenarios and extend an initial catalogue. It is not the goal of this paper to present a complete catalogue of scenarios, but however it intends to show how complex scenarios can be constructed from primitive scenarios combining them from some rules. In this case, we have seen that some usual scenarios can be obtained only by the sequencing of the primitive scenario-types. Other composition rules should be considered in the future, as well as the formal formulation of pre-conditions and post-conditions and the connection between them in composed scenario-types.

IMS DRI, LORI and LTSC specifications have been considered to help us to choose the more basic scenario-types and then the design by contract technique has been applied to define in an automation-oriented way the SCP of them. The next step will consist in defining the scenarios in a formal way. It means that an ontology about learning scenarios has to be created and later these formal specifications are going to be expressed in an executable language specification, which will contribute to achieve some automation in the most common functions that any LMS usually uses.

References

(Sicilia et al. 05) Sicilia, M.A. and Lytras, M. (2005).Scenario-oriented reusable learning object characterizations. International Journal of Knowledge and Learning, 1(4):332–341.

(Sicilia et al. 04) Sicilia, M.A., García, E., Pagés, C., Sánchez, S.and Rius, A. (2004). Specifying semantic conformance profiles in reusable learning objects, Information Technology Based Higher Education and Training (ITHET04), 93–97

(Rius, 2006) Rius, A., Sicilia, M.A., García, E. and Macarro, G. (2006). Beyond contents and activities: Specifying learning processes in learning technology. Multimedia and Information and Communication (m-ICTE2006), V1, 301–305.

(IMS, 03) IMS Global Consortium Inc.(2003) IMS Digital Repositories Interoperability- Core Functions Information Model Version 1.0 Final Specification.

(CEN, 05) European Committee for Standardization (2005) A Simple Query Interface Specification for Learning Repositories. CWA 15454 Nov-2005.

(LTSC, 01) IEEE Learning Technology Standards Committee (LTSC), (2001). Draft Standard for Learning Technology – Learning Technology Systems Architecture (LTSA). Technical report, IEEE P1484.1/D9.

Acknowledgements

This work has been partially supported by Spanish Government grant ref. TIN2006-15107- C02-01/02 and by Internet Interdisciplinary Institute (IN3) grant ref. IN3-ACG6A.