Instructional Domain Analysis:
A methodological approach to Service-Oriented Learning

David Díez, Camino Fernández, Juan Manuel Dodero, Paloma Díaz, Ignacio Aedo
DEI Laboratory - Computer Science Department
Universidad Carlos III de Madrid, Spain
{david.diez, camino.fernandez, juanmanuel.dodero, mpaloma.diaz, ignacio.aedo}@uc3m.es

Abstract
The development of a learning system based on integrating a number of learning services needs both a technological platform to support the system as well as a methodology that manages the development process. Such technological platform is provided by the Service-Oriented Learning Architecture (SOLA) that defines an architecture style guiding several aspects of deploying units of learning that are packaged as services. The purpose of this paper is to review the SOLA concept and present a specific methodology, based on analysing instructional domains, to complement the SOLA approach. The method here proposed addresses the identification and classification of learning services to facilitate the integration of such services.

1. Introduction

The definition of standards and specifications in eLearning aims at improving reusability, portability and interoperability [5]. These qualities should apply both to the learning system and to the learning components used by the system. So far, the standardization process has been primarily focused on defining standards that specify contents and their structure. This strategy is suitable to achieve reusability and portability, but it has shown limited capacity to support interoperability; especially, dynamic semantic interoperability [3]. Dynamic semantic interoperability includes the capability of learning systems and learning components to exchange information and to use the information that has been exchanged without any additional user interaction.

The solution to these shortcomings might lie in service-oriented learning. In this approach, a learning system can be defined as a set of units of learning packaged as services. In order to accomplish interoperability, each learning service is assigned its own functional context (learning objectives) and it is encompassed by a set of capabilities (inputs, outputs and preconditions) related to its educational context. An eLearning platform based on services will be able to dynamically discover and assemble learning components to achieve the desired specific learning purpose [3].

The assembling of services implies the coordination and orchestration of services [4]. Service coordination and orchestration enable services to be strung together following predefined patterns. These patterns establish how services interoperate to achieve the completion of a set of learning objectives. Hence, the development of a learning system based on integrating services needs an analysis process that identifies such patterns in order to know which units of learning should be encapsulated by each learning service. Despite its relevance, there are no current suitable proposals that guide the processes of identification and composition of services.

The rest of the paper is organized as follows. Section 2 reviews the concept of service-orientation and its utility in the design of computer-based learning systems. Section 3 introduces an analysis methodology conceived to support service-oriented learning. Finally, conclusions and a set of recommendations for further work are drawn in the last section.

2. Service-Oriented Learning

The basic idea underlying service-oriented learning is that a learning environment should be conceived as a set of independent units of learning, packaged as services. The design, development and deployment of learning functionalities in the form of services provide instruments to reuse learning components and to establish flexible eLearning platforms [13]. Subsequent sections delve into service-oriented learning approach. Emphasis is put into the definition of the ‘learning-service’ concept and the review of the basic notions of Service-Oriented Learning Architecture (SOLA).
2.1. The concept of learning service

A learning service is usually understood as a supporting activity [14]. Authentication, logging or authorisation are examples of learning services. In the service-oriented paradigm, a service has a different meaning which is related to the system functionality: a service resolves a set of functional goals. Since an instructional process is designed to achieve some educational objectives, we will use here the term learning service in the sense of service-orientation and, therefore, we propose the following definition:

Definition 1: Learning Service. It is a set of learning activities, learning resources and support facilities that are used to satisfy some specific learning objectives.

A learning service is composed by one or many operations. Each operation represents either a learning activity (e.g. question and answer activity) or a support facility (e.g. e-mail). As far as a learning object is concerned, a learning object is conceived as any reusable digital resource that is encapsulated into either a learning service or a message among services. In this approach, there is a relationship between learning services and units of learning (UoL) [6]. Both of them encapsulate a unit of business logic, a delimited piece of education or training. However, they differ in their autonomy. The logic governed by a service resides within an explicit boundary. The service has control within this boundary and is not dependent on other services or tools for executing its functionality. On the other hand, a UoL does not have control over whatever learning content it represents; thus, a UoL is restricted to content modelling with the purpose of facilitating its later reuse. Hence, UoL needs specific management utilities provided with the runtime system [9]. For instance, before executing a UoL it is necessary to map out all the context-specific resources identified at the design stage. The way in which resources are mapped is determined by the runtime system. This restriction reduces the autonomy and interoperability of UoLs.

2.2. Service-Oriented Learning Architecture

The key feature of a service is to provide a specific behaviour that can be applied by different technological platforms [14]. In order to fulfil this requirement, baseline architecture technology is needed, for which the Service-Oriented Architecture (SOA) can be a solution. SOA is a term that represents an architecture style in which automation logic is decomposed into distinct units of logic. Applying this approach to the educational context, Service-Oriented Learning Architecture (SOLA) has emerged. SOLA is defined as “an IT strategy that organizes the discrete functions contained in an educational network into interoperable, standards-based services that can be combined and reused to meet teaching, learning, research and administration needs” [1].

SOLA provides a technological platform to integrate independent learning services. One of the most interesting aspects of SOLA is the ‘layer of services’ concept. A layer of services abstracts a specific operational aspect of a system [4]. In particular, three layers of services are identified:

- Application service layer. Services that reside within this layer can be referred to application components and operations. Authentication or authorisation can be included within this layer.
- Business service layer. While application services are responsible for representing technology and application logic, the business service layer is solely concerned with representing business logic. Units of learning and learning object packaged as services should be integrated in this layer.
- Orchestration service layer. The orchestration service layer consists of one or more services that compose business and application services taking as a reference an instructional process. Course management, sequencing or learning flows can be included within this layer.

According to the SOLA paradigm, the development of computer-supported leaning systems is based on identifying and composing different learning services within the appropriate layer. However, this task can be difficult, particularly if developers are not provided with mechanism to support an identification and composition process which is integrated with the educational requirements of the system. In this context, we propose a framework oriented towards supporting an analysis process that will assist in the definition of a business model known as the instructional domain model. The instructional domain model provides a representation of domain knowledge structures that can be used to classify the learning services as described in the next section.

3. A Service-Oriented Learning Analysis

A learning system is the product of an instructional engineering process [10]. Hence, the development of a service-oriented learning system requires both a technological architecture and a systematic method. In this paper, we focus on methodological approaches and more specifically on one of the earliest phases of the development process: the analysis phase.
Service-oriented analysis focuses on identifying and modelling candidate services. This process can therefore be guided by two main analysis strategies [4]: top-down and bottom-up strategies. The top-down strategy promotes the formal definition of business models prior to modelling service boundaries. The bottom-up strategy is based on the delivery of application services on an ‘as needed’ basis. In this paper we postulate a top-down analysis phase based on domain analysis and feature models. Domain analysis is a technique used in software reuse to identify knowledge structures common to an application domain with the purpose of making such structures reusable [11]. There are different artifacts to represent the domain knowledge like ontologies, taxonomies or domain-specific languages. The method here exposed proposes feature model as a domain knowledge representation. Feature models are particularly useful artifacts to improve reusability. The reason is that reusable components contains inherently more variability that specific components, and feature modelling is the key technique for identifying and capturing such variability [2].

3.1. Instructional Domain Analysis

Instructional Domain Analysis is an analysis approach based on applying domain analysis principles to the instructional design. Our approach postulates the definition of a feature model, which represents the characteristics of the instructional domain, in order to facilitate the identification of candidate services. Prior to explain the analysis process, a set of concepts about instructional domain analysis will be exposed.

3.1.1. Instructional Domain. The first aspect to be defined concerns the domain and its scope. In particular, an instructional domain is defined as:

Definition 2: Instructional Domain. The set of educational scenarios that share contents, an area of work, as well as pedagogical objectives and a specific instructional form. The magnitude of the educational contents considered must make sense by themselves.

The scope of a domain is determined by the set of systems that satisfy a common functionality. In the case of instructional systems, the functionality to be solved is understood as the systematization of the learning process [12]. Given that the learning process depends on the instructional procedure (which is in turn conditioned by the contents, the context and the pedagogical objectives) these factors have been taken as essential for the delimitation of a domain.

3.1.2. Feature modelling. Feature modelling captures the general facilities of an application in a domain in terms of features. In our work, a feature is defined as:

Definition 3: Feature. A distinguishable characteristic of a system that is relevant to the instructional process in a specific instructional domain.

The system characteristics can be categorized in relation to their usefulness in the development process. Especially, and taking into account the features categories defined by the FORM method [8], we have classified the instructional features into four categories:

- Implementation technique. This category represents implementation details used to deploy a computer-supported learning system. It compiles the instructional standards, specifications and services used in the domain.
- Capability. Features that are associated with the operations or functions provided by the system. It corresponds to contents, learning activities and units of learning.
- Operation environment. An operation environment feature represents attributes of the environment in which a system is used. It comprises platforms, tools, course management systems used in the domain.
- Domain technology. This category represents instructional design details. In an educational context, domain technology features compile the characteristics of potential students, instructional theories and the most suitable learning methods to the domain.

We can establish a mapping between the problem space (feature model) and the solution space (layers of services in the SOLA architecture). In particular, we can identify relationships between the implementation technique category and the application layer; the capability features and the business layer, and the operation environment features and the orchestration service layer. The domain technology refers to the instructional process; hence, the domain technology features define the business process that manages the services of the system.

3.2. Instructional Domain Analysis Method

The instructional domain analysis method aims at identifying and modelling candidate services taking as reference a feature model. The analysis process is broken down into two stages: domain analysis and service analysis. The following paragraphs describe the activities that make up each of these stages.
3.2.1. Domain Analysis. The purpose of the domain analysis is to collect relevant domain information and integrate such information into a domain model. The final goal is to specify domain learning commonalities in order to identify learning services. The domain analysis is divided into three activities:

- Domain scoping. A domain is a logic abstraction that groups particular set of courses or areas of knowledge. Thus, the choice of a domain should be driven by the contents and the pedagogical objectives.

- Domain identifying. Once the domain has been scoped, the features of the domain should be identified. Feature identification is guided by ‘feature starter sets’ [2]. Feature starter sets define the collection of features suitable for the domain. In our method, such sets of features correspond to the four categories previously enumerated. For each category, compiled domain knowledge is reviewed, differences and similarities between domain entities are recorded, and specific features are identified.

- Domain modelling. The features of the domain are integrated into a feature model. As shown in Fig. 1, a feature model organizes features into a hierarchical diagram that allows classifying such features as mandatory, alternative and optional [7]. In addition to the feature model, a set of business restrictions rules amongst features should be defined. Such rules define the semantic constraints existing between features that are not gathered in the feature diagram. For instance, we can define a relationship between the instructional method and capability features: problem-based learning is advisable so as to acquire knowledge on code implementation. The formulation of restriction rules is based upon the features involved and the type of restriction between those features. i.e One feature requires another feature.

The domain model is dynamic, that is, it can be refined continuously over its life-cycle by adding new domain knowledge. As more knowledge is added to the model, it becomes more precise and useful

3.2.2. Service Analysis. Service analysis is the process of identifying and classifying candidate learning services. A candidate service corresponds with a logical context. Such logical context is defined by a set of one or more semantic-associated features. This staged is composed by the following steps:

- Service identifying. This activity addresses the identification of candidate service based on the result of the domain modelling. The first task is to select a set of features of the domain model that describes the learning system under development. For the selection of features some guidelines can be considered: (1) First of all, an effective method for identifying such a feature set is based on the four categories previously enumerated, i.e. first considering domain technologies, then capabilities, and finally the operation environment and the implementation techniques; (2) Secondly, restrictions rules set up constraints and optimal selections, i.e. if two features may not be held concurrently the selection of one of them rules out the other. Once the system configuration is defined, the semantic of features is reviewed in order to recognize logical contexts. For example, portfolio assessment and programming exercises are functionally related; thus, an ePortfolio that supports the management of programming exercises may be considered as a candidate learning service.

- Service classifying. The classification of candidate services is guided by the feature categories previously enumerated. A candidate service can be regarded as a logical group of features; however, all features of the group do not have necessarily the same priority or responsibility. A candidate service should be
classified into the service layer associated with its preference features. For instance, an ePortfolio service should be classified into the application layer because its main feature (assessment procedure) belongs to the implementation technique category.

The result of the service analysis process is a classification that represents the candidate learning services grouped by layers. Such classification will be used as starting point to specify the required service contracts. Thus, for instance, and taking as a referent the feature model depicted in Fig. 1, it is possible to identify the need for message and validation services that allow the achievement of peer tutoring activity. The first two services, because of their correspondence to the features of the implementation technique category, should be classified in the application layer; in the same fashion, the peer tutoring activity should be designed as a service embedded in the business layer.

4. Conclusions and further works

The application of service-orientation to instructional design increases the interoperability and flexibility of computer-supported learning systems [3]. According to this idea, an approach for the analysis of service-oriented learning has been presented in this paper. This approach proposes a systematic process to discover learning services taking as a reference an instructional domain model. By examining a family of related domain knowledge, it is possible to obtain a set of reference models expressed in terms of ‘features’. The model that captures the commonalities and differences is called ‘feature model’, and it is used to support both engineering of learning services and the specification of a new computer-supported learning system. Furthermore, the use of a method based on domain analysis establishes the baseline for the definition of a complete service-oriented development method. In this case, the existence of groups of domain features would be related to templates of services.

The approach here presented is being refined and improved through its application to real cases where we are assessing both its utility to analyse learning domains and its usefulness to identify candidate learning services.

5. Acknowledgements

This work is supported by the Moduweb project (TIN2006-09678) funded by the Spanish Ministry of Science and Education (MEC).

6. References


