Validation of the learning ecosystem metamodel using transformation rules

Alicia García-Holgado [a], Francisco José García-Peñalvo [a]

[a] GRIAL Research Group, Computer Sciences Department, Research Institute for Educational Sciences, University of Salamanca, Paseo de Canalejas 169, 37008 Salamanca, Spain {aliciagh, fgarcia}@usal.es

Corresponding author: Alicia García-Holgado, aliciagh@usal.es

Abstract

The learning ecosystem metamodel is a platform-independent model to define learning ecosystems. It is based on the architectural pattern for learning ecosystems. To ensure the quality of the learning ecosystem metamodel is necessary to validate it through a Model-to-Model transformation. Specifically, it is required to verify that the learning ecosystem metamodel allows defining real learning ecosystems based on the architectural pattern. Although this transformation can be done manually, the use of tools to automate the process ensures its validity and minimize the risk of bias. This work describes the validations process composed of eight phases and the results obtained, in particular: the transformation of the MOF metamodel to Ecore to use stable tools for the validation, the definition of a platform-specific metamodel for defining learning ecosystems and the transformation from instances of the learning ecosystem metamodel to instances of the platform-specific metamodel using ATL. A quality framework has been applied to the three metamodels involved in the process to guarantee the quality of the results. Furthermore, some phases have been used to review and improve the learning ecosystem metamodel in Ecore. Finally, the result of the process demonstrates that the learning ecosystem metamodel is valid. Namely, it allows defining models that represent learning ecosystems based on the architectural pattern that can be deployed in real contexts to solve learning and knowledge management problems.

Keywords

Metamodel, Model Driven Development, learning ecosystems, information systems, software engineering, Ecore, software ecosystems, technological ecosystems

1. Introduction

Knowledge management is defined as the process of capturing, storing, sharing, and using knowledge; it allows for finding particular information more efficiently and organize that information for quick retrieval and reuse [1-3]. Nowadays, companies and institutions are focused on finding solutions to support the knowledge management both in their internal and external processes. The adequate management of knowledge, in particular, the teaching and learning processes within a company or an institution, directly influences the improvement of business processes [4, 5]. The evolution from Information Society to the current Knowledge Society has led to change in the knowledge management, the emphasis has moved from knowledge transfer to knowledge creation, from implicit to tacit knowledge, and where building relationships that foster trust and common benefit are the basis for sustainable, ethical progress [6]. The Knowledge Society is a Learning Society, where the learning is the key factor in order to persons, business, regions, and countries achieve success [7].

From a technological point of view, the information management needs in any company or institution are fully covered by traditional information systems, but there is a need to support and improve the knowledge management. In this context, an evolution of the traditional information systems emerges to fulfill the challenges, the technological ecosystems or software ecosystems (SECO) [8, 9]. A technological ecosystem covers the information management needs but also covers the knowledge management needs in any company or institution. According to [10, 11] the institutions adopt a strategy of software ecosystem to expand its organizational boundaries, share its platforms and resources with third parties and define new business models.

The first definition of SECO in the literature is written by Messerschmitt and Szyperski [12], a software ecosystem refers to a collection of software products that have some given degree of symbiotic relationships. Other authors consider that software ecosystems have a relatively closed core software system or technological platform that can be extended with software and services to provide a new functionality [10, 13]. The software ecosystems can be classified according to different characteristics, for example, depending on the license of its components there are proprietary SECO [14] and Open Source SECO [15].

Free Software [16] and Open Source developments [17] are the principal terms used to discuss technological tools to manage the knowledge generated by companies and institutions. This trend is linked to some changes in the technological context. There is a large number of open source tools that allow the

knowledge management in different ways, with special emphasis on content managers and documentary repositories. The present work considers technological ecosystems based on Open Source software components that are connected using web services to support the information flows established among them, combining different programming languages, software, and hardware requirements and heterogeneous users [18]. More concretely, it is focused on learning ecosystems, a kind of technological ecosystem focus on learning management.

The definition, development, and deployment of this type of software solutions is complex and involves several problems identified in previous works [19]. Based on this analysis, an architectural pattern has been defined [18] as an input to define a metamodel to support Model-Driven Development (MDD) of learning ecosystems. The basic idea of a metamodel is to identify the main concepts and their relations of a given problem domain used to describe the models of that domain [20]. There are works on software ecosystem modeling, but most approaches are not supported by a methodology that uses the standards defined by the Object Management Group (OMG). Also, most of them are focused on relationships and collaborations between members of the software ecosystem, including developers [21-23]. The systematic literature reviews by Manikas [10, 24], Barbosa and Alves [25], Pettersson, Svensson, Gil, Andersson and Milrad [22] and Franco-Bedoya, Ameller, Costal and Franch [15] highlight the lack of works in the field of software ecosystem modeling. Franco-Bedoya [15], as other authors [25, 26], affirms that the development of analysis and modeling techniques is one of the main challenges of open source software ecosystems.

The learning ecosystem metamodel [27] is a Platform-Independent Model (PIM) to define learning ecosystems that can be deployed in within any organization, from small and medium enterprises (SMEs) to the biggest corporations, and from training centers to universities. It is based on the architectural pattern for learning ecosystems described in previous works [18, 28, 29]. It has been defined using the Model-Driven Architecture (MDA) proposed by OMG to apply MDD using its standards for visualizing, storing, and exchanging software designs and models. The learning ecosystem metamodel is an M2-model in the four-layer metamodel architecture; it is an instance of the Meta Object Facility (MOF).

To ensure the quality of the learning ecosystem metamodel is necessary to validate it. A preliminary validation has been carried out in previous works through two Model-to-Model (M2M) transformations to test that the metamodel allows defining a real learning ecosystem [30, 31]. To complete the validation process is necessary to verify that the instances of the learning ecosystem metamodel are reciprocated to the deployment of the learning ecosystem in a real context, in other words, it is necessary to transform the

instances of the PIM metamodel in a PSM (Platform-Specific Model) model. To ensure the validity of the process and minimize the risk of bias, the transformations should be done using a tool, not manually.

Although OMG provides several standards to support MDA, there are no stable tools to support the definition and mapping of metamodels and models using those standards. To solve this, the learning ecosystem metamodel has been transformed in an instance of Ecore [32] instead MOF with the purpose of using the tools provided by the Eclipse Modelling Project (EMF). It is set of Eclipse plugins that provide a framework to develop metamodels using Ecore and to support automatic Model-to-Model and Model-to-Text transformation through the definition of transformation rules using ATL transformation languages. Ecore [32] is a meta-metamodel based on MOF focused on being more straightforward and practical. Further, the designers of Ecore have participated in the definition of the core of MOF 2.0, Essential MOF or EMOF, so both are very similar.

The main purpose of this work is to guarantee the quality through the validation of the learning ecosystem metamodel. This paper is an extension of the work presented initially at the WorldCist'18 - 6th World Conference on Information Systems and Technologies [33].

The paper has been divided into eight parts. The second part presents the learning ecosystem metamodel. The third part describes the methodology used to validate the learning ecosystem metamodel in Ecore. The fourth part presents the metamodel in Ecore. The fifth part explains the platform-specific metamodel for defining learning ecosystems based on Open Source software. The sixth part is focused on the transformation from PIM instances to PSM. The seventh part analyses the quality of the proposed metamodels. Finally, the last part summarizes the main conclusions of this work.

2. Learning ecosystem metamodel

A technological ecosystem is composed of a collection of two types of components, on the one hand, software tools and, on the other hand, the human factor represented in different ways, not only as users of the system. People have an essential role in the ecosystems life cycle as well as in the natural ecosystems [34]. Humans are drivers [35], they produce many impacts on natural and technological ecosystems, both harmful and beneficial. Like humans have to work to improve their role as drivers in natural ecosystems, from a technological point of view, the tools and methods to define and develop learning ecosystems have to consider the critical role that people have in their success or failure. Booher [36] stands up for a focus on the human element to achieve both dramatic increases in system performance and productivity and dramatic

reductions in problems in contexts where technology plays a fundamental role. Both Booher [36] and Knodel y Manikas [37] ratify that the human factor must be considered a critical component in complex systems, which supports the need to incorporate this factor as an inherent part of technological ecosystems.

The learning ecosystem metamodel reflects both types of components, as well as the relationships established between them. It was defined in previous works to define Platform-Independent Models of learning ecosystems [27]. The first version of the metamodel is an M2-level model in the OMG four-layer metamodel architecture. It is an instance of MOF, the standard language defined by OMG. The high-level requirements for the learning ecosystem metamodel, defined by the authors in previous work, are the following [27]:

- The metamodel shall enable capture of the high-level description of the learning ecosystem components.
- The metamodel shall enable capture of the human factor as part of the learning ecosystem.
- The metamodel shall enable capture of the information flows between the learning ecosystem components.
- The metamodel shall enable capture of the configurations of the software components.
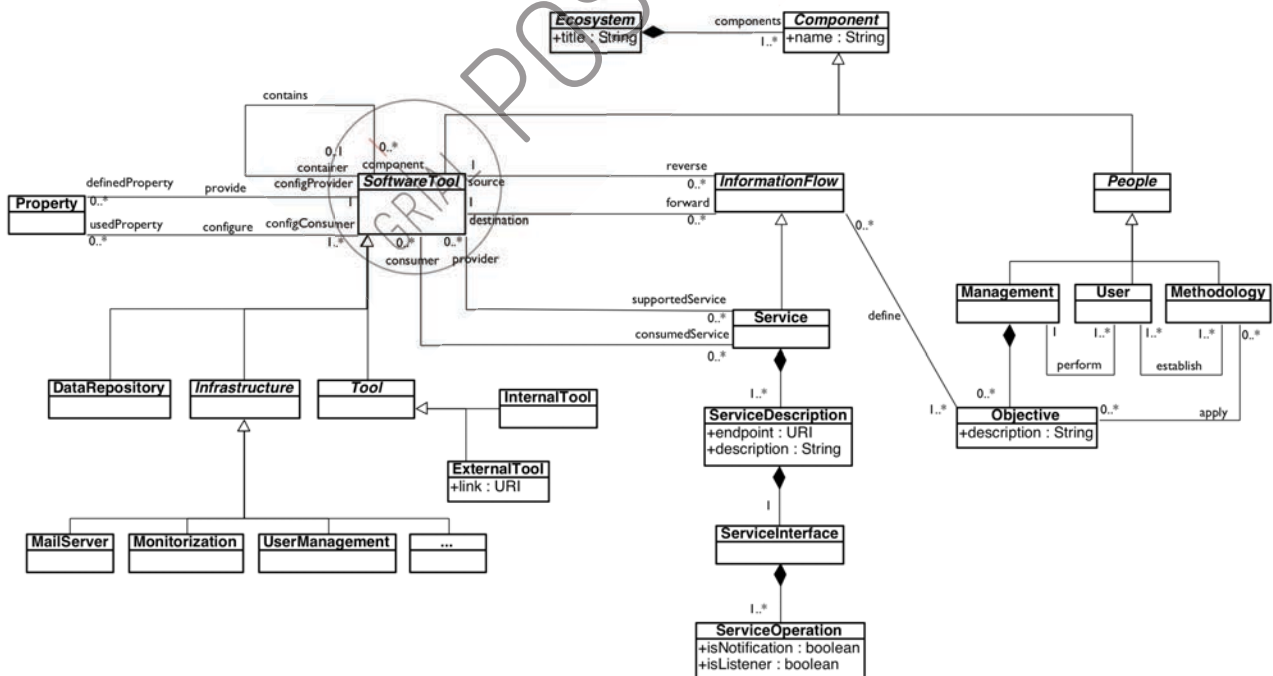

Figure 1. The first version of the learning ecosystem metamodel in MOF [27]

Figure 1 shows the first version of the learning ecosystem metamodel, moreover, a high-resolution version is available at http://doi.org/10.5281/zenodo.829859. The metamodel is completed with four Object Constraint

Language (OCL) constraints included as notes in the diagram [27]. They are used to guarantee the correct instantiation of the metamodel. The main OCL constraint is focused on ensuring the components that always should be part of a learning ecosystem.

3. Methodology

To guarantee the quality of the learning ecosystem metamodel is necessary to validate it. Moreover, the validation process allows for ensuring the robustness and reliability of the metamodel to define learning ecosystems. The process to validate the learning ecosystem metamodel is composed of eight phases (Figure 2). Each phase has an input and gives a result that is used in the next phase. Moreover, some phases are used to review and improve the learning ecosystem metamodel in Ecore.
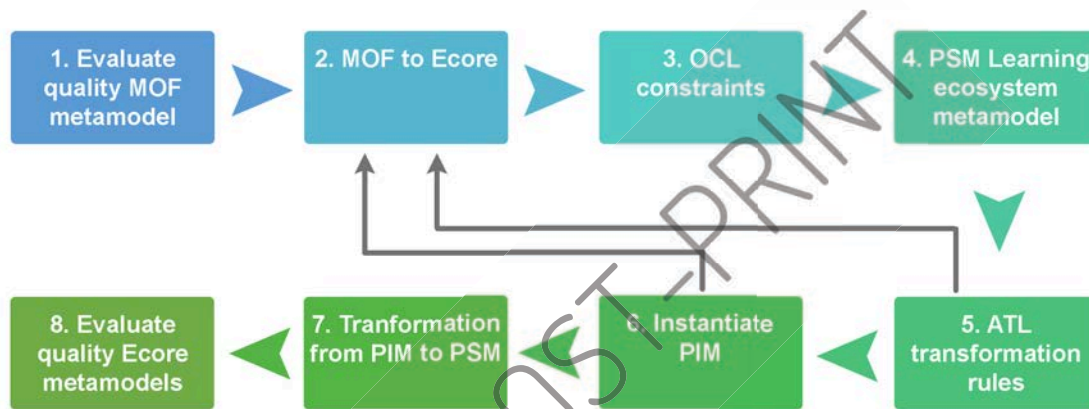

Figure 2. Workflow to validate the learning ecosystem metamodel and quality assurance

The first phase is focused on the quality evaluation of the MOF version of the learning ecosystem metamodel (Figure 1). In particular, the quality has been checked using the metamodel quality framework proposed by López-Fernández, Guerra and de Lara [38]. This framework is composed of thirty features that metamodels should follow. The features are divided into four categories: (1) design, properties signaling a faulty design (an error); (2) best practices, basic design quality guidelines (a warning); (3) naming conventions, questions related to the use of verbs, nouns, etc.); (4) metrics, measurements of metamodel elements and their threshold value [39].

The second phase takes the quality analysis and the MOF version of the metamodel to define the Ecore version. It is a transformation between two M2-level models in the OMG four-layer metamodel architecture defined in MDA, namely an M2M transformation. Although MOF and Ecore support the use of XMI enabling the interchange of models, and model instances through XML based on DTDs/XML schemas generated from

the corresponding models [40], the transformation has been made manually because of several problems with the tool used to define the metamodel in MOF. This one was made with a UML class diagram in Visual Paradigm, and it has not been possible to import it into Eclipse using XML Metadata Interchange (XMI). The instance of Ecore has been made using the Graphical Modelling for Ecore included in EMF. Finally, the Ecore version has been improved to solve the quality problems detected in the previous phase.

Once the version in Ecore is done, the next phase deals with the review and improvement of the OCL constraints. The metamodel instantiated from MOF includes the constraints as text notes in the diagram. Instead, the Ecore version includes the constraints as part of the metamodel using the OCLinEditor provided by EMF. The OCL constraints in an Ecore model are automatically checked when a user tries to instantiate the metamodel.

The fourth phase is focused on the Platform-Specific Model (PSM) for developing learning ecosystems based on Open Source software. A PSM is a model that includes information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform. The learning ecosystem metamodel is a Platform-Independent Model (PIM), it contains no specific information of the platform or the technology that is used to realize it. The platform-specific learning ecosystem metamodel provides the guidelines, software tools and specific technological and human mechanisms to implement the learning ecosystem firstly defined with the learning ecosystem metamodel.

The following phases deal with the transformation of a PIM instance from the learning ecosystem metamodel to an instance of the platform-specific learning ecosystem metamodel. Figure 3 shows the results from the different phases and the relationship between the models, in particular, the transformation from PIM to PSM. The transformation is carried out using a set of rules defined with ATL. Each rule defines a mapping between the elements from the PIM to PSM, namely, from the conceptual elements to specific elements such as Open Source software components, documents, technical details to implement communication mechanism between the components.

The sixth phase is used to instantiate the learning ecosystem metamodel in order to get models of a real ecosystem, in particular, a technological ecosystem for knowledge management in the Spanish Public Administration [41]. The model serves as conceptual map to define the implementation of the technological ecosystem. Moreover, the fifth and sixth phases has served to review the metamodel in Ecore and introduce

some changes in order to get a better metamodel, this feedback is represented in Figure 2 with two lines from these phases to phase 2.

During the seventh phase, the transformation rules take this model as input and supply an instance of the platform-specific learning ecosystem metamodel as output. Finally, the quality of the metamodels in Ecore has been checked, both learning ecosystem metamodel and platform-specific metamodel. It has been used the same metamodel quality framework than in the MOF version.

Three projects have created in EMF as part of the validation process. All the source files are available at a repository in GitHub https://github.com/aliciagh/ecometamodel. In particular, the version used in this work can be accessed through http://doi.org/10.5281/zenodo.1253633 [42].
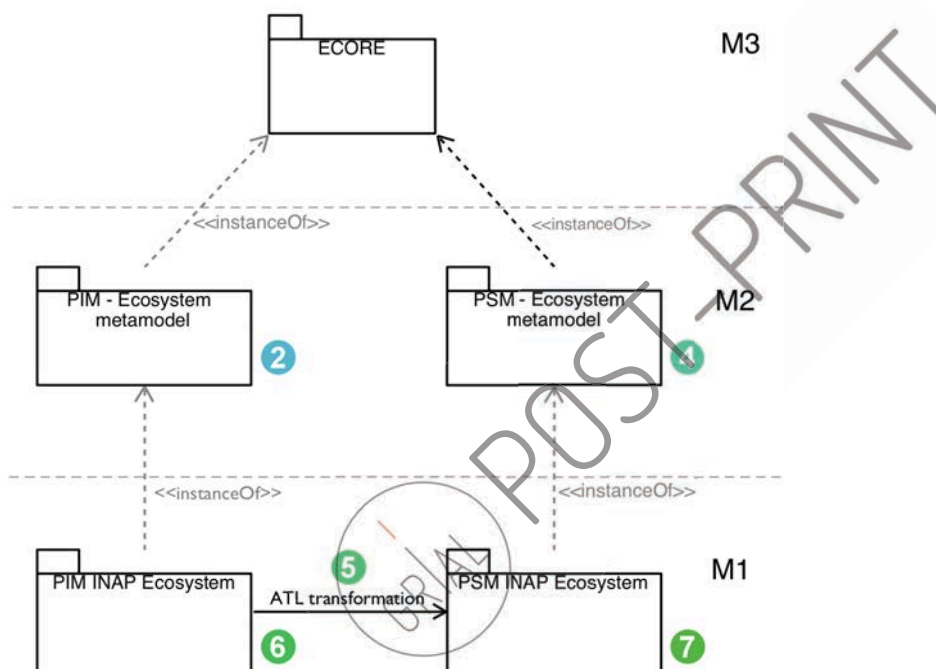

Figure 3. Models and transformations as results of the validation phases, organized in the four-layer metamodel architecture. Model layer (M1), metamodel layer (M2) and meta-metamodel layer (M3).

4. Metamodel in Ecore

Although MDD is supported by OMG through several standards, the stable tools that support MDA are focused on code generation from UML models such as the software tool AndroMDA (https://www.andromda.org). There are no stable tools to support the definition and mapping between metamodels and models using those standards, so the only way to validate the MOF metamodel is manual. On the other hand, the Eclipse community defined Ecore, a meta-metamodel based on MOF, and provides

stable tools to validate the metamodel through the Eclipse Modeling Project. The validation of the learning ecosystem metamodel is carried out using the modeling tools provided by Eclipse in order to minimize the risk of bias introduced by the authors if the process is done manually. The second and third phases of the validation process (Figure 2) are focused on getting the metamodel in Ecore. Regarding the first phase related to the quality evaluation, it is described in the last section together with the final quality evaluation.

First, the mapping between the MOF version of the metamodel to an instance of Ecore, as well as a set of improvements to ensure the quality of the technological ecosystems instantiated from the final version of the metamodel. The transformation process has made manually. This work uses the "MOF" prefix for concepts in MOF and the "E" prefix for concepts in Ecore to prevent confusion.

The main components of the metamodel in MOF are classes (MOFClass), attributes (MOFAttribute) and associations (MOFAssociation). These elements appear in the Ecore metamodel; the classes are represented by an EClass component, the attributes by an EAttribute component and the associations by an EReference component.

The transformation has started with the mapping of each MOFCLass in EClass. Moreover, three new Ecore classes have been included to improve the metamodel. These changes have been a consequence of the feedback from the different phases that compose the validation process (Figure 2), with particular attention to the quality analysis performed on the metamodel in MOF. The new EClass components are described below:

- A new software tool to model indexing tools to improve the knowledge discovery and the search processes in technological ecosystems. This new component, *IndexingService*, is added to the hierarchy as a child of *Infrastructure* EClass.
- A new software tool to replace the MOFClass "…" because ellipsis is forbidden symbols in EClass names. The new EClass is *OtherSystemTool*. It represents other types of software components that are not described in the hierarchy, which allows that the metamodel can evolve.
- A new EClass for modeling the mechanisms to implement the information flows in a technological ecosystem. The *InformationFlow* MOFClass is transformed in an EClass that represents the flows between software tools, either through human interaction or the development of software mechanisms. The new class, *CommunicationMechanism* EClass, models the technological solutions to establish the information flows. It is a root class with two children, one for modeling services and

one for using property files. This new hierarchy facilitates the extension of the communication mechanisms.

After, each MOFAttribute has been mapped in an EAttribute. Regarding the attributes, there are several differences between MOF and Ecore versions of the metamodel, in particular, there is a best practice in Ecore related to the EClass component: each EClass must have a unique identifier attribute. This characteristic is mandatory if the user needs to instantiate the model or apply transformation rules, because of the tool will need to identify each EClass unambiguously. Specifically, an EAttribute *name* or *title* has been added to *InformationFlow*, *CommunicationMechanism*, *ServiceInterface,* and *ServiceOperation*. The other EClasses inherit the identifier from their superclasses.

Moreover, other EAttributes have been included in the metamodel in Ecore based on the feedback got after the transformation phases between instances of the learning ecosystem metamodel to platform-specific instances (Figure 2). In particular, there is some information that should be provided in an instance of an EClass in the PIM in order to decide how to turn it into a particular EClass in the PSM. The new EAttributes have been added to *ExternalTool*, *InternalTool* and *User* EClasses, the classes that represent the main services provided to the users [27]. To *ExternalTool*, two new EAttributes related to the connection between the ecosystem and the external tool (*ExternalTool.id*, *ExternalTool.key*). To *InternatTool*, three new EAttributes to determine some features related to information needs - complexity of the contents (*InternalTool.complexContentType*), use of questionnaires or surveys (*InternalTool.questionnaire*) and use for teaching (*InternalTool.teaching*) - To *User*, a new EAttribute to distinguish his/her role in the institution, specifically, an EAttribute of type *userType*, a new EEnum added to the metamodel.

Finally, the associations between the different classes have been transformed into references. Namely, each MOFAssociation has been mapped in an EReference. This process has been more difficult because in the learning ecosystem metamodel the MOFAssociations have not defined navigability. Instead, Ecore supports uni-directional and bi-directional references, and it is mandatory to define the navigability and a unique name for each EReference. Also, the upper and lower bounds of EReferences have been reviewed, and some changes have been made. First, the lower bound of the EReference *configConsumer* is 0 instead of 1 not to force that all property files (*Property*) are consumed by at least one software tool, and the lower bound of the EReference *establishedMethodology* is 0 instead of 1 not to force all users to establish at least one methodology.

Figure 4 shows the result of the mapping process from MOF to Ecore and the changes made to support the M2M transformations in EMF. The final version of the learning ecosystem metamodel in Ecore is available in high resolution on the following link https://doi.org/10.5281/zenodo.1066369.

Figure 4. Learning ecosystem metamodel in Ecore

During the third phase of the validation process (Figure 2), the OCL constraints are reviewed and included in the metamodel. In particular, eight new OCL constraints have been defined, and two of the previous constraints have been modified.

The main constraint guarantees the components that should be part of a learning ecosystem. These components are defined in the architectural pattern in which the metamodel is based. The constraint ensures that each instance of the metamodel has a mail server, a user management system, and at least one management input stream, one methodology input stream, one user, one internal tool and one monitorization system. The last one, the monitorization system requirement has changed from the first version. The constraint defined in [27] allows only one monitorization tool, but the constraint has been modified to allow more than one. This modification is because sometimes several monitorization tools are part of other components and combined provide the monitorization of the ecosystem.

The second modified constraint is related to the information flows. It ensures that a software tool cannot consume a service provided by itself, that is to say, an information flow always involves two different software tools. In the Ecore version, the technical part of the information flows has been represented by communication mechanisms that encompass both services and properties. The constraint has been modified

to include the properties, namely if a software tool defines a property, this cannot be used by the same software tool.

Regarding the new constraints, there are five to limit the relationships among the components in a learning ecosystem. On the one hand, a software tool cannot be contained itself directly or by transitivity. On the other hand, an external tool cannot contain or be a container of other software tools and a data repository cannot be a component of another software tool.

There are two constraints to ensure that there is at least one information flow between two software tools when these two tools are communicated by a service or a property. The two remaining OCL constraints are those come from the MOF metamodel and remain in the metamodel in Ecore. On the one hand, the constraint to ensure that the point of access to services is unique throughout the system. On the other hand, the restriction for the mail server to at least provide one property. The OCL constraints are available at http://doi.org/10.5281/zenodo.1253633 [42].

5. Platform-Specific Metamodel for learning ecosystems

The learning ecosystem metamodel provides a conceptual guide for defining learning ecosystems but contains no specific information to real technological solutions and human elements to implement the ecosystem. To validate this metamodel is necessary to verify that the instances of the learning ecosystem metamodel are reciprocated to the deployment of the learning ecosystem in a real context. It is necessary to define a PSM in the M2-level of the OMG four-layer metamodel architecture. The fourth phase of the validation process is focused on the definition of this metamodel (Figure 2).

A PSM provides information about the specific technology that is used in the realization of it on a specific platform, and hence possibly contains elements that are specific to the platform. The platform-specific metamodel for defining learning ecosystems provides the guidelines, software tools and specific technological and human mechanisms to implement the learning ecosystem firstly defined with the learning ecosystem metamodel.

The definition of this PSM is based on two pillars. On the one hand, the learning ecosystem metamodel that provides a framework to define specific solutions for the different concepts defined in the metamodel. On the other hand, the experience acquired in the last decade developing technological ecosystems based on Open Source software. Highlight the ecosystem for supporting knowledge management and learning processes in the University of Salamanca [43], the ecosystems for managing informal learning in institutions or companies

developed in the European project "Tagging, Recognition and Acknowledgment of Informal Learning ExpeRiences" (TRAILER) [44, 45] and the ecosystem for knowledge management in the Spanish Public Administration [41, 46].

The high-level requirements for the platform-specific metamodel are based on the requirements of the learning ecosystem metamodel; in particular, both share the first two requirements:

- The metamodel shall enable include of the human factor as part of the learning ecosystem.
- The metamodel shall enable capture of the technological solutions to implement the information flows between the learning ecosystem tools.
- The metamodel shall provide specific software tools to support the infrastructure, the data management and the services that can be part of a learning ecosystem.
- The metamodel must use Open Source solutions.

A learning ecosystem is made up of a collection of three type of components, software tools, documents, and people, represented by the abstract EClasses *SoftwareTool*, *Document*, and *People*, respectively. The software tools are organized in a hierarchical structure that provides the different Open Source tools to implement the infrastructure, the data management and the services of a learning ecosystem. The hierarchy allows the evolution of the metamodel; it is prepared to be extended with other Open Source tools. The metamodel includes the following tools:

- *DSpace* to support document management.
- Infrastructure services for user management (*CASoverLDAP*), monitoring (*Prometheus*), indexing data (*ApacheSolr*) and providing a mail server (*Hakara*).
- A set of services to represent the connection of the learning ecosystem with social networks such as *Twitter* or *Facebook*.
- A set of services to knowledge and learning management: *WordPress*, *Drupal*, *Moodle*, *LimeSurvey*.
- A way to represent the software tools that are part of another software tool through the *Plugin* concept.

The human factor is modeled through the *Document* and *People* EClasses. Regarding people, the users are not modeled in the metamodel, only people that influence directly in the definition and evolution of the learning ecosystems are modeled, specifically, managers (*Manager*) and Information Technology (IT) managers (*ITManager*). These concepts are not synonymous of one person; they can represent a workgroup

or a department. Regarding the documents, two concepts represent the methodology (*MethodologyDocument*) and the management (*ManagementPlan*) required as input in a technological ecosystem according to the architectural pattern proposed by [18]. There are associations between *MethodologyDocument* and *People* to model who establishes the methodology, and between *ManagementPlan* and *People* to indicate who performs the management.

Finally, the third main element in a learning ecosystem, the relationship between the components, it is modeled by the *CommunicationSolution* and *Dependency* EClasses. From a technological point of view, the information flows between components are modeled by a communication solution, in this case, the *CommunicationSolution* is an abstract EClass with two children, one that represents the information flow through files (*File*), and another one using RESTful web services (*RESTfulAPI*). Like other parts of the metamodel, it is prepared to be extended, to evolve, other ways to implement communication between two software tools can be added, for example, SOAP [47]. The aim of the metamodels described in this work is not to provide a detailed solution to model web services; there are several authors in the literature that provide suitable solutions to model that part of the ecosystem [47-49].

Furthermore, there is information flows between components that are not established by a technological solution, but by a person. To model this concept, the EClass *Dependency* is used to document the relationships between components in the learning ecosystem.

Figure 5 shows the final version of the PSM in Ecore. Moreover, a high-resolution version is available on the following link http://doi.org/10.5281/zenodo.1252185.
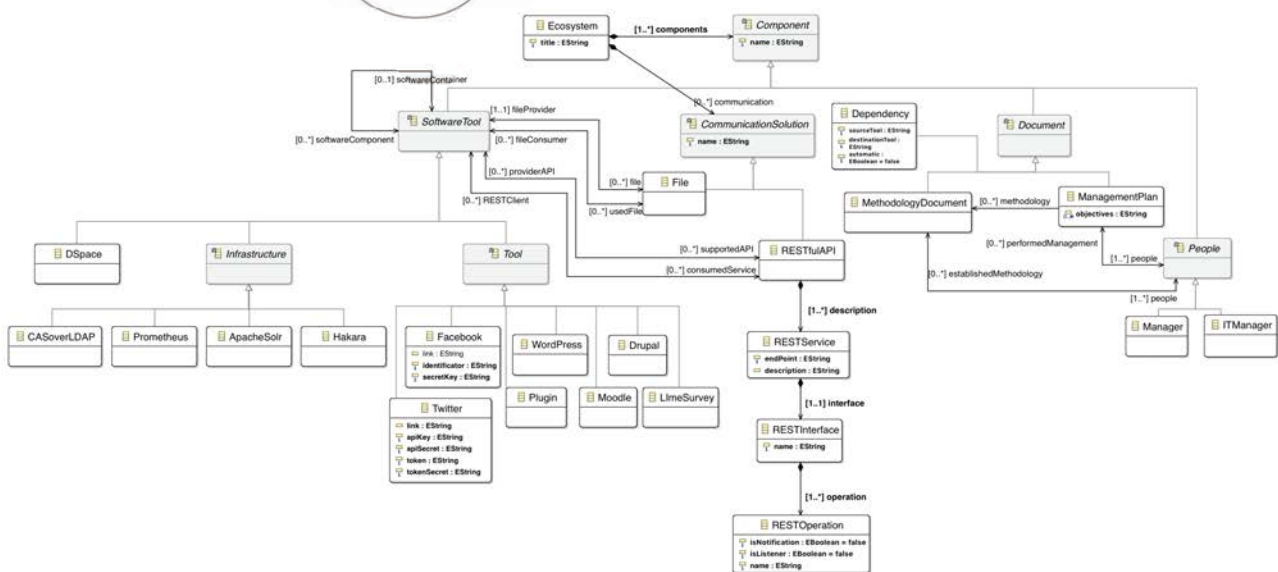

Figure 5. Platform-Specific Metamodel for defining learning ecosystems based on Open Source software

The metamodel proposed in Figure 5 is completed with a set of constraints defined with OCL and included in the Ecore file available at http://doi.org/10.5281/zenodo.1253633 [42]. Five constraints have been defined to guarantee the correct instantiation of the platform-specific metamodel for defining learning ecosystems. Highlight the constraint to ensure that a *Plugin* is contained in another software tool and it does not contain other tools. The rest of the constraints are similar to those mentioned in the platform-independent metamodel.

6. Transformation from PIM to PSM

The validation process of the learning ecosystem metamodel implies a transformation from instances of PIM to PSM, that is to say, from the concepts defined in the learning ecosystem metamodel to specific solutions provided by the metamodel describe in the previous section. This transformation allows verifying that the instances of the learning ecosystem metamodel are reciprocated to the deployment of the learning ecosystem in a real context. Although this transformation can be done manually, the use of an automatic process ensures its validity and minimize the risk of bias.

The transformation is carried out using a set of rules defined with ATL. ATL is a language for expressing model transformations; it is a hybrid language, a mix of declarative and imperative constructions designed to express model transformation as required by any MDA approach [50].

The ATL transformation available at [42] provides a formal and detailed definition of the translation from an instantiated model of the learning ecosystem metamodel to platform-specific solutions. Table 2, Table 3 and Table 4 depict the rationale that underlies the transformations from PIM to PSM.

Basically, in the transformation related to the software tools (Table 2), each concept is transformed into Open Source tools, for example, data repository into DSpace (http://www.duraspace.org/dspace), monitoring system into Prometheus (https://prometheus.io), or user manager tool into a combination of CAS (https://www.apereo.org/projects/cas) and OpenLDAP (https://www.openldap.org). Regarding the internal tools, there are a set of boolean attributes used to transform them into different solutions (Table 1). In particular, the attributes indicate the complexity of the contents (*InternalTool.complexContentType*), the need for questionnaires (*InternalTool.questionnaire*) or surveys and if it will be mainly used for teaching (*InternalTool.teaching*).

Table 1. Value of each attribute in the learning ecosystem metamodel to transform an internal tool into one of the four available Open Source tools

Open Source tool	*complexContentType*	*questionnaire*	*teaching*
Moodle	-	-	true
LimeSurvey	false	true	false
WordPress	false	false	false
Drupal	true	-	false

Table 2. Software tools transformation rationale

PIM (learning ecosystem metamodel)	PSM (PSM metamodel for learning ecosystems)
Software tools	
Ecosystem	Ecosystem
DataRepository	DSpace
MailServer	Hakara
Monitorization	Prometheus
UserManager	CASoverLDAP
IndexingService	ApacheSolr
InternalTool	Moodle
	LimeSurvey
	WordPress
	Drupal
ExternalTool	Facebook
	Twitter
SoftwareTool	Plugin

Table 3. People transformation rationale

PIM (learning ecosystem metamodel)	PSM (PSM metamodel for learning ecosystems)
People	
User	Manager
	ITManager
Management	ManagementPlan
Methodology	MethodologyDocument
Objective	ManagementPlan.objectives

Regarding the transformation related to the human factor as part of an ecosystem (Table 3), each user is transformed into a manager or an IT manager depending on the *User.type* attribute, each management concept into a management plan, each methodology into a document with the methodology, and the objectives into an attribute of the management plan.

Finally, regarding the concepts related to the information flows, each information flow is transformed into a dependency, each property into a file, each service into a RESTful API composed by a set of REST services, interfaces, and operations (Table 4).

Table 4. Information flows transformation rationale

PIM (learning ecosystem metamodel)	PSM (PSM metamodel for learning ecosystems)
Information flows	
InformationFlow	Dependency
Property	File
Service	RESTfulAPI
ServiceDescription	RESTService
ServiceInterface	RESTInterface
ServiceOperation	RESTOperation

To complete the validation process is necessary to verify that the learning ecosystem metamodel allows defining real learning ecosystems in keeping with the architectural pattern defined and tested in previous works [18].

Several learning ecosystems have been developed from the architectural pattern not only within educational institutions but also in companies and public administration. Highlight the learning ecosystem for scientific knowledge management in Ph.D. Programmes [34, 51], the learning ecosystems for knowledge management in the Spanish Public Administration [18, 30, 41, 46] and the WYRED ecosystem for supporting social dialogues and research processes [52, 53].

In order to validate the metamodel, the instances of the learning ecosystem metamodel should be reciprocated to the deployment of the learning ecosystem in a real context. The sixth and seventh phases of the validation process are focused on carrying out an M2M transformation using the tools provided by EMF.

First, the learning ecosystem metamodel has been instantiated to model a reduced version of one of the real ecosystems described above, in particular, the ecosystem for Spanish Public Administration or INAP Ecosystem. The instance includes all the software tools and human factors of the ecosystem, but it does not include all the information flows, only a service to implement single-sign-on, a property related to the configuration of the mail server and the associated information flows. The model does not include all services, properties and information flows to simplify the case study and focus on the validation goal. Figure 6 shows the instance.

Then, the ATL transformation takes the instantiated model as input and supplies an instance of the platform-specific learning ecosystem metamodel as output. The result is shown in the right part of Figure 6. The model matches with the implementation of the ecosystem. Moreover, both models comply with the OCL constraints defined as part of the corresponding metamodels.

This example verifies that the learning ecosystem metamodel allows defining a model of a real learning ecosystem.
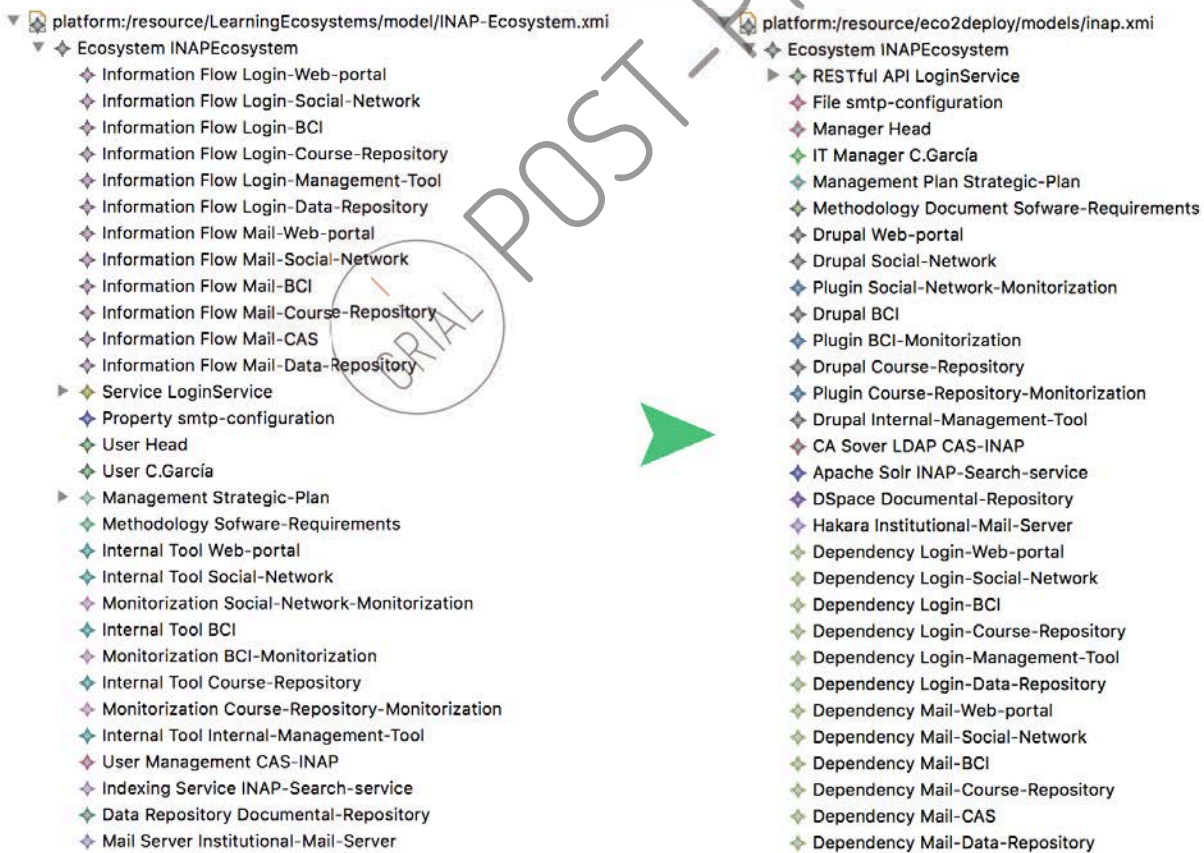
Figure 6. Instance of the learning ecosystem metamodel depicted in Figure 4 (left) and transformation result (right)

7. Quality of the metamodels

The validation process has two phases destined to evaluate the quality of the metamodels. Specifically, the first phase is focused on the quality evaluation of the MOF version of the learning ecosystem metamodel; and the last phase ensures the quality of the two metamodels defined during the validation, the Ecore version of the learning ecosystem metamodel and the platform-specific metamodel for defining learning ecosystems based on Open Source software. All quality evaluations have checked according to the metamodel quality framework proposed by [38]. They propose a set of thirty features that correspond mainly to syntactic rules that metamodels should follow (Table 5). Their features are categorized in design flaws, best practices, naming conventions and metrics.

Table 5. Features of the metamodel quality framework [38]

	Design
D01	An attribute is not repeated among all specific classes of a hierarchy.
D02	There are no isolated classes (i.e., not involved in any association or hierarchy).
D03	No abstract class is super to only one class (it nullifies the usefulness of the abstract class).
D04	There are no composition cycles.
D05	There are no irrelevant classes (i.e., abstract and subclass of a concrete class).
D06	No binary association is composite in both member ends.
D07	There are no overridden, inherited attributes.
D08	Every feature has a maximum multiplicity greater than 0.
D09	No class can be contained in two classes, when it is compulsorily in one of them.
D10	No class contains one of its superclasses, with cardinality 1 in the composition end (this is not finitely satisfiable).
	Best practices
BP01	There are no redundant generalization paths.
BP02	There are no uninstantiable classes (i.e., abstract without concrete children).
BP03	There is a root class that contains all others (best practice in EMF).
BP04	No class can be contained in two classes (weaker version of property D09).
BP05	A concrete top class with subclasses is not involved in any association (the class should be probably abstract).
BP06	Two classes do not refer to each other with non-opposite references (they are likely opposite).
	Naming conventions
N01	Attributes are not named after their feature class (e.g., an attribute paperID in class Paper).
N02	Attributes are not potential associations. If the attribute name is equal to a class, it is likely that what the designer intends to model is an association.
N03	Every binary association is named with a verb phrase.
N04	Every class is named in pascal-case, with a singular-head noun phrase.
N05	Element names are not too complex to process (i.e., too long).
N06	Every feature is named in camel-case.
N07	Every non-boolean attribute has a noun-phrase name.
N08	Every boolean attribute has a verb-phrase (e.g., isUnique).

N09	No class is named with a synonym to another class name.
Metrics	
M01	No class is overloaded with attributes (10-max by default)
M02	No class refers to too many others (5-max by default) – a.k.a. efferent couplings (Ce).
M03	No class is referred from too many others (5-max by default) – a.k.a. afferent couplings (Ca).
M04	No hierarchy is too deep (5-level max by default) – a.k.a. depth of inheritance tree (DIT).
M05	No class has too many direct children (10-max by default) - a.k.a. number of children (NOC).

The first version of the metamodel did not comply with the D03 (*No abstract class is super to only one class*) and BP03 (*There is a root class that contains all others*) features. The MOF version of the metamodel has an abstract class, *InformationFlow*, that was a superclass of only one class, *Service*. In the Ecore version of the metamodel, in order to comply the feature D03, the *Property* class has been included in the hierarchy of *InformationFlow*. Furthermore, the *InformationFlow* class has been divided into two classes, one with the same name that represents the communication between two tools and another one named *CommunicationMechanism* to describe the software mechanism used to establish that communication in case there was.

Regarding the BP03 feature, there is a class in the metamodel in MOF, *Ecosystem*, that contains all classes except two, *Property* and *InformationFlow*. The Ecore version of the metamodel has two new composition associations, one between the root class and *InformationFlow*, and other between the root class and the new class *CommunicationMechanism*.

The learning ecosystem metamodel instantiated from Ecore complies with the thirty features that compose the framework. Highlight the metrics:

- M01. The maximum number of attributes in a class of the metamodel is 4.

- M02. The classes with more references to others are *InformationFlow*, *SoftwareTool* and *Ecosystem* with a Ce value of 3.

- M03. The classes more referred from others are *InformationFlow* with a Ca value of 4, and *SoftwareTool* and *Objective* with a Ca value of 3.

- M04. The deepest hierarchy has a DIT value of 4, where the root class is *Component*.

- M05. The class with more children is *Infrastructure* with a NOC value of 5.

Also, the platform-specific metamodel defined as part of the validation process complies with the thirty features. The associated metrics for this metamodel are:

- M01. The maximum number of attributes in a class is 4.

- M02. The classes with a higher Ce are *Ecosystem*, *SoftwareTool*, *ManagementPlan* and *People* with a value of 2.

- M03. The classes with a higher Ca are *SoftwareTool*, *Methodology* and *People* with a value of 2.

- M04. The deepest hierarchy has a DIT value of 4, where the root class is *Component*.

- M05. The class with more children is *Tool* with a NOC value of 6.

Therefore, according to this quality framework, the metamodels defined in Ecore, both the PIM and the PSM, meet all the quality criteria.

8. Conclusions

The definition, development, and deployment of technological ecosystems and, specifically, learning ecosystems is complex and involves several problems identified in previous works [18]. The learning ecosystem metamodel aims to provide a framework to support Model-Driven Development of learning ecosystems within any organization, not only educational contexts but also companies or public administrations.

The first version of the metamodel was defined as a Platform-Independent Model using MOF. To ensure the robustness and reliability of the metamodel to define learning ecosystems it is necessary to validate it. In previous works a preliminary validation has been carried out; two M2M transformations have been made to test that the metamodel allows defining real learning ecosystems [30, 31]. These preliminary validations have been made manually because there are no stable tools that support the standards defined by OMG. For this reason, the metamodel has been redefined using Ecore in order to utilize the modeling tools provided by Eclipse.

The validation process composed of eight phases, not only covers the validation but also guarantee the quality of the metamodel. On the one hand, a metamodel quality framework has been applied to the three metamodels involved in the process. On the other hand, some phases have been used to review and improve the learning ecosystem metamodel in Ecore.

There are several results associated with the validation. First, the Ecore version of the learning ecosystem metamodel. Second, the definition of a platform-specific metamodel for defining learning ecosystems, which provides a set of recommended tools to implement learning ecosystems based on Open Source software.

Finally, the ATL transformation used to transform instances of the learning ecosystem metamodel into instances of the platform-specific metamodel.

The metamodel allows defining models that correspond to real learning ecosystems based on the architectural pattern mentioned above. All elements of the model resulting from the application of the ATL transformation match with the implementation of the real ecosystem selected as an example. Moreover, the constraints defined in both metamodels have complied.

The platform-specific metamodel for defining learning ecosystems based on Open Source software is not the only way to develop learning ecosystems. The learning ecosystem metamodel can be transformed in other PSM, for example, one that combines Open Source and proprietary software. The only requirement is to define the PSM and the corresponding transformation rules.

It would be interesting to carry out more case studies with other real ecosystems to confirm the validation process. Moreover, future works could be oriented to define other platform-specific metamodels to extend the application of the learning ecosystem metamodel.

Acknowledgement

References

[1] J.-N. Lee, The impact of knowledge sharing, organizational capability and partnership quality on IS outsourcing success, Information & Management, 38 (2001) 323-335.

[2] C.P. Lee, G.G. Lee, H.F. Lin, The role of organizational capabilities in successful e-business implementation, Business Process Management Journal, 13 (2007) 677-693.

[3] Á. Fidalgo-Blanco, M.L. Sein-Echaluce, F.J. García-Peñalvo, Epistemological and ontological spirals: From individual experience in educational innovation to the organisational knowledge in the university sector, Program: Electronic library and information systems, 49 (2015) 266-288.

[4] P. Quintas, P. Lefrere, G. Jones, Knowledge management: A strategic agenda, Long Range Planning, 30 (1997) 385-391.

[5] A. Tiwana, The knowledge management toolkit: practical techniques for building a knowledge management system, Prentice Hall PTR, 2000.

[6] E. Rubio Royo, S. Cranfield McKay, J.C. Nelson-Santana, R.N. Delgado Rodríguez, A.A. Ocon-Carreras, Web Knowledge Turbine as a Proposal for Personal and Professional Self-Organisation in Complex Times: Application to Higher Education, Journal of Information Technology Research (JITR), 11 (2018) 70-90.

[7] OECD, Knowledge Management in the Learning Society, OECD Publishing, Paris, 2000.

[8] K.C. Laudon, J.P. Laudon, Essentials of Management Information Systems: Transforming Business and Management, Prentice Hall, Upper Saddle River, NJ, 1991.

[9] B. Langefors, Information systems theory, Information Systems, 2 (1977) 207-219.

[10] K. Manikas, K.M. Hansen, Software ecosystems – A systematic literature review, Journal of Systems and Software, 86 (2013) 1294-1306.

[11] T. Weiblen, A. Giessmann, A. Bonakdar, U. Eisert, Leveraging the software ecosystem:Towards a business model framework for marketplaces, in: Proceedings of the International Conference on Data Communication Networking, e-Business and Optical Communication Systems, SciTePress, 2012, pp. 187-193.

[12] D.G. Messerschmitt, C. Szyperski, Software ecosystem: understanding an indispensable technology and industry, MIT Press Books, 1 (2005).

[13] S. Jansen, A. Finkelstein, S. Brinkkemper, A Sense of Community: A Research Agenda for Software Ecosystems, in: 31st International Conference on Software Engineering - Companion Volume, 2009. ICSE-Companion 2009. Vancouver, BC, 16-24 May 2009, IEEE, USA, 2009, pp. 187-190.

[14] K. Manikas, Supporting the Evolution of Research in Software Ecosystems: Reviewing the Empirical Literature, in: A. Maglyas, A.L. Lamprecht (Eds.) Software Business. Lecture Notes in Business Information Processing, Springer, Cham, 2016, pp. 63-78.

[15] O. Franco-Bedoya, D. Ameller, D. Costal, X. Franch, Open source software ecosystems: A Systematic mapping, Information and Software Technology, 91 (2017) 160-185.

[16] R.M. Stallman, GNU Emacs Manual, Free Software Foundation, El proyecto GNU–Fundación para el software libre, (1986).

[17] Open Source Initiative, Open Source Definition, in, 2008.

[18] A. García-Holgado, F.J. García-Peñalvo, Architectural pattern to improve the definition and implementation of eLearning ecosystems, Science of Computer Programming, 129 (2016) 20-34.

[19] A. García-Holgado, F.J. García-Peñalvo, The evolution of the technological ecosystems: an architectural proposal to enhancing learning processes, in: Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality (TEEM'13) (Salamanca, Spain, November 14-15, 2013), ACM, New York, 2013, pp. 565-571.

[20] J.J. Gómez-Sanz, J. Pavón, Meta-modelling in agent oriented software engineering, in: F.J. Garrijo, J.-C. Riquelme, M. Toro (Eds.) Ibero-American Conference on Artificial Intelligence 2002, Springer, Heidelberg, 2002, pp. 606-615.

[21] I. Van den Berk, S. Jansen, L. Luinenburg, Software ecosystems: a software ecosystem strategy assessment model, in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, Copenhagen, Denmark, 2010, pp. 127-134.

[22] O. Pettersson, M. Svensson, D. Gil, J. Andersson, M. Milrad, On the role of software process modeling in software ecosystem design, in: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ACM, New York, NY, USA, 2010, pp. 103-110.

[23] I.T. Hawryszkiewycz, A Metamodel for Modeling Collaborative Systems, Journal of Computer Information Systems, 45 (2005) 63-72.

[24] K. Manikas, Revisiting software ecosystems Research: A longitudinal literature study, Journal of Systems and Software, 117 (2016) 84-103.

[25] O. Barbosa, C. Alves, A systematic mapping study on software ecosystems, in: 3rd International Workshop on Software Ecosystems 2011, IWSECO 2011, CEUR-WS, Brussels, Belgium, 2011, pp. 15-26.

[26] M.H. Sadi, E. Yu, Designing software ecosystems: How can modeling techniques help?, in: K. Gaaloul, R. Schmidt, S. Nurcan, S. Guerreiro, Q. Ma (Eds.) Enterprise, Business-Process and Information Systems Modeling. BPMDS 2015, EMMSAD 2015. Lecture Notes in Business Information Processing, Springer, Cham, 2015, pp. 360-375.

[27] A. García-Holgado, F.J. García-Peñalvo, A Metamodel Proposal for Developing Learning Ecosystems, in: P. Zaphiris, A. Ioannou (Eds.) Learning and Collaboration Technologies. Novel Learning Ecosystems. 4th International Conference, LCT 2017. Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017. Proceedings, Part I, Springer International Publishing, Switzerland, 2017, pp. 100-109.

[28] A. García-Holgado, F.J. García-Peñalvo, Architectural pattern for the definition of eLearning ecosystems based on Open Source developments, in: J.L. Sierra-Rodríguez, J.M. Dodero-Beardo, D. Burgos (Eds.) Proceedings of 2014 International Symposium on Computers in Education (SIIE) (Logroño, La Rioja, Spain, November 12-14, 2014), Institute of Electrical and Electronics Engineers. IEEE Catalog Number CFP1486T-ART, 2014, pp. 93-98.

[29] A. García-Holgado, F.J. García-Peñalvo, Á. Hernández-García, F. Llorens-Largo, Analysis and Improvement of Knowledge Management Processes in Organizations Using the Business Process Model Notation, in: D. Palacios-Marqués, D.R. Soriano, K.H. Huarng (Eds.) New Information and Communication Technologies for Knowledge Management in Organizations: 5th Global Innovation and Knowledge Academy Conference, GIKA 2015, Valencia, Spain, July 14-16, 2015, Proceedings, Springer International Publishing, Switzerland, 2015, pp. 93-101.

[30] A. García-Holgado, F.J. García-Peñalvo, Preliminary validation of the metamodel for developing learning ecosystems, in: J.M. Dodero, M.S. Ibarra Sáiz, I. Ruiz Rube (Eds.) Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM 2017) (Cádiz, Spain, October 18-20, 2017), ACM, New York, NY, USA, 2017.

[31] A. García-Holgado, F.J. García-Peñalvo, Definición de ecosistemas de aprendizaje independientes de plataforma, in: IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2017) (Zaragoza, Spain, October 4-6, 2017), 2017, pp. 668-673.

[32] Eclipse Foundation, Description of the interface for the Ecore dialect of UML, in, Eclipse Foundation, 2015.

[33] A. García-Holgado, F.J. García-Peñalvo, Learning Ecosystem Metamodel Quality Assurance, in: Á. Rocha, A. Correia, H. Adeli, L. Reis, S. Costanzo (Eds.) Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing, Springer International Publishing, Switzerland, 2018, pp. 787-796.

[34] A. García-Holgado, F.J. García-Peñalvo, Human interaction in learning ecosystems based in Open Source solutions, in: Learning and Collaboration Technologies. Design, Development and Technological Innovation. 5th International Conference, LCT 2018. Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Springer International Publishing, Switzerland, 2018.

[35] K. Chopra, R. Leemans, P. Kumar, H. Simons, Ecosystems and human well-being: policy responses, in, Island Press, 2005.

[36] H.R. Booher, Introduction: Human Systems Integration, in: H.R. Booher (Ed.) Handbook of Human Systems Integration, John Wiley & Sons Inc., Hoboken, NJ, USA, 2003, pp. 1-30.

[37] J. Knodel, K. Manikas, Towards a Typification of Software Ecosystems, in: J. Fernandes, R. Machado, K. Wnuk (Eds.) Software Business. ICSOB 2015. Lecture Notes in Business Information Processing, Springer, Cham, 2015, pp. 60-65.

[38] J.J. López-Fernández, E. Guerra, J. de Lara, Assessing the Quality of Meta-models, in: F. Boulanger, M. Famelis, D. Ratiu (Eds.) MoDeVVa, CEUR Workshop Proceedings, Valencia, Spain, 2014, pp. 3-22.

[39] S.R. Chidamber, C.F. Kemerer, A metrics suite for object oriented design, IEEE Transactions on Software Engineering, 20 (1994) 476-493.

[40] A. Gerber, K. Raymond, MOF to EMF: there and back again, in: Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange, ACM, 2003, pp. 60-64.

[41] A. García-Holgado, J. Cruz-Benito, F.J. García-Peñalvo, Analysis of Knowledge Management Experiences in Spanish Public Administration, in: G.R. Alves, M.C. Felgueiras (Eds.) Proceedings of the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15) (Porto, Portugal, October 7-9, 2015), ACM, New York, NY, USA, 2015, pp. 189-193.

[42] A. García-Holgado, F.J. García-Peñalvo, Code repository that supports the research presented in the paper "Validation of the learning ecosystem metamodel using transformation rules", in, 2018.

[43] F.J. García-Peñalvo, Docencia, in: J. Laviña Orueta, L. Mengual Pavón (Eds.) Libro blanco de la Universidad digital 2010, Ariel, Barcelona, 2008, pp. 29-61.

[44] F.J. García-Peñalvo, M.Á. Conde-González, V. Zangrando, A. García-Holgado, A.M. Seoane-Pardo, M. Alier, N. Galanis, F. Brouns, H. Vogten, D. Griffiths, A. Mykowska, G.R. Alves, M. Minović, TRAILER project (Tagging, recognition, acknowledgment of informal learning experiences). A Methodology to make visible learners' informal learning activities to the institutions, Journal of Universal Computer Science, 19 (2013) 1661-1683.

[45] F.J. García-Peñalvo, M. Johnson, G.R. Alves, M. Minović, M.Á. Conde-González, Informal learning recognition through a cloud ecosystem, Future Generation Computer Systems, 32 (2014) 282-294.

[46] A. García-Holgado, F.J. García-Peñalvo, Knowledge management ecosystem based on drupal platform for promoting the collaboration between public administrations, in: F.J. García-Peñalvo (Ed.) Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'14) (Salamanca, Spain, October 1-3, 2014), ACM, New York, NY, USA, 2014, pp. 619-624.

[47] A. Navarro, A. da Silva, A metamodel-based definition of a conversion mechanism between SOAP and RESTful web services, Comput. Stand. Interfaces, 48 (2016) 49-70.

[48] H. Jegadeesan, S. Balasubramaniam, An MOF2-based Services Metamodel, Journal of Object Technology, 7 (2008) 71-96.

[49] B. Simon, B. Goldschmidt, K. Kondorosi, A Metamodel for the Web Services Standards, Journal of Grid Computing, 11 (2013) 735-752.

[50] F. Allilaire, T. Idrissi, ADT: Eclipse development tools for ATL, in: Proceedings of the Second European Workshop on Model Driven Architecture (MDA) with an emphasis on Methodologies and Transformations (EWMDA-2), 2004, pp. 171-178.

[51] A. García-Holgado, F.J. García-Peñalvo, M.J. Rodríguez-Conde, Definition of a Technological Ecosystem for Scientific Knowledge Management in a PhD Programme, in: G.R. Alves, M.C. Felgueiras (Eds.) Proceedings of the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15) (Porto, Portugal, October 7-9, 2015), ACM, New York, NY, USA, 2015, pp. 695-700.

[52] F.J. García-Peñalvo, The WYRED Project: A Technological Platform for a Generative Research and Dialogue about Youth Perspectives and Interests in Digital Society, Journal of Information Technology Research, 9 (2016) vi-x.

[53] F.J. García-Peñalvo, A. García-Holgado, A. Vázquez-Ingelmo, A.M. Seoane-Pardo, Usability test of WYRED Platform, in: Learning and Collaboration Technologies. Design, Development and Technological Innovation. 5th International Conference, LCT 2018. Held as Part of HCI International 2018, Las Vegas, NV, USA, July 15–20, 2018, Springer International Publishing, Switzerland, 2018.

Vitae

Alicia García-Holgado hold[...]ences and an M.Sc. in Intelligent Systems [...]anca, Spain. She is currently pursuir[...] Knowledge Society at the University [...] the definition and development of techn[...]ing management in various contexts. [...] Research Group of the University of [...] She has participated in many national and international R&D projects; [...] she has more than 50 international publications both in journals and scientific conferenc[...] ISI-JCR and SCOPUS.

[...]sco José García-Peñalvo received the degrees in computing from [...]e University of Salamanca and the University of Valladolid, and a Ph.D. from the University of Salamanca. He is currently the Head of the GRIAL Research Group. He has led and participated in over 50 research and innovation projects. He was the Vice Chancellor of Innovation with the University of Salamanca from 2007 to 2009. His main research interests focus on eLearning, computers and education, adaptive systems, Web engineering, semantic Web, and software reuse. He is currently the Editor-in-Chief of the International Journal of Information Technology Research and the Education in the Knowledge Society Journal.