García-Holgado, A., & García-Peñalvo, F. J. (2017). Preliminary validation of the metamodel for developing learning ecosystems. In J. M. Dodero, M. S. Ibarra Sáiz, & I. Ruiz Rube (Eds.), *Fifth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'17) (Cádiz, Spain, October 18-20, 2017)* (Article 91). New York, NY, USA: ACM. doi:10.1145/3144826.3145439

Preliminary validation of the metamodel for developing learning ecosystems

Alicia García-Holgado
GRIAL Research Group,
Research Institute for Educational Sciences, University of Salamanca
37008 Salamanca, Spain
aliciagh@usal.es

Francisco J. García-Peñalvo
GRIAL Research Group,
Research Institute for Educational Sciences, University of Salamanca
37008 Salamanca, Spain
fgarcia@usal.es

ABSTRACT

Learning ecosystems are positioned as a technological solution able to provide learning environments that evolve over time to cover the changing needs of users or include new tools for knowledge management. The definition, development and deployment of this type of software solutions is complex and involves several problems identified and analyzed in previous works. In order to solve these problems, an architectural pattern based on Buschmann's Layers pattern has been defined. The accumulated experience developing learning ecosystems along with the problems analysis using Business Process Model and Notation and the architectural pattern, have provided a knowledge base to define an ecosystems metamodel. The proposed metamodel provides a framework for Model-Driven Development of learning ecosystems. The main objective of the present work is to validate the ecosystems metamodel from modeling a learning ecosystem.

CCS CONCEPTS

• Information systems → Open source software • Software and its engineering → Software design engineering • Software and its engineering → Software development techniques

KEYWORDS

Technological ecosystems, software ecosystems, MDA, modeling, software engineering, knowledge management, Open Source.

1 INTRODUCTION

Technological ecosystems, also called software ecosystems (SECO) in the literature, are the evolution of the traditional information systems with two main differences. First, the technological ecosystems should have the ability to evolve in different dimensions [1]: (1) each software component evolves individually, so updating each component should be possible; (2) software components can be replaced or deleted; (3) the organization's or institution's needs evolve throughout time so the architecture of the ecosystem will require changes; (4) according to Alspaugh et al. [2] there is a fourth mechanism of evolution related to the licenses of one or several software components.

Secondly, people are an element of the ecosystem as important as software components. Usually, people are users of the technology, they interact with it, but in a technological ecosystem people are part of the ecosystem in a symbiotic way. The user's needs influence in the evolution of the ecosystem, the software tools should adapt to the user's needs and should be prepared to evolve their relationships between software and "human" components.

These two key differences are closely linked to the definition of technological ecosystem. A large number of authors in the literature provide definitions of the technological ecosystems as a metaphor of natural ecosystems [3-10]. In particular, the authors propose a definition of technological ecosystem in previous works [11, 12] that extrapolates the main elements of a natural ecosystem to the field of technology. The software components and the people that compose the technological ecosystem correspond to the biotic factors; The information flows represent or establish relationships between organisms; and abiotic factors are the elements that allow the ecosystem operation (hardware, network, etc.). In this way, a technological ecosystem is a set of users and software components that are related to each other through information flows in a physical environment that provides the support for those flows.

The technological ecosystems have been defined and interpreted in different ways, depending on the point of view. Some works consider that software ecosystems consist of a relatively closed core software system that provides the basic functionality and that is developed by a more or less stable core team of developers, surrounded by a large collection of contributions provided by peripheral developers or even end-users [8, 13, 14]. 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.

The learning ecosystems are a type of technological ecosystem focus on knowledge management in different educational contexts. In the educational field, learning ecosystems propose a true network of learning services beyond providing a collection of fashionable technologies [15-18]. These technological ecosystems allow establishing learning ecologies, learning environments with a strong interactive component that allow the exchange of knowledge in an informal and unstructured way.

The use of information technologies in learning and knowledge management in the near future will be marked by personalization and adaptability [19]. Learning ecosystems should be able to incorporate emerging tools for knowledge management, as well as to remove those that are obsolete or that users do not use. Ecosystems should be able to withstand the increase of internal complexity to transparently offer more functionality and simplicity to users.

The definition, development and deployment of this type of software solutions is complex and involves several problems identified in previous works through a Strengths, Weaknesses, Opportunities and Threats (SWOT) analysis [12]. A first step to solve these problems has been to define an architectural pattern based on the Layers pattern proposed by Buschmann [1]. A second step has been to define a metamodel for developing learning ecosystems following the Model-Driven Architecture (MDA) [20]. 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 [21]. The ecosystems metamodel provides a framework for Model-Driven Development (MDD) of learning ecosystems.

There is no defined metamodels in previous works for developing technological ecosystems and, in particular, for learning ecosystems. The following section delves into the related work.

Although the process to define the ecosystems metamodel is based on the acquired experience defining and developing several learning ecosystems in different contexts, it is required to validate it. A work sent to other conference presents a Platform-Independent Model (PIM) of a learning ecosystem for scientific knowledge management in PhD programs. The main objective of the present work is to provide another case study to validate the ecosystems metamodel.

The paper has been organized in the following way. Section 2 describes the related work about metamodels for developing ecosystems. Section 3 presents the methodology used to validate the metamodel. Section 4 shows the ecosystems metamodel. Section 5 describes the case study. Finally, Section 6 summarizes the main conclusions of this work.

2 RELATED WORK

The goal of the ecosystems metamodel is to give a platform-independent metamodel to describing learning ecosystems composed by heterogeneous elements (Open Source software components, different types of end-users, information flows). There are works about modeling software ecosystems but most of the approaches are not supported by a methodology covering MDA. Moreover, most of them are focused on relationships and collaborations between software ecosystem members, including the developers [22-24].

Although, there no have been found similar approaches published, one part of the ecosystems metamodel is based on the metamodel for web services defined by Jegadeesan and Balasubramaniam [25]; specifically the elements to support the definition of information flows that connect different components. There are several proposal related to metamodels for web services [26-28]. Simon, Goldschmidt and Kondorosi [26] summarize the related works in modeling web services and provide a metamodel that support web services standards (so-called WS*- standards). Regarding the proposal by Jegadeesan and Balasubramaniam [25], they argue: “this is a very detailed metamodel, and it is capable of describing a lot of aspects of web services. However, they only support the simple WS-Policy standard, and they did not include model elements and semantics for WS-* protocols”. Instead, it was selected because the ecosystems metamodel is not focus on providing a complete definition of web services, it uses a very simplified version of the service capability view.

3 METHODOLOGY

Model-Driven Architecture (MDA) provides a framework for software development that uses models to describe the system to be built [29]. It allows to separate the data and operations specification of the system from the details of the platform or platforms on which it will be built. MDA is the proposal of the Object Management Group to apply Model-Driven Development (MDD) using the OMG standards for visualizing, storing, and exchanging software designs and models [30]: Meta Object Facility (MOF), Unified Modeling Language (UML), XML Metadata Interchange (XMI) and Query/View/Transformation (QVT).

The main objectives of MDA are to improve productivity, portability, interoperability and reuse of systems; essential characteristics in tools such as the learning ecosystems which combine software components developed in different programming languages, with different software and hardware requirements, and which they must work together for a common goal.

Moreover, OMG provides a metamodeling architecture, the four-layer metamodel architecture. In this architecture, a model at one layer is used to specify models in the layer below [31]. The four layers are the meta-metamodel layer (M3-layer), the metamodel layer (M2-layer), the user model layer (M1-layer) and the user object layer (M0-layer). Models in below layers are instantiations from models in the top layer.

In this work, MDA is used as guidelines to define conceptual models of learning ecosystems that are PIM. In particular, it has been defined the model of a learning ecosystem for knowledge management in the Spanish Public Administration. This model is an instance of M2-model and it has been defined manually in order to check if the metamodel allows the definition of real learning ecosystems.

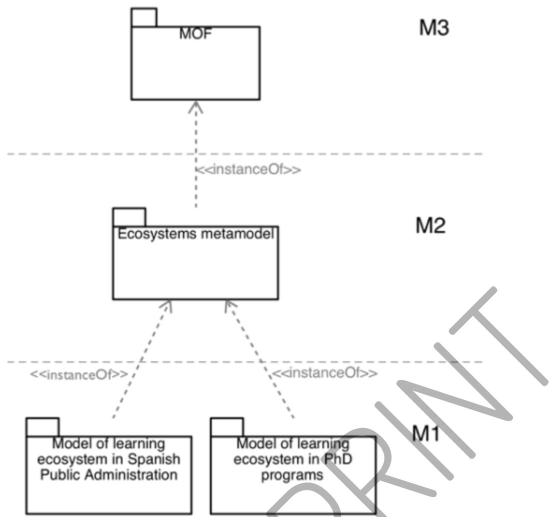

Figure 1: Model layers. The modeling layer (M1); the metamodel layer (M2); and the meta-metamodel layer (M3)

The Fig. 1 shows the four-layers metamodel architecture with the ecosystems metamodel and the models of the learning ecosystems. The learning ecosystem for scientific knowledge management in PhD programs has been modeled in another work. The M0-layer is not represented because the Platform Specific Models (PSM) are not object of study in the present work.

The validation process will provide information to define transformation rules from the ecosystems metamodel, a Computing Independent Model (CIM), to the models of learning ecosystems.

4 THE ECOSYSTEMS METAMODEL

The metamodel proposed in previous work [20] has been defined using MDA and the OMG four-layer metamodel architecture.

The ecosystems metamodel is a M2-model instantiated from MOF, a M3-model. It is used to define PIM of learning ecosystems (M1-models). The main objective of this metamodel is to provide a CIM for describing learning ecosystems build from software components, human elements and information flows between components which are represented by web services. The ecosystems metamodel is not focus on capturing the requirements related to the software or human components of the ecosystem, these elements are black boxes; the metamodel is focus on the connections and adaptations of the components.

The three main elements of a learning ecosystem can be identified in the ecosystems metamodel Fig. 2 the different types of software components; the human factor; and the information flows to establish the relationship between the previous one.

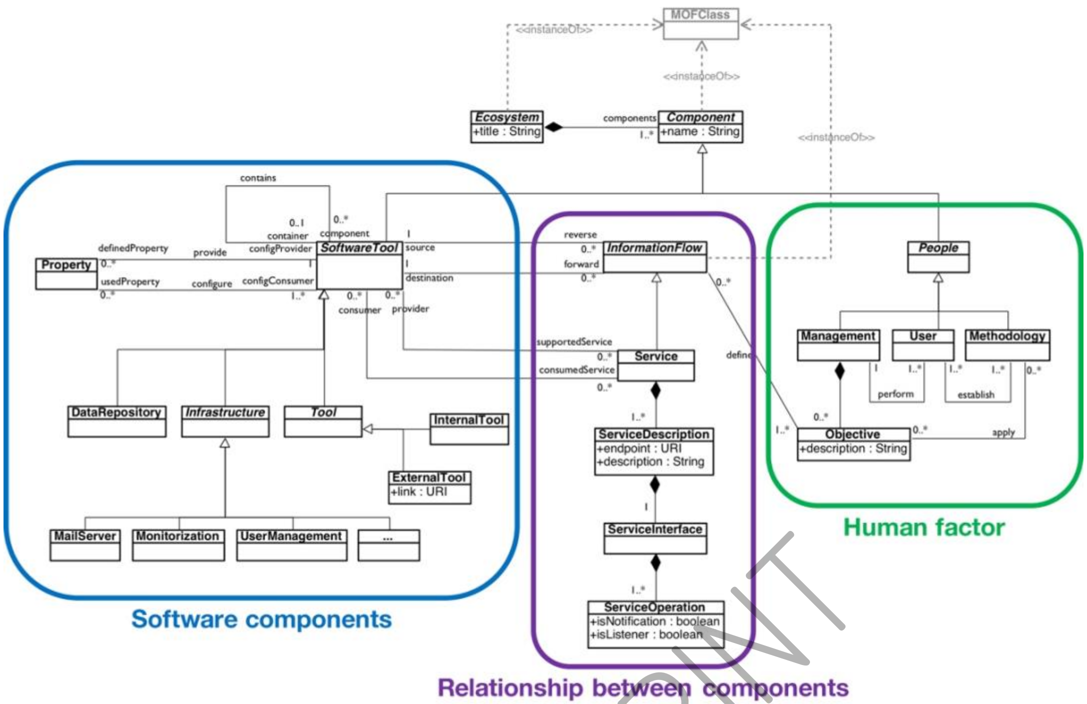

Figure 2: Main parts of the ecosystems metamodel proposal

The metamodel is completed with a set of constraints defined with Object Constraint Language (OCL) to ensure the characteristics identified previously in the architectural pattern of learning ecosystems [1].

The metamodel diagram can be seen in detail on the following link https://doi.org/10.5281/zenodo.829859.

5 SPANISH PUBLIC ADMINISTRATION ECOSYSTEM

5.1 Context

During last years, the Spanish government has implemented the Electronic Administration and has made a strong investment in the 2.0 approach and has fostered the use of Open Source software to cover their technological needs.

The National Institute of Public Administration (INAP) has included in its strategic plan the definition and implementation of a learning ecosystem to promote informal learning inside the public organisms.

The INAP is an autonomous body within the national Government of Spain, attached to the Department of Finance and Public Administration. The Institute possesses a huge experience in knowledge management inside the Public Administration. The following activities can be found between its main tasks: training and professional improvement for public employees; recruitment into the Civil Service; and promoting research and studies regarding government and the different levels on Public Administration from an interdisciplinary perspective.

The learning ecosystem of INAP is composed by a large number of components oriented to cover the different knowledge management needs both Institute inside and outside. There are three main components: the communities of practices based on a vertical social network for public employees (https://social.inap.es); the Knowledge Bank or BCI (*Banco de Conocimiento*, in Spanish) to share the knowledge generated inside the Public Administration (https://bci.inap.es); and the courses repository to share developed course among all institutions that compose the Spanish Public Administration (https://compartir.inap.es). The social and knowledge ecosystem of INAP is fully described in [32, 33].

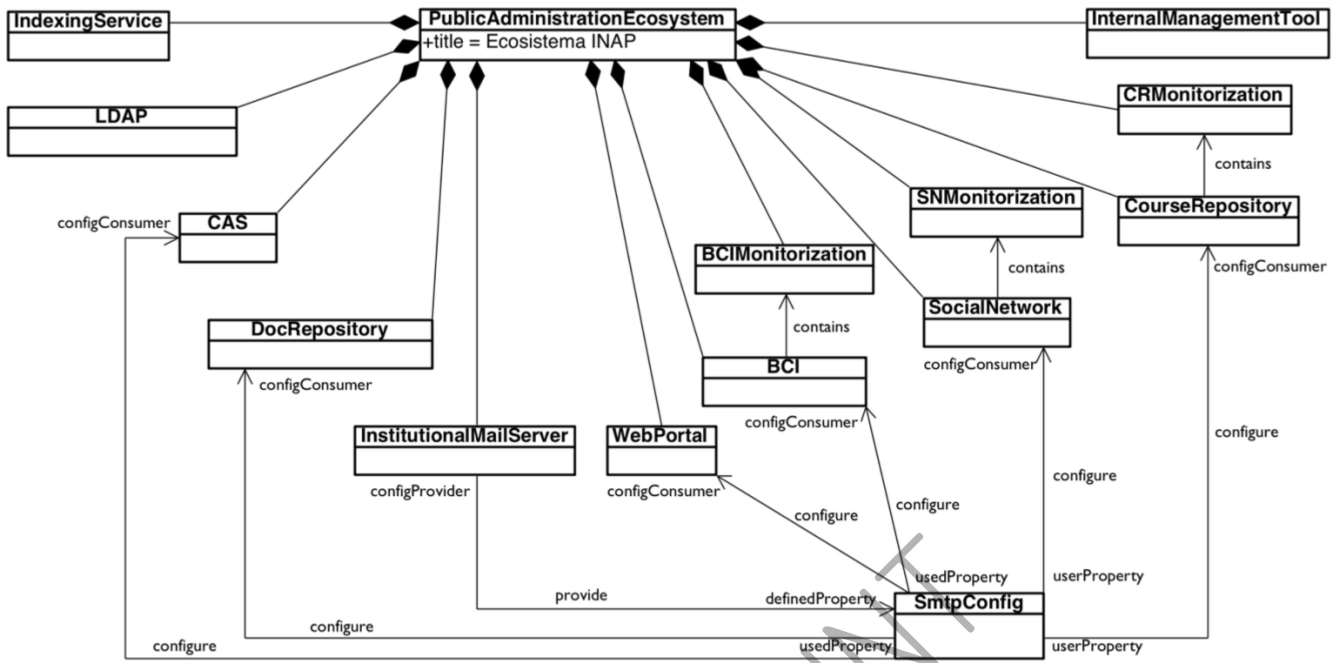

Figure 3: View of software components of the learning ecosystem

5.2 Platform-Independent Model

The conceptual model definition of the learning ecosystem for knowledge management in the Spanish Public Administration has been made by defining three views or packages from the ecosystems metamodel presented above.

The views correspond to the three main parts identified in the ecosystems metamodel (Fig. 2): software components, human elements and the relationship among each other.

First, Fig. 3 shows the view of software components that compose the learning ecosystem of Spanish Public Administration. There are three main types of components: infrastructure, data repositories and tools.

The *PublicAdministrationEcosystem* class represents the ecosystem, it is the main element that contains the other model elements. It is instantiated from the *Ecosystem* class of the metamodel.

The software components are related to the ecosystem using a composition relationship. The classes that represent the infrastructure are: LDAP and CAS that are instances of *UserManagement* class of the metamodel; the *InstitutionalMailServer* class which is an instance of *MailServer*; *BCIMonitorization*, *SNMonitorization* and *CRMonitorization* that provide the monitorization functionality required in all ecosystems according to the architectural pattern for learning ecosystems; and the *IndexingService* class that represents a tool to improve the information searching processes.

The instances of *DataRepository* class, the *DocRepository* class encapsulates the information management that can be shared among the different services.

The tools that provide the user-level services are instances of *InternalTool* class of the metamodel: *BCI*, *SocialNetwork*, *CourseRepository*, *WebPortal* and *InternalManagementTool*. The learning ecosystem modeled does not use external tools such as Facebook or Twitter. The metamodel supports this scenario because there are no OCL rules that force the instantiation of *ExternalTool* class.

Finally, the *SmtpConfig* class is an instance of the *Property* class, whose purpose is to support information flows through configuration files that establish communication protocols. The *SmtpConfig* is provided by *InstitutionalMailServer* class and it is consumed by much of the represented tools.

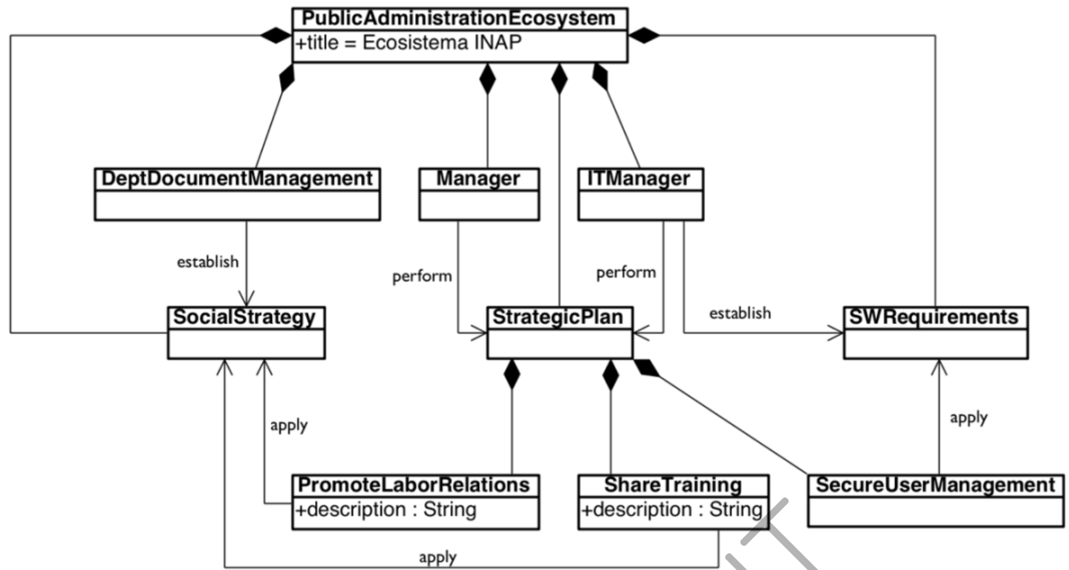

Figure 4: View of human factor of the learning ecosystem

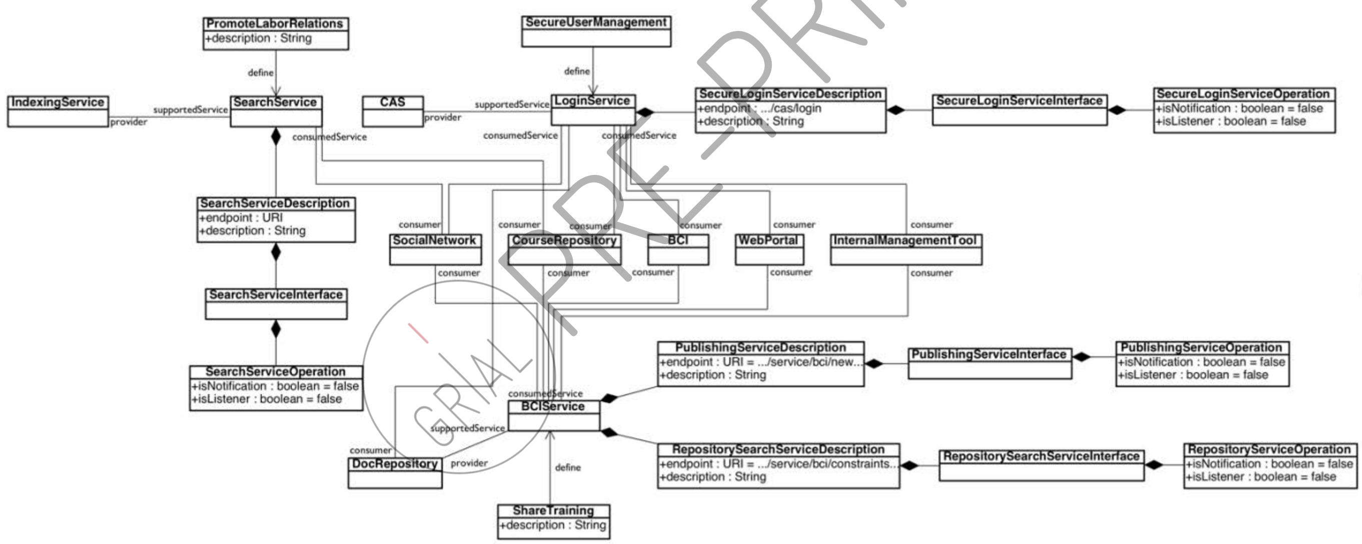

Figure 5: View of the relationships between components of the learning ecosystem

Fig. 4 is focus on the human factor as a key element of the learning ecosystem. The users and the input streams identified in the architectural pattern for learning ecosystems are represented here as instances of *User*, *Management* and *Methodology* classes of the metamodel. In particular, *DeptDocumentManagement*, *Manager* and *ITManager* are instances of *User* class that represent the Department of Publications, Studies and Documentation, the General Management Branch and the Information Technologies manager. The General Management Branch and the Information Technologies Manager perform the Strategic Plan of the INAP represented by the *StrategicPlan* class. The *SocialStrategy* and *SWRequirements* classes are instances of *Methodology* class and provide guidelines to implement the Strategic Plan.

The Strategic Plan is composed by a set of objectives. Only three objectives have been modeled in order to simplify the diagram. The instances of *Objective* class of the metamodel are *PromoteLaborRelations*, *SecureUserManagement and ShareTraining* classes.

Finally, Fig. 5 shows the view of the information flows between components of the ecosystem. This view only shows the information flows related to the objectives previously identified. There are three services defined, one per each objective. First, *SearchService* class represents a service provides by *IndexingService* class and is consumed by two instances of *InternalTool* class. *LoginService* class provides a service to support Single Sign On (SSO) through *CAS* class; the main software components defined above consume this service. *BCIservice* class is provided by *DocRepository* class and it has two service descriptions, one to support the publication of documents and multimedia materials (*PublishingServiceDescription* class) and one to allow searchers (*RepositorySearchServiceDescription* class); all instances of *InternalTool* class use this service.

Regarding the OCL rules proposed in [20] all are fulfilled. First rule forces the instantiation of one *MailServer* class, one *Monitorization* class, one *UserManagement* class, and at least one *InternalTool* class, one *Management* class, one *Methodology* class and one *User* class. In the defined model, there are one instance of *MailServer* class (*InstitutionalMailServer*), three instances of *Monitorization* class (*BCIMonitorization*, *CRMonitorization*, *SNMonitorization*), two instances of *UserManagement* class (*LDAP*, *CAS*), five instances of *InternalTool* class (*BCI*, *SocialNetwork*, *CourseRepository*, *WebPortal*, *InternalManagementTool*), one instance of *Management* class (*StrategicPlan*), two instances of *Methodology* class (*SocialStrategy*, *SWRequirements*), and three instances of *User* class (*DeptDocumentationManagement*, *Manager*, *ITManager*).

Second rule ensure that mail server provides at least one property. The defined model has one instance of *Property* class provides by an instance of *MailServer* class, *SmtpConfig* class.

Third rule, "the value of the endpoint attribute defined in the service descriptions should be unique in the whole ecosystem", is fulfilled, each service description provides different endpoints.

Fourth rule ensures that an instance of *InternalTool* class cannot provide and consume the same service. All defined services are provided by one class and consumed by another.

Furthermore, the model supports the evolution of the ecosystem without invalidating the metamodel. For example, if a software component is deleted the model can include that change and show how the ecosystem is adapted, or if a new component is included, the metamodel provide the classes to instantiate it.

6 CONCLUSIONS

The MDA framework allows to define platform-independent systems through conceptual modeling from different abstraction levels. This approach, applying to the context of technological ecosystems and in particular of learning ecosystems, solves one of the main problems in defining and developing this type of software solutions, since one of its main characteristics is the integration of heterogeneous software components.

The metamodel has allowed to define a model of a real learning ecosystem. All elements of the learning ecosystem for knowledge management in the Spanish Public Administration has been represented in the defined model through instances of ecosystems metamodel. Moreover, the defined model fulfills the metamodel constraints.

This work in conjunction with the model of the learning ecosystem for scientific knowledge management in a PhD program defined in other work, validate of the ecosystems metamodel proposed in [20].

Although the Model-Driven Development is supported by OMG through several standards, there are no stable tools to support the definition and mapping of metamodels and models using those standards. The Eclipse community defined ECORE, a meta-metamodel based on MOF. Eclipse Modeling Project provides tools to develop metamodels using ECORE. Moreover, López-Fernández et al. [34, 35] provide an Eclipse-based tool that facilitates the integral testing of meta-models by making available two dedicated testing languages. It would be interesting to validate the ecosystems metamodel with a testing language. Further work needs to be done to map the metamodel to an instance of ECORE and apply tools like the proposal by López-Fernández et al [34].

ACKNOWLEDGMENTS

This research work has been carried out within the University of Salamanca PhD Programme on Education in the Knowledge Society scope (http://knowledgesociety.usal.es) and was supported by the Spanish Ministerio de Educación, Cultura y Deporte under a FPU fellowship (FPU014/04783).
This work has been partially funded by the Spanish Government Ministry of Economy and Competitiveness throughout the DEFINES project (Ref. TIN2016-80172-R).

REFERENCES

[1] Alicia García-Holgado and Francisco José García-Peñalvo. 2016. Architectural pattern to improve the definition and implementation of eLearning ecosystems. *Science of Computer Programming* 129, 20-34. DOI:http://dx.doi.org/10.1016/j.scico.2016.03.010.

[2] Thomas A Alspaugh, Hazeline U Asuncion, and Walt Scacchi. 2009. The Role of Software Licenses in Open Architecture Ecosystems. In *IWSECO@ ICSR*.

[3] Wu Chen and E. Chang. 2007. Exploring a Digital Ecosystem Conceptual Model and Its Simulation Prototype. In *Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on*, 2933-2938. DOI:http://dx.doi.org/10.1109/ISIE.2007.4375080.

[4] Deepak Dhungana, Iris Groher, Elisabeth Schludermann, and Stefan Biffl. 2010. Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* ACM, New York, NY, USA, 96-102.

[5] Mart Laanpere. 2012. Digital Learning ecosystems: rethinking virtual learning environments in the age of social media. In *Proceedings of the IFIP-OST'12: Open and Social Technologies for Networked Learning* (Taillin2012).

[6] Kai Pata. 2011. Meta-design framework for open learning ecosystems. In *Proceedings of the Mash-UP Personal Learning Environments (MUP/PLE 2011)* (Open University of London, 09/06/2011 2011).

[7] Eric Yu and Stephanie Deng. 2011. Understanding Software Ecosystems: A Strategic Modeling Approach. In *IWSECO-2011 Software Ecosystems 2011. Proceedings of the Third International Workshop on Software Ecosystems. Brussels, Belgium, June 7th, 2011.*, S. Jansen, J. Bosch, P. Campbell and F. Ahmed Eds. CEUR Workshop Proceedings, Aachen, Germany, 65-76.

[8] Tom Mens, Maálick Claes, Philippe Grosjean, and Alexander Serebrenik. 2014. Studying evolving software ecosystems based on ecological models. In *Evolving Software Systems*, T. Mens, A. Serebrenik and A. Cleve Eds. Springer, Berlin, Heidelberg, 297-326. DOI:http://dx.doi.org/10.1007/978-3-642-45398-4_10.

[9] Mircea F Lungu. 2009. Reverse engineering software ecosystems Università della Svizzera italiana.

[10] Mircea F Lungu. 2008. Towards reverse engineering software ecosystems. In *Software Maintenance, 2008. ICSM 2008. IEEE International Conference on* IEEE, 428-431.

[11] Francisco José García-Peñalvo and Alicia García-Holgado (Ed.). 2017. *Open Source Solutions for Knowledge Management and Technological Ecosystems*. IGI Global.

[12] Alicia García-Holgado and Francisco José García-Peñalvo. 2013. 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, NY, USA, 565-571. DOI:http://dx.doi.org/10.1145/2536536.2536623.

[13] Gregorio Robles, Jesus M Gonzalez-Barahona, and Israel Herraiz. 2009. Evolution of the core team of developers in libre software projects. In *Mining Software Repositories, 2009. MSR'09. 6th IEEE International Working Conference on* IEEE, 167-170.

[14] Kumiyo Nakakoji, Yasuhiro Yamamoto, Yoshiyuki Nishinaka, Kouichi Kishida, and Yunwen Ye. 2002. Evolution patterns of open-source software systems and communities. In *Proceedings of the international workshop on Principles of software evolution* ACM, 76-85.

[15] Faraón Llorens. 2011. La biblioteca universitaria como difusor de la innovación educativa. Estrategia y política institucional de la Universidad de Alicante. *Arbor* 187, Extra_3, 89-100.

[16] Faraón Llorens. 2009. La tecnología como motor de la innovación educativa. Estrategia y política institucional de la Universidad de Alicante. *Arbor* 185, Extra, 21-32.

[17] Francisco José García-Peñalvo, Ángel Hernández-García, Miguel Á Conde-González, Ángel Fidalgo-Blanco, María L Sein-Echaluce, Marc Alier, Faraón Llorens-Largo, and Santiago Iglesias-Pradas. 2015. Learning services-based technological ecosystems. In *Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality* ACM, 467-472.

[18] Nikolas Galanis, Enric Mayol, M. J. Casany, and Marc Alier. 2017. Tools Interoperability for Learning Management Systems. In *Open Source Solutions for Knowledge Management and Technological Ecosystems* IGI Global, Hershey, PA, USA, 25-49. DOI:http://dx.doi.org/10.4018/978-1-5225-0905-9.ch002.

[19] Faraón Llorens. 2014. Campus virtuales: De gestores de contenidos a gestores de metodologías. *RED, Revista de Educación a Distancia* 42, 1-12.

[20] Alicia García-Holgado and Francisco José García-Peñalvo. 2017. A Metamodel Proposal for Developing Learning Ecosystems. In *Learning and Collaboration Technologies. Novel Learning Ecosystems. LCT 2017. Lecture Notes in Computer Science*, P. Zaphiris and A. Ioannou Eds. Springer, Cham. DOI:http://dx.doi.org/10.1007/978-3-319-58509-3_10.

[21] Jorge J Gómez-Sanz and Juan Pavón. 2002. Meta-modelling in agent oriented software engineering. In *Ibero-American Conference on Artificial Intelligence 2002*, F.J. Garrijo, J.-C. Riquelme and M. Toro Eds. Springer, Heidelberg, 606-615.

[22] Ivo Van Den Berk, Slinger Jansen, and Lútzen Luinenburg. 2010. Software ecosystems: a software ecosystem strategy assessment model. In *Proceedings of the Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (Copenhagen, Denmark2010). ACM, 1842781, 127-134. DOI:http://dx.doi.org/10.1145/1842752.1842781.

[23] Oskar Pettersson, Martin Svensson, Didac Gil, Jesper Andersson, and Marcelo Milrad. 2010. On the role of software process modeling in software ecosystem design. In *Proceedings of the Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (Copenhagen, Denmark2010). ACM, 1842778, 103-110. DOI:http://dx.doi.org/10.1145/1842752.1842778.

[24] I. T. Hawryszkiewycz. 2005. A Metamodel for Modeling Collaborative Systems. *Journal of Computer Information Systems* 45, 3 (2005/03/01), 63-72. DOI:http://dx.doi.org/10.1080/08874417.2005.11645844.

[25] Harshavardhan Jegadeesan and Sundar Balasubramaniam. 2008. An MOF2-based Services Metamodel. *Journal of Object Technology* 7, 8, 71-96.

[26] Balazs Simon, Balazs Goldschmidt, and Karoly Kondorosi. 2013. A Metamodel for the Web Services Standards. *Journal of Grid Computing* 11, 4, 735-752. DOI:http://dx.doi.org/10.1007/s10723-013-9273-4.

[27] Cao Fei, B. R. Bryant, Zhao Wei, C. C. Burt, R. R. Raje, A. M. Olson, and M. Auguston. 2004. A meta-modeling approach to Web services. In *Proceedings. IEEE International Conference on Web Services, 2004.*, 796-799. DOI:http://dx.doi.org/10.1109/ICWS.2004.1314824.

[28] Fei Cao, Barrett R Bryant, Wei Zhao, Carol C Burt, Rajeev R Raje, Andrew M Olson, and Mikhail Auguston. 2004. A Meta-modeling approach to Web Services. In *Web Services, 2004. Proceedings. IEEE International Conference on* IEEE, 796-799.

[29] Stephen J. Mellor, Kendall Scott, Axel Uhl, and Dirk Weise. 2002. Model-Driven Architecture. In *Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier, France, September 2, 2002 Proceedings*, J.-M. Bruel and Z. Bellahsene Eds. Springer Berlin Heidelberg, Berlin, Heidelberg, 290-297. DOI:http://dx.doi.org/10.1007/3-540-46105-1_33.

[30] David Frankel. 2002. *Model Driven Architecture: Applying MDA to Enterprise Computing*. John Wiley & Sons, Inc.

[31] José M. Álvarez, Andy Evans, and Paul Sammut. 2001. Mapping between Levels in the Metamodel Architecture. In *UML*, M. Gogolla and C. Kobryn Eds. Springer, Berlin, Heidelberg, 34-46. DOI:http://dx.doi.org/10.1007/3-540-45441-1_4.

[32] Alicia García-Holgado and Francisco José García-Peñalvo. 2014. Knowledge management ecosystem based on drupal platform for promoting the collaboration between public administrations. In *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'14) (Salamanca, Spain, October 1-3, 2014)*, F.J. García-Peñalvo Ed. ACM, New York, NY, USA, 619-624. DOI:http://dx.doi.org/10.1145/2669711.2669964.

[33] Alicia García-Holgado, Juan Cruz-Benito, and Francisco José García-Peñalvo. 2015. Analysis of Knowledge Management Experiences in Spanish Public Administration. In *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, 189-193. DOI:http://dx.doi.org/10.1145/2808580.2808609.

[34] Jesús J. López-Fernández, Esther Guerra, and Juan De Lara. 2014. Meta-Model validation and verification with MetaBest. In *Proceedings of the Proceedings of the 29th ACM/IEEE international conference on Automated software engineering* (Vasteras, Sweden2014). ACM, 2648617, 831-834. DOI:http://dx.doi.org/10.1145/2642937.2648617.

[35] Jesús J. López-Fernández, Esther Guerra, and Juan De Lara. 2014. Assessing the Quality of Meta-models. In *MoDeVVa*, F. Boulanger, M. Famelis and D. Ratiu Eds. CEUR Workshop Proceedings, Valencia, Spain, 3-22.