

García-Holgado, A., & García-Péñalvo, F. J. (2017). A metamodel proposal for developing learning ecosystems. In P. Zaphiris & A. Ioannou (Eds.), *Learning and Collaboration Technologies. Technology in Education. 4th International Conference, LCT 2017. Held as Part of HCI International 2017, Vancouver, BC, Canada, July 9–14, 2017. Proceedings, Part I* (pp. 100-109). Switzerland: Springer International Publishing. doi: 10.1007/978-3-319-58509-3_10

The final publication is available at link.springer.com

A metamodel proposal for developing learning ecosystems

Alicia García-Holgado^{1*}, Francisco José García-Péñalvo¹

¹GRIAL Research Group, Computer Sciences Department,
Research Institute for Educational Sciences,
University of Salamanca, Salamanca, Spain
{aliciah,fgarciap}@usal.es}

Abstract. The definition and development of learning ecosystems is a complex process with a wide range of requirements. Although two different institutions or companies share the same problems and goals regarding their learning and training processes, the learning ecosystems to support them are different. The components of the ecosystem, including the human factor as a key element, and the relationships between them, change over time. In other words, learning ecosystems evolve as natural ecosystems; there are many factors, both internal and external, that influence an entity. The authors have defined and developed different learning ecosystems. Moreover, they have transferred the same learning ecosystem, specifically a learning ecosystem for knowledge management in a PhD Program, to different domains. These experiences have provided the required information to define the ecosystems metamodel following the Model Driven Architecture proposed by the Object Management Group. The aim of this metamodel is to define a Domain Specification Language to develop learning ecosystems.

Keywords: learning ecosystems; metamodel; MOF; Model-Driven Architecture; information systems

1 Introduction

The technological ecosystems are the evolution of the traditional information systems. They provide support to information and knowledge management in heterogeneous environments [1, 2]. The learning ecosystems are a type of technological ecosystems focus on learning management processes.

The metaphor of technological ecosystem transfers the main properties of biological or natural ecosystems to the technological area. The organisms of the natural ecosystems are the users and the software components from a technological

point of view; the relationships among the organisms are the information flows between the components; and the physical environment are the mechanism to establish and support such flows [3]. This relationship between technology and nature appears in many authors that provide their own definitions of technological ecosystems, also called Software ECOSystems (SECO) [4-9].

Technological ecosystems should connect and relate the different tools and services that arise and serve for the knowledge management, building technological ecosystems, increasingly complex internally, from the semantic interoperability of its components to transparently provide more functionality and simplicity to its users [10].

One of the main problems to define and develop technological ecosystems, particularly learning ecosystems, is the need to adapt them to the natural evolution of the companies and institutions. A technological ecosystem should adapt to different contexts or domains, i.e. it should be transferable between different domains.

Currently, the development of a learning ecosystem is influenced by many factors, both internal and external. Although the main goals are the same, the software components and information flows can change even in the same entity. Each new ecosystem or the transfer of an existing one involves carrying out a large number of ad-hoc developments.

Within the Research Group in InterAction and eLearning (GRIAL) of the University of Salamanca, the authors have participated in the development of several learning ecosystems in different contexts to solve real problems [11-14]. Moreover, they have transferred the same learning ecosystem, specifically a learning ecosystem for knowledge management in a PhD Program [15], to different domains.

To improve the definition and development of learning ecosystems, it is needed to provide a platform-independent solution. The main objective of this paper is to define a metamodel for developing learning ecosystems following the Model-Driven Architecture (MDA) proposed by the Object Management Group (OMG).

The paper has the following structure: Section 2 describes the methodology used to formalize the metamodel proposal. Section 3 sets the high-level requirements for the learning ecosystems development. Section 4 describes the metamodel as a M2-level model in OMG four-layer metamodel architecture. In section 5 a real learning ecosystem is modelled using the Ecosystems Metamodel. Finally, section 6 summarizes the main conclusions from this study.

2 Methodology

Model-Driven Architecture provides a framework for software development that uses models to describe the system to be built [16]. 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 [17]: Meta Object Facility

(MOF), Unified Modeling Language (UML), XML Metadata Interchange (XMI) and Query/View/Transformation (QVT).

In this work, the MDA is used as guidelines to define learning ecosystems using high-level conceptual models, platform-independent models (PIM). These models can later be translated to concrete executable specifications or code using standard-mappings. This is performed according the OMG four-layer metamodel architecture. In this architecture, a model at one layer is used to specify models in the layer below [18]. The four layers are the meta-metamodel layer (M3), the metamodel layer (M2), the user model layer (M1) and the user object layer (M0).

The Ecosystems Metamodel is an instantiation of the meta-metamodel layer using MOF language, the abstract language used to describe MOF metamodel.

3 High-level requirements

The main problems associated with the definition and development of learning ecosystems have been identified in a previous work through a comparative analysis of the Strengths, Weaknesses, Opportunities and Threats (SWOT) of several real case studies developed in different contexts [19]. Based on this analysis, the identified problems were modelled with Business Process Model and Notation (BPMN) to provide a high abstraction level of the main problems in learning ecosystems and define an architectural pattern to resolve them during the definition phase [20].

Learning ecosystems modeling involves the proposed architectural pattern, a pattern based on the Layers pattern defined by Buschmann [21] with a top-down scheme composed of four layers. Modelling of learning ecosystems must be supported by a formal Ecosystems Metamodel. This metamodel is not focus on capturing the requirements related to the software or human components of the ecosystem. The components are black boxes; the Ecosystems Metamodel does not enable capture of the description of a specific component because of learning ecosystems are based on connect or adapt existing components. The metamodel should enable capture of a small set of modeling elements to define the relationships among components.

The high-level requirements for ecosystems metamodel are the following:

- 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.

4 Ecosystems Metamodel

In this section, the formal semantics of the Ecosystems Metamodel is described. The metamodel is a M2-level model in the four-layer metamodel architecture. The Ecosystems Metamodel is an instance of the MOF meta-metamodel (M3-level model) (Fig. 1).

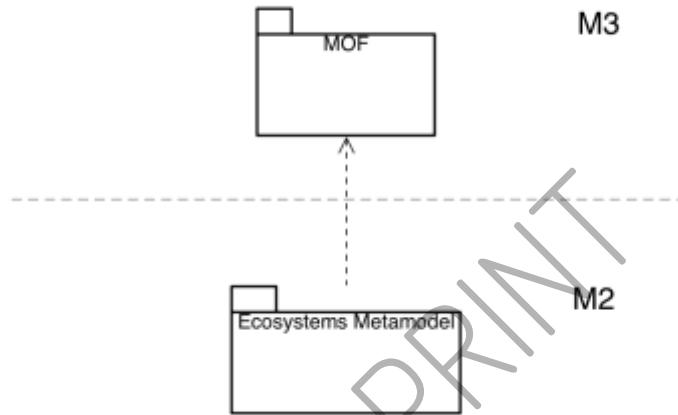


Fig. 1. Model layers. The meta-metamodel layer, M3, and the metamodel layer, M2.

The Ecosystems Metamodel defines an ecosystem following the architectural pattern proposed in a previous work [20]. The pattern is composed by four layers – presentation, services, static data management and infrastructure – and two input streams which introduce the human factor as a key element. Also, the pattern provides a set of software components that should be part of a learning ecosystem to resolve some problems of this kind of technological solutions. The layers and the components proposed are reflected in the metamodel.

An ecosystem is made up of a collection of two type of components, software tools and people. Both components are abstract classes that inherit from a root class named *Component*. The software tools are organized in a hierarchical structure that provides three layers of the pattern: the service layer through *Tool* class; the static data management layer with the *DataRepository* class; and the infrastructure layer through a class with the same name. The authors have decided that the presentation layer is not part of the metamodel due to the interfaces of the software components are closely related to the technology used in each component.

Moreover, the ecosystem can be composed by software tools that contains others ones, this is modeled with a recursive association in *SoftwareTool* class.

The set of software components are part of the hierarchy describe above. *MailServer*, *Monitorization* and *UserManagement* inherit from *Infrastructure* class. Moreover, other child classes can be defined from *Infrastructure*.

Regarding the tools in the service layer, these are modeled as child classes of the *Tool* class, *InternalTool* and *ExternalTool*.

The human factor is modeled through *Management*, *Methodology* and *User* classes. *User* performs management and establishes one or more methodologies. Moreover, the *Management* is composed by zero or more objectives modeled by the *Objective* class.

These objectives are the key element to define the information flows. An *InformationFlow* is an abstract root class that establishes a relationship between two *SoftwareTool* instances. The information flows are based on services. This requirement is modeled through a leaf class, *Service*. The four classes related to the services are a very simplified version of the service capability view of the services metamodel proposed by Jegadeesan and Balasubramaniam [22]. *ServiceDescription* has a semantic description of the service and also includes the endpoint of the service. *ServiceInterface* represents the underlying capabilities brought to bear by a service. *ServiceOperation* represents an underlying capability and allows modeling event-driven scenarios using the *isNotification* and *isListener* attributes.

Finally, the *Property* class provide the semantic to model the configuration provided by some software components and used by another. This part of the metamodel complements the information flows to establish different relationship levels among the components.

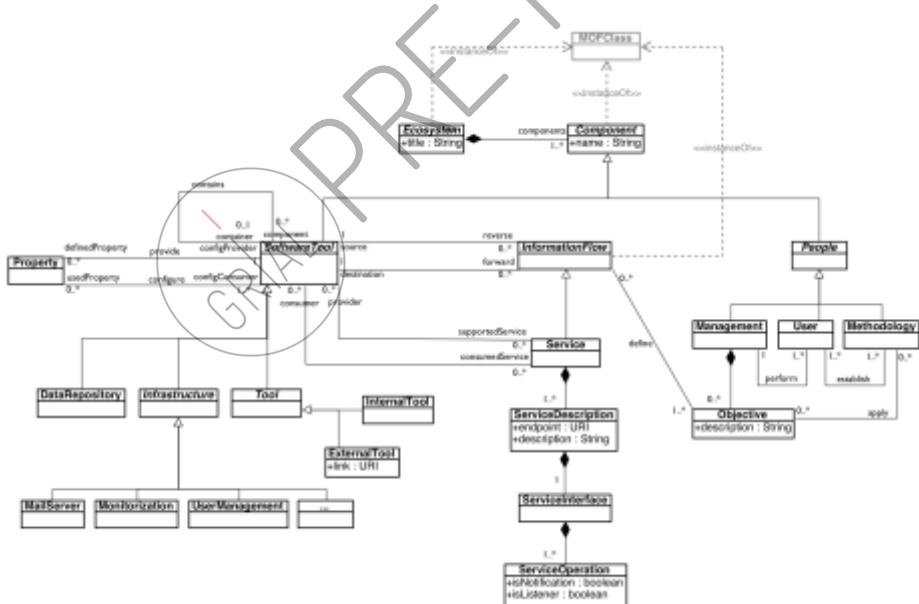
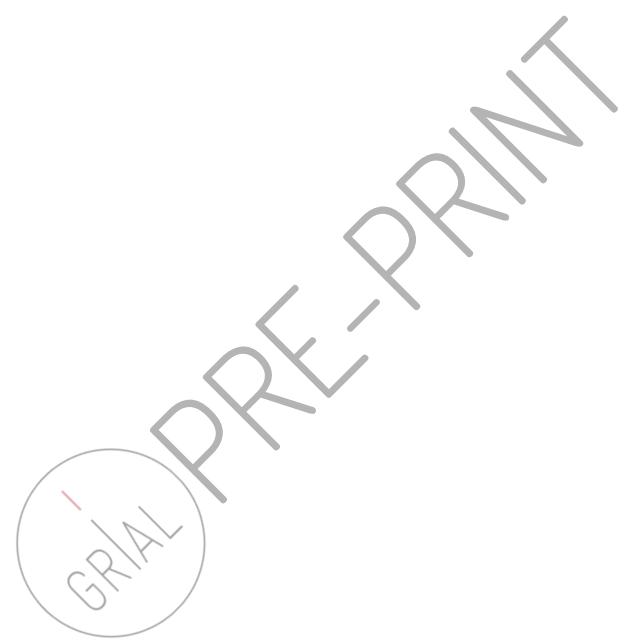


Fig. 2. Ecosystems Metamodel

The metamodel proposed in the Fig. 2 is completed with a set of constraints defined with Object Constraint Language (OCL).

An ecosystem must have one mail server, one monitorization system, one user management system, and at least one internal tool, one management input stream, one methodology input stream and one user:



```

context EcosystemInv:
self.components -> select(c |
  c.oclIs TypeOf( MailServer)) -> size() = 1 and self.components -> select(c |
  c.oclIs TypeOf( Monitoring)) -> size() = 1 and
self.components -> select(c |
  c.oclIs TypeOf( UserManagement)) -> size() = 1 and self.components ->
select(c |
  c.oclIs TypeOf( InternalTool)) -> notEmpty and
self.components -> select(c |
  c.oclIs TypeOf( Management)) -> notEmpty and
self.components -> select(c |
  c.oclIs TypeOf( Metadatalogy)) -> notEmpty and
self.components -> select(c |
  c.oclIs TypeOf( User)) -> notEmpty

```

The value of the endpoint attribute defined in the service descriptions should be unique in the whole ecosystem:

```

context ServiceDescriptionInv:
ServiceDescriptions.allInstances -> forAll(p1, p2 |
  p1 <> p2 implies p1.endpoint <> p2.endpoint)

```

A software tool cannot consume a service provided by itself, that is to say, the information flows always involve two different software tools:

```

context SoftwareToolInv:
self <> self.consumedService.provider

```

The mail server must provide at least one property:

```

context MailServerInv:
self.definedProperty -> notEmpty

```

5 Modeling the ecosystem for Scientific Knowledge Management in a PhD Program

The learning ecosystem modeled from the Ecosystems Metamodel is oriented to manage the scientific knowledge generated in the scope of the PhD Program on Education in the Knowledge Society at the University of Salamanca (<https://knowledgesociety.usal.es>). This ecosystem is described by García-Holgado, García-Péñalvo and Rodríguez Conde [15].

This learning ecosystem is composed by three elements in the infrastructure layer: (1) a mail server provided by the University of Salamanca; (2) a user management tool that is part of a component located in the service layer, the portal; (3) and a monitorization tool that is also part of the portal.

The static data management layer is provided by the institutional repository of the University of Salamanca.

Finally, the service layer has a main component, a user-centered portal which provides most features required by the business logic; and a set of external tools focused on the dissemination of the scientific knowledge in different social networks (Twitter, YouTube, SlideShare and Facebook) and sending bulk emails (Mailchimp).

The model (M1-level) has been divided in three packages: the ecosystem tools model; the ecosystem users model; and the ecosystem services model. Furthermore, the classes from the metamodel (M2-level) are represented to indicate which classes are used to model the example.

In Fig. 3, the main software components of the learning ecosystem are modeled. The *PhDEcosystem* is composed by: a *GmailSmtpServer*, modeled using the *MailServer* class; the *InstitutionalRepository* that instances the *DataRepository* class; the *PhDPortal* which contains the *PortalUserManagement* and the *PortalMonitorization*, instances of *InternalTool*, *UserManagement* and *Monitorization*, respectively; and a set of modeled elements using the *ExternalTool* class.

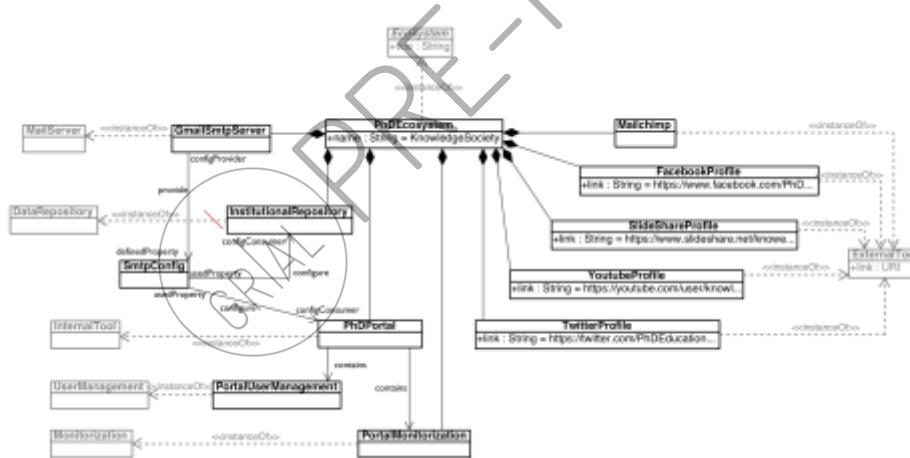


Fig. 3. Ecosystem tools model

The human factor is represented by the Academic Committee that is in charge of the management of the PhD Program, including the learning ecosystem. And a Quality Committee that provide the methodology to ensure the quality of the Program and the learning ecosystem.

In Fig. 4, the human factor is modeled. The *AcademicCommittee* performs the *PhDGGuidelines* and the *PhDProcedures*, both modeled using the *Management* class. The *QualityCommittee* establishes the *QualityPlan* modeled using the *Methodology*

class. The *PhDProcedures* provides a set of objectives, that are involved in the modeling of the information flows.

Finally, the information flows of the learning ecosystem are modeled in the Fig. 5. The relationships among the components are supported by several services to achieve the objectives (*PublishEvidences*, *DisseminateActivities* and *GetEvaluationIndicators*) that are part of the *PhDProcedures* in Fig. 4. Each service is modeled using *Service* class and involves two instances of *SoftwareTool*.

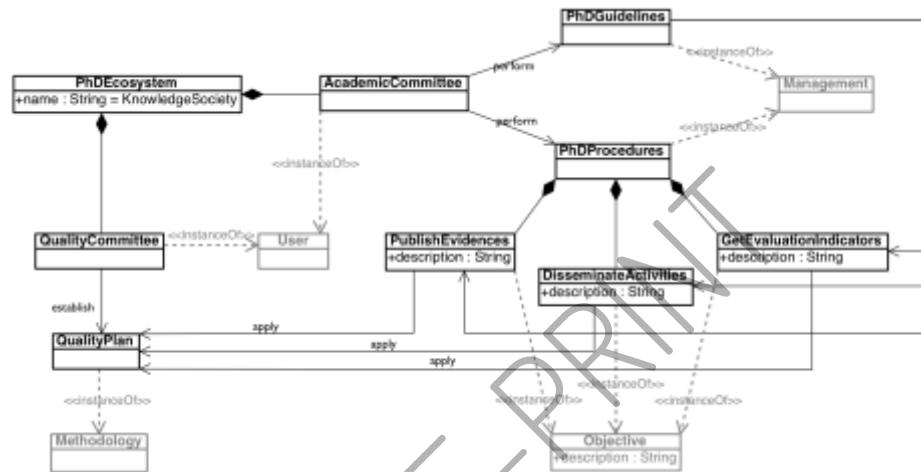


Fig. 4. Ecosystem users model

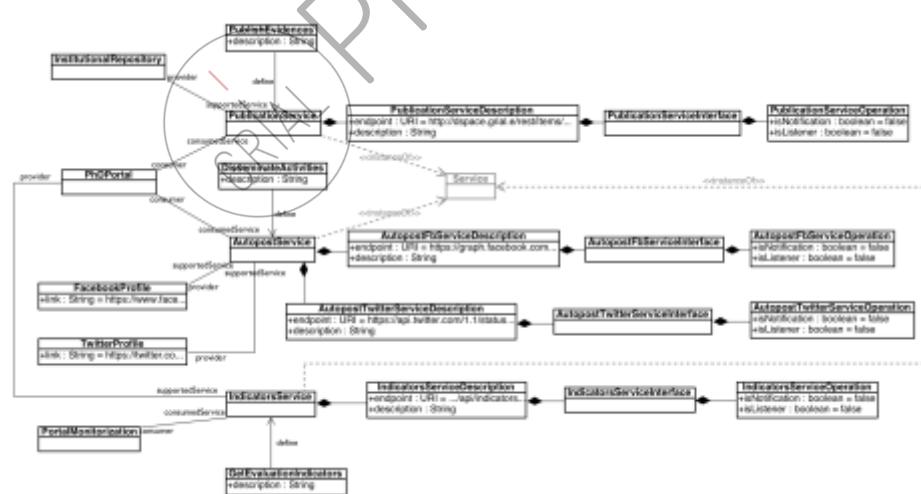


Fig. 5. Ecosystem services model

The *PublicationService* is provided by *InstitutionalRepository* and it is consumed by *PhDPortal* to support the *PublishEvidences* objective. The *AutopostService* is provided by two instances of *ExternalTool*, *FacebookProfile* and *TwitterProfile* to support the *DisseminateActivities* objective. And the *IndicatorsService* is defined by the *GetEvaluationIndicators* objective and involves *PhDPortal* and *PortalMonitorization*.

It should be noted that the *PhDPortal* consumes two services and provides another. Also, there is a service, *AutopostService*, composed by two instances of *ServiceDescription*, *AutopostFbServiceDescription* and *AutopostTwitterServiceDescription*.

6 Conclusions

This paper presents an ecosystems metamodel based on MOF to model different perspectives of learning ecosystems development. The metamodel have been used to model a real learning ecosystem for a PhD Program, which have been transferred to other domains. The modelling example demonstrate how a real ecosystem could be modeled using the Ecosystems Metamodel.

This work provides a base to define a UML profile for the Ecosystems Metamodel in order to leverage existing tool support. Furthermore, a set of transformation rules using QVT could be defined to transform the models into executable specifications to provide an ecosystem prototype.

7 Acknowledgement

This research was performed within the University of Salamanca PhD Program 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. García-Peña, F.J., Hernández-García, Á., Conde, M.Á., Fidalgo-Blanco, Á., Sein-Echaluce, M.L., Alier, M., Llorens-Largo, F., Iglesias-Pradas, S.: Learning services-based technological ecosystems. In: Proceedings of the 3rd International Conference on Technological Ecosystems for Enhancing Multiculturality, pp. 467-472. ACM, (Year)
 2. Ochoa, A., Yañez, S., Martinez-Arevalo, C., Alvarez, F., Caja, J., Maresca, P., Adrada, T., Merino, M., Zanón, A.: Competencias en el contexto del Proyecto Mentor de una Escuela de Ingeniería. In: Fidalgo Blanco, Á., Sein-Echaluce Lacleta, M.L., García-Peña, Á.

F.J. (eds.) La Sociedad del Aprendizaje. Actas del III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad. CINAIC 2015 (14-16 de Octubre de 2015, Madrid, España), pp. 670-675. Fundación General de la Universidad Politécnica de Madrid, Madrid, Spain (2015)

3. García-Holgado, A., García-Peñalvo, F.J.: The evolution of the technological ecosystems: an architectural proposal to enhancing learning processes. Proceedings of the First International Conference on Technological Ecosystem for Enhancing Multiculturality (TEEM'13) (Salamanca, Spain, November 14-15, 2013), pp. 565-571. ACM, New York, NY, USA (2013)
4. Chang, E., West, M.: Digital Ecosystems A Next Generation of the Collaborative Environment. In: Eight International Conference on Information Integration and Web-based Application & Services, pp. 3-23. (Year)
5. Chen, W., Chang, E.: Exploring a Digital Ecosystem Conceptual Model and Its Simulation Prototype. In: Industrial Electronics, 2007. ISIE 2007. IEEE International Symposium on, pp. 2933-2938. (Year)
6. Dhungana, D., Groher, I., Schludermann, E., Biffl, S.: Software ecosystems vs. natural ecosystems: learning from the ingenious mind of nature. In: Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, pp. 96-102. ACM, (Year)
7. Laanpere, M.: Digital Learning ecosystems: rethinking virtual learning environments in the age of social media. IFIP-OST'12: Open and Social Technologies for Networked Learning, Taillin (2012)
8. Messerschmitt, D.G., Szyperski, C.: Software ecosystem: understanding an indispensable technology and industry. MIT Press Books 1, (2005)
9. Pata, K.: Meta-design framework for open learning ecosystems. Mash-UP Personal Learning Environments (MUP/PLE 2011), Open University of London (2011)
10. Pardo, A., Delgado Kloos, C.: Stepping out of the box: Towards analytics outside the learning management system. LAK '11 Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 163-167. ACM, New York, NY, USA (2011)
11. García-Holgado, A., García-Peñalvo, F.J.: Knowledge management ecosystem based on drupal platform for promoting the collaboration between public administrations. In: García-Peñalvo, F.J. (ed.) Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'14) (Salamanca, Spain, October 1-3, 2014), pp. 619-624. ACM, New York, NY, USA (2014)
12. García-Peñalvo, F.J., Conde-González, M.Á., Zangrando, V., García-Holgado, A., Seoane Pardo, A.M., Alier, M., Galanis, N., Brouns, F., Vogten, H., Griffiths, D., Mykowska, A., Alves, G.R., Minović, M.: 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, 1661-1683 (2013)
13. García-Peñalvo, F.J., Johnson, M., Alves, G.R., Minović, M., Conde-González, M.Á.: Informal learning recognition through a cloud ecosystem. Future Generation Computer Systems 32, 282-294 (2014)
14. Orueta, J.L., Pavón, L.M. (eds.): Libro Blanco de la Universidad Digital 2010. Ariel Colección Fundación Telefónica. Cuaderno 11 (2008)

15. García-Holgado, A., García-Peñalvo, F.J., Rodríguez Conde, M.J.: Definition of a Technological Ecosystem for Scientific Knowledge Management in a PhD Programme. Proceedings of the Third International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'15) (Porto, Portugal, October 7-9, 2015), pp. 695-700. ACM, New York, NY, USA (2015)
16. Mellor, S.J., Scott, K., Uhl, A., Weise, D.: Model-Driven Architecture. In: Bruel, J.-M., Bellahsene, Z. (eds.) Advances in Object-Oriented Information Systems: OOIS 2002 Workshops Montpellier, France, September 2, 2002 Proceedings, pp. 290-297. Springer Berlin Heidelberg, Berlin, Heidelberg (2002)
17. Frankel, D.: Model Driven Architecture: Applying MDA to Enterprise Computing. John Wiley & Sons, Inc. (2002)
18. Álvarez, J.M., Evans, A., Sammut, P.: Mapping between Levels in the Metamodel Architecture. In: Gogolla, M., Kobryn, C. (eds.) «UML» 2001 — The Unified Modeling Language. Modeling Languages, Concepts, and Tools: 4th International Conference Toronto, Canada, October 1–5, 2001 Proceedings, pp. 34-46. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
19. García-Holgado, A., García-Peñalvo, F.J.: Architectural pattern for the definition of eLearning ecosystems based on Open Source developments. In: Sierra-Rodríguez, J.L., Dodero-Beardo, J.M., Burgos, D. (eds.) Proceedings of 2014 International Symposium on Computers in Education (SIIE) (Logroño, La Rioja, Spain, November 12-14, 2014), pp. 93-98. Institute of Electrical and Electronics Engineers. IEEE Catalog Number CFP1486T-ART (2014)
20. García-Holgado, A., García-Peñalvo, F.J.: Architectural pattern to improve the definition and implementation of eLearning ecosystems. Science of Computer Programming 129, 20-34 (2016)
21. Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, M.: Pattern-oriented software architecture: a system of patterns (1996)
22. Jegadeesan, H., Balasubramaniam, S.: An MOF2-based Services Metamodel. Journal of Object Technology 7, 71-96 (2008)