2 Methodology

In order to guarantee the quality of the learning ecosystem metamodel, a series of analysis and transformations have been performed. First, the metamodel instantiated from MOF has been analysed following the quality framework provided by López-Fernández, Guerra and de Lara [12]. The objective has been to know the quality problems of the metamodel to solve them in the Ecore version.

After the analysis, the learning ecosystem metamodel in MOF has been transformed in an instance of Ecore. Both metamodels are M2-model in the four–layer metamodel architecture provided by MDA (Fig. 1). Both MOF and Ecore support the use of XMI enabling the interchange of models and model instances through XML based on DTDs/XMLSchemas generated from the corresponding models [13]. However, in this work 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.

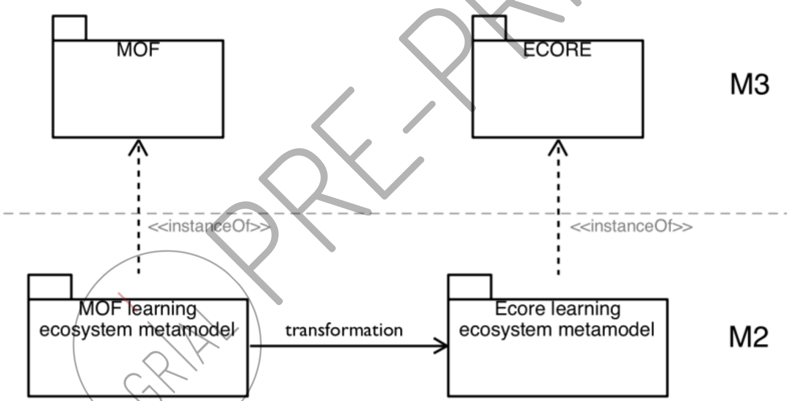

Fig. 1. Different abstraction levels of the learning ecosystem metamodel

The first version of the metamodel has a set of constraints defined with Object Constraint Language (OCL) and included in the metamodel as annotations. During the transformation process, these constraints have been reviewed and finally twelve constraints have been included in the Ecore metamodel. Moreover, the constraints in the Ecore metamodel are included in each element using the OCLinEditor provided by EMF.

Finally, the quality of the learning ecosystem metamodel instantiated from Ecore has been checked. It has been used the same metamodel quality framework than in the MOF version.

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 the features D03 and BP03. 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 in two classes, one with the same name that represent 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 feature BP03, 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 (Fig. 2) fulfils with the thirty features that compose the framework. Highlight the metrics:

- M01. 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.

5 Conclusions

The learning ecosystem metamodel is a M2-model in the four–layer metamodel architecture provided by MDA. The main objective of this metamodel is to provide a Computing Independent Model (CIM) for describing learning ecosystems building from software components, human elements and information flows between them.

The validation of the metamodel is necessary to provide a robust solution for the development of this type of technological solutions. In previous works a first phase has been carried out; two M2M transformations have been made to test that the metamodel allows to define real learning ecosystems. These preliminary validations have been made manually because there are no stable tools that support the standards defined by OMG.

The transformation from MOF to Ecore of the learning ecosystem metamodel represents an important step in the validation process because of the Ecore version can be an input in the different modelling tools provided by Eclipse. Furthermore, the metamodel instantiated from Ecore (Fig. 1) is a quality metamodel according to the quality framework proposed by López-Fernández, Guerra and de Lara [12].

In future works the validation process will be completed defining a set of transformation rules to transform a PIM model instantiated from the learning ecosystem metamodel to a PSM model that represent the deployment of the learning ecosystem in a real context.

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 *Ministry of Education, Culture and Sport* 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) and the Ministry of Education of the Junta de Castilla y León (Spain) throughout the T-CUIDA project (Ref. SA061P17).

References

1. García-Peñalvo, F.J., García-Holgado, A. (eds.): Open Source Solutions for Knowledge Management and Technological Ecosystems. IGI Global (2017)

2. 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)
3. Alspaugh, T.A., Asuncion, H.U., Scacchi, W.: The Role of Software Licenses in Open Architecture Ecosystems. In: IWSECO@ ICSR. (Year)
4. 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)
5. Gómez-Sanz, J.J., Pavón, J.: Meta-modelling in agent oriented software engineering. In: Garrijo, F.J., Riquelme, J.-C., Toro, M. (eds.) Ibero-American Conference on Artificial Intelligence 2002, vol. 2527, pp. 606-615. Springer, Heidelberg (2002)
6. García-Holgado, A., García-Peñalvo, F.J.: A Metamodel Proposal for Developing Learning Ecosystems. In: Zaphiris, P., Ioannou, A. (eds.) Learning and Collaboration Technologies. Novel Learning Ecosystems. LCT 2017. Lecture Notes in Computer Science, vol. 10295. Springer, Cham (2017)
7. Frankel, D.: Model Driven Architecture: Applying MDA to Enterprise Computing. John Wiley & Sons, Inc. (2002)
8. 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)
9. García-Holgado, A., García-Peñalvo, F.J.: Definición de ecosistemas de aprendizaje independientes de plataforma. IV Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2017) (Zaragoza, Spain, October 4-6, 2017), (2017)
10. García-Holgado, A., García-Peñalvo, F.J.: Preliminary validation of the metamodel for developing learning ecosystems. In: Dodero, J.M., Ibarra Sáiz, M.S., Ruiz Rube, I. (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)
11. Eclipse Foundation, http://download.eclipse.org/modeling/emf/emf/javadoc/2.11/org/eclipse/emf/ecore/package-summary.html
12. López-Fernández, J.J., Guerra, E., de Lara, J.: Assessing the Quality of Meta-models. In: Boulanger, F., Famelis, M., Ratiu, D. (eds.) MoDeVVa, vol. 1235, pp. 3-22. CEUR Workshop Proceedings, Valencia, Spain (2014)
13. Gerber, A., Raymond, K.: MOF to EMF: there and back again. In: Proceedings of the 2003 OOPSLA workshop on eclipse technology eXchange, pp. 60-64. ACM, (Year)
14. Chidamber, S.R., Kemerer, C.F.: A metrics suite for object oriented design. IEEE Transactions on Software Engineering 20, 476-493 (1994)