Wong, L.-H. et al. (Eds.) (2013). Proceedings of the 21st International Conference on Computers in Education. Indonesia: Asia-Pacific Society for Computers in Education

Understanding Software Ecosystems for Technology-Enhanced Learning – a Case Study

Oskar PETTERSSON, Jesper ANDERSSON & Marcelo MILRAD
Faculty of Technology, Linnaeus University, Sweden.
{firstname.lastname}@lnu.se

Abstract: The increased use of information and communication technologies(ICT) in schools promises up-to-date, interactive and collaborative learning content, However, this has proved difficult to fulfill as the requirements from students and teachers combined with devices in a variety of contexts are expensive to meet. Software reuse is a proven way to decrease development time.This paper explores the characteristics of a software ecosystem approach to cater for a new digital school and presents an enhanced reference model developed forthe field of technology-enhanced learning (TEL).

Keywords: Software Ecosystems, Software Ruse, Reference Model, TEL

1. Introduction

The deployment of ICT in schools and the demand for digital education have increased significantly in recent years (Ruth, 2010). Despite an abundance of open educational resources made available to teachers, the current educational practices supported by ICT are still in question with respect to sustainability and effectiveness (Dowries, 2007). Facilitating every student in almost every situation is the grand challenge, which implies that every digital learning resource requires a specific configuration per student, per course, per task, per context etc. In order to facilitate efficient development of software for this domain, various software reuse strategies have been adopted, for instance, Software Product Lines (SPL). The core of SPLs isinherently intra-organizational. This becomes a challenge inlarge domains with great variability. In such domains it is not feasible for a single organization to cater for all variations. To mitigate such risks there is a need for an inter-organizational approach to development. Software Ecosystem is one example that is better suited for such problem domains (Bosch, 2009). This work explores the characteristics ofanecosystem for pedagogical software with built-in support for collaboration and co-creation for different players, such as teachers and companies. The goal is to develop a foundation for more efficient, effective, and sustainable technology integration in schools.The contributions of this paper include current practices, identification of challenges, and a set of possible improvements. We use a three-layered perspective of ecosystems as a vehicle to analyze how a set of Swedish schools isusingICT to support education. The results are analyzed and we identify challenges fromcurrent practices. The outcomes of addressing these challenges are then concretized in three scenarios.These scenarios then serve as input to govern the extension of a reference model, which is the second core contribution. A reference model is a domain-specific ontology that packages the collective knowledge within an application domain. It transfers knowledge from previous to future projects in a domain assistingdevelopers. Finally, we conclude andidentify directions for future work.

2. Background

In the past, software applications where designed to fill a single purpose or function.As organizations became more reliant on systems and software, these applications had to be interoperable with other applications in order to function. The same line of development can be also seen in schools. In order to meet the needs of an organization, entire organizationmust be considered (Greefhorst & Proper, 2011). Enterprise Architecture (EA) is an approach that considers organizationsas systems and software (Zachman, 1987). One widely-used process framework for EA activities isThe Open Group Architecture Framework(TOGAF)(Josey, 2009). TOGAF can be used to describe a system from various focal points or “architecture domains” which are subsets of an overall enterprise architecture.These are (1) *Business*, which describes the processes that constitute the principal architecture drivers, (2) *Data*, which describes how data is accessed and stored, (3) *Applications*, which describes how individual applications are designed and how they interact with other applications and, (4) *Technology,* which describes the infrastructure that facilitate view1 - 3 in terms of hardware and software. In our studies we map the three topmost layers onto ecosystem layers.The fourth layer, technology, is not considered. This mapping is depicted in Figure 1. These layers have been considered independently in previous work. Liu (2010) defines the scaffolding for a content ecosystem, which we map the data level to. Jansen, Finkelstein, &Brinkkemper(2009) define software ecosystems, which applications map to. Moreover, Moore (1993)&Tian et al., (2008) define the Business ecosystem.

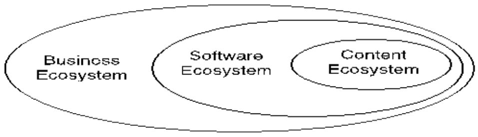
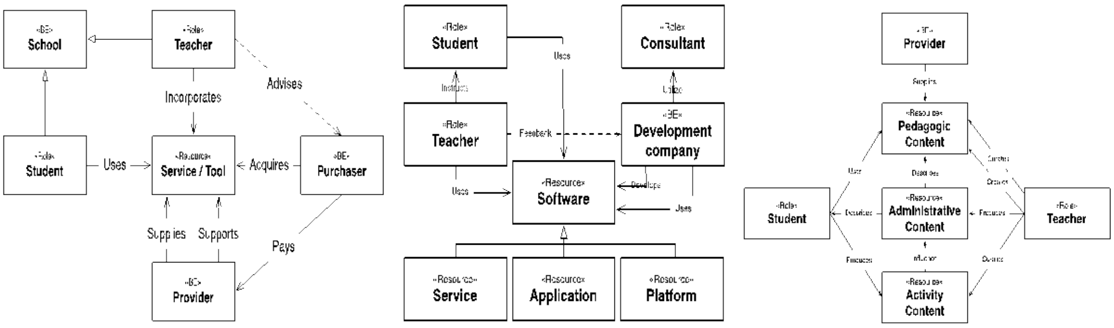


Figure 1. The layers of an Ecosystem.

This mapping was discussed with practitioners and analyzed in a case study (Pettersson et al.,in preparation)based on interviews with ten teachers and members of four companies. The analysis identifies three different views, which we describe in detail below using concept and interaction abstractions.



(A) The Business View(B) The Software View (C) The Content View

Figure 2. Three concept-interaction models for TEL ecosystem view-points

The three following views takes two perspectives, the teachers and the educational softwarecompanies. The first is the business view depicted in Figure 2A. A big challenge for ICT in schools isthat individual teachers cannot purchase material directly. This means that if a teacher cannot motivate for colleagues that there is anadded value, nothing will be purchased. With the hesitancy many teachers show for ICT,this becomes a significant problem for advancing its use in schools (Holloway, 2012). This is also reflected in the companies that say that they never sell to individual teachers. The companies sell their products as services and charges per annum per student, mainly directly to municipalities. Furthermore, the content providers are strongly depending upon external actors for providing anything not part of their core business.The software view depicted in Figure 2B,is concerned with services, applications, platforms, and reusable assets such as frameworks and components. From the teacher perspective, software may be divided into a three categories; (1) hardware (such as dedicated software for an interactive whiteboard), (2) specific subject software (such as dedicated physics software) and (3) communication. The common factor for all of these is that the teacher is free to promote which software students should use at their own discretion, if the school owns the software or the software is free. The teachers rarely provide feedback to the companies that develop the software. At the same time, they express a wish for being able to customize the softwarethey use. The development company initiates development of new products and hires consultants for a variety of tasks. The content view contains three categories depicted in Figure 2C, (1) *Pedagogic*, (2) *Administrative* and (3) *Activity* content. The pedagogic category includes assets used for teaching activities such as digitalized textbooks, videos and games. Content spans the range from the smallest building blocks to a complete digital course package. The case study indicates that the most common use case is teachers modifying existing pedagogic content. Administrative content is to a large extent meta-data. This category includes pedagogical plans, curriculums, or student reviews. The case study showed that the shear amount and verbosity of administrative content have steadily increased in recent years. The activity category includes assets that are produced as an outcome of learning activities and tasks, for example student authored media or tests. In this category, we also include feedback on activities, reviews or other metrics related to, for instance, performance. There was no significant evidence for social production communities within schools with the rare exception of a few wikis.

3. Towards Strategic Reuse

In the previous section,we pointed out several discrepancies between state of the art and reuse practices in schools. A lack of digital curation causes a difficulty for teachers to share digital content and collaborate despite customizing content is a very common use case. Furthermore, the acquisition of content is also inflexible to such a degree that teachers in many cases simply develop content themselves. There is no sign of systematic reuse, mainly due to the lack of widely adopted standards and patterns. In order to establisha sustainable approach to ICT, teachers and businessesrequire means to develop more effectively. Strategic reuse is a proven method to improve software development (Jacobson, Griss, & Jonsson, 1997). It is clear that strategic reuse is not about code, it is more about organization and processes (Frakes, 2005). This observation has resulted insoftware reuse methods with two parallel focuses, the domain and the product (Pohl et al., 2005). Strategic reuse and SPLs in particular is based in three hypotheses; (1) the redevelopment, (2) the oracle and (3) the organizational hypothesis (Weiss & Lai, 1999). The redevelopment hypothesis states that, software development is actually software redevelopment. The oracle hypothesis states that it is possible to derive future changes from previous for a software system. The organizational hypothesis states that it is possible to organize both software and organization in such a way that it benefits from predicted future changes. In order to establish a successful strategic reuse program all three must be satisfied. We define three reuse usage scenarios. These, presented as challenges and goals,demonstrate what is required to satisfy these hypotheses in TEL.

3.1 Usage Scenarios

The usage scenarios where derived as a vehicle to identify the existing gap between current practices and state-of-the art, strategic reuse practices. The scenarios outline threechallenge-goal pairs that identify gaps and highlight current impedimentsfor adopting strategic reuse practices in the TEL field.

Table 1: Usage scenarios – Challenges and Goals.

ID	Problem	Challenge	Goal
S1	Teachers find pedagogical content inflexible and difficult to adapt. (redevelopment satisfied)	Satisfy hypotheses (1) and (2). E.g., identify and introducevariation points in pedagogical content and the means required by teachers to extend and adapt it.	Have pedagogical content with explicit variability. Define and provide means to use, modify and extend pedagogical content.
S2	Collaboration between stakeholders is perceived as difficult. (redevelopment and oracle satisfied)	Satisfy hypothesis (3). Define interfaces and methods for interchange of development assets between stakeholders	Have functionality that enables cooperation with support for access control,contributionand governance.
S3	Teachers lack best practices and guidance.(redevelopment and organizational satisfied)	Satisfying hypothesis (2) by enabling the formalization of practices within the platform.	Have activities and artifacts in the platform that utilize knowledge transfer.

3.2 Scenario 1 (S1) – Teacher reusing and extending content

In order to perform this activity, the teacher logs into the platform and browse the repository foran activity that matches the teacher's needs. The system presents two activities to the teacher that collectpH with the particular devices available at the school. The teacher walks through the processes inthe activity, customizes them fit the location where the class will take place through an authoring interface. The following day, the students perform the activity collecting measurements and the data is made available for classroom analysis.

3.3 Scenario 2 (S2) – Software companies collaborating with Teacher

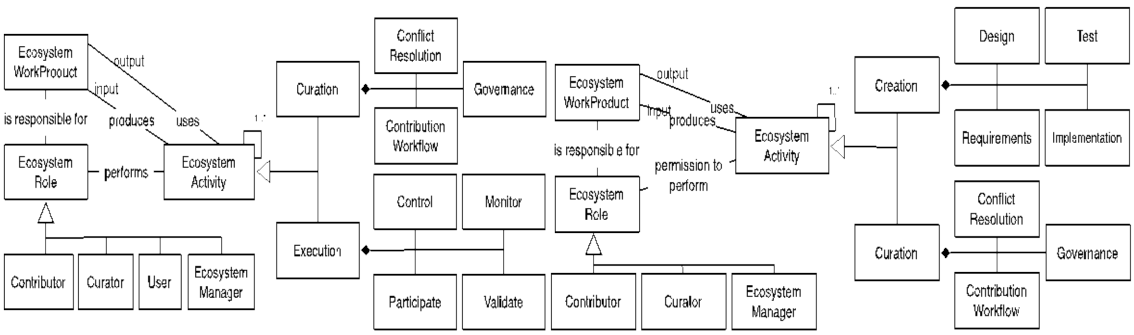
The company wants to start developing a new software product in the subject of biology. The envisioned product is a mobile application and they have gathered requirements and devised a product design internally. However, the company does not have sufficient resources and skills to manage the pedagogical content. For this, the company has planned to find interested teachers in the ecosystem's network and announces a request for participation. This request contains a description of the project, work descriptions with process models that model how the contributions will fit into the project context.

3.4 Scenario 3 (S3) – Teacher models a learning process

The teacher deploys the activity, distribute the roles and material to the students and begin to follow the suggested workflow. The teacher finds that the roles suggested by the activity make the groups slightly to large, which leads to students not being active in the groups. The teacher makes a note for later reporting and adjusts the groups. When the activity is concluded he teacher modifies the workflow to reflect the practical changes. The teacher then commits the changes to the platform where the author gets notified and other teachers can access the changes.

4. A Software Ecosystem Reference Model for the TEL domain

Table 1 describes three goalsfor supporting successful strategic reuse in the TEL domain. These goals are related to (1) *the ability to extend and reuse content and components*, (2) *extending development processes between organizations*, and (3) *the ability to formalize processes*. We find that SPLs could provide a way to achieve the goal of S1, providing the ability to extend and reuse content and components. However, SPLs fall short in respect to goal 2 due to the lack of inter-organizational focus. Comparing the required mechanisms with an existing reference model (Pettersson et al., 2010) for the domain, we find that the existing model falls short.The model is lacking detailed support for collaboration between the stakeholders as well as facilitating curation of ecosystem assets. This section describes the enhancements made to meet each goal and combine them together into an enhanced reference model.



(A) Extension for reuse (B) Extension for inter-organizational development

Figure 3. Extensions

Goal 1 requires two groups of mechanisms, execution and digital curation, depicted in Figure 3A. Execution is related to how various executable artifacts are processed in the ecosystem. For this, we turn to the process enactment language xSPIDER_ML(Portela et al., 2012). For our purposes, we reduce the language into four core concepts *Control, Monitor, Participate and Validate*. *Control*introduces mechanisms to influence the execution of an artifact through the provisioning of information, for example, providing sensor data, answers to questions, and rich media content like pictures and videos. *Monitor* provides mechanisms for observing and supervising artifacts that are executed in the ecosystem. *Participate* provides mechanisms that allow multiple users to control the execution of artifacts. The final concept, *Validate*, provides mechanisms for validation of the executable assets including the external inputs and control directives they receive. The second group, digital curation, is concerned with the artifacts' life cycles in an ecosystem(Liu, 2010). Digital curation relies on three concepts; *Contribution Workflow*, *Conflict Resolution* and*Governance*(Lam, 2012). Contribution Workflow determines how tasks are distributed among roles including related pre and post activities after the creative phase. Conflict Resolution is related to how the ecosystem managesdisagreements in the curation activities. Governance is related to an ecosystem's policies and regulations.It extracts rules and policies defined by the first two groups. On top of this it also defines obligations and consequences of violating the rules.Goal 2 is achieved by software product lines, more specifically a two-tier lifecycle model (Pohl et al., 2005). This outlines four distinct development activities*Requirements, Design, Implementation* and *Test*, all regular software development activities.To support inter-organizational development, mechanism that manages interactions between independent entities is required. To achieve this, we refine digital curation concepts digital curation described above.

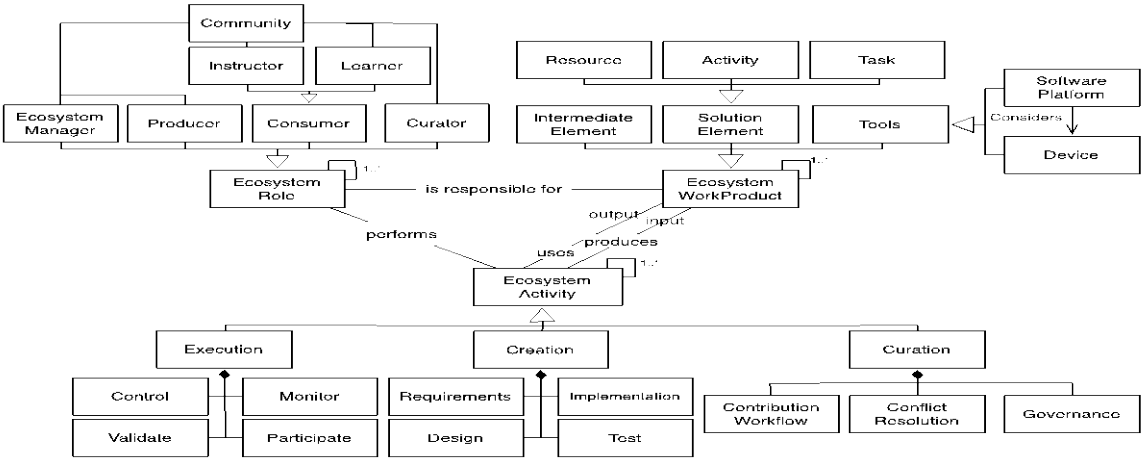


Figure 4. A software ecosystem reference model for the TEL domain

We merge the extensions into a single reference model, depicted in Figure 4. The concepts are subtypes of the Ecosystem Role and Ecosystem Activity. In terms of Ecosystem Roles there are four new concepts, (1) Ecosystem Manager, (2) Producer, (3) Consumer and (4) Curator. Ecosystem Manager is essentially an extension of the product manager from a SPL, with added responsibilities for Ecosystem. Producer can be anything from a company contributing an entire platform to a photographer providing a picture. Consumersare regarded as end users in the ecosystem, e.g. students or teachers using assets. Curator is a local ecosystem manager but for a small scope of the ecosystem and the caretakers that sort, filters and marks the assets of the ecosystem. Furthermore, the Ecosystem Activity has three new concept groups, Execution, Creation and Curation, which were also discussed above.

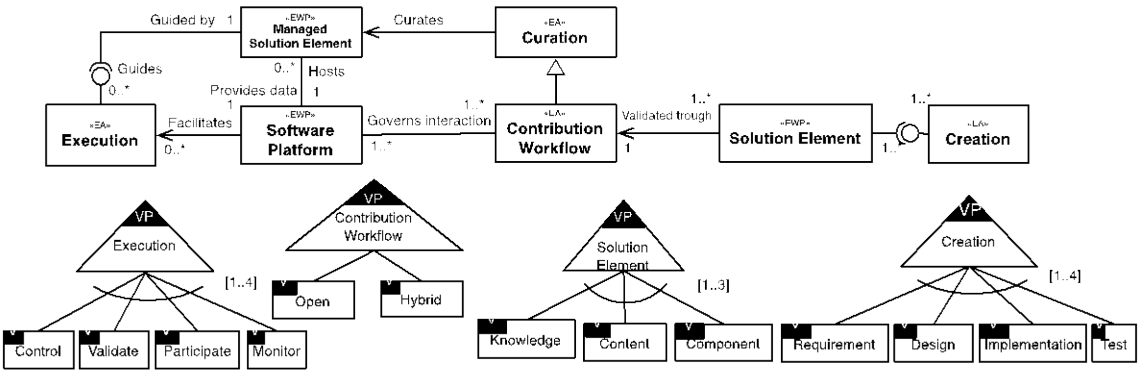


Figure 5. Instantiation of the reference model

In order to initially evaluate theenhanced reference model, we instance it with variation points in Figure 5 and perform a brief scenario walkthrough of S1 – S3. The Ecosystem Activities (EA) and Ecosystem WorkProducts (EWP) are derived from the reference model. Offering managed solution elements and execution trough the software platform as reusable artifacts fulfills S1 together with extension of content trough the contribution workflow and creation with their possible variations. S2 is satisfied by the contribution workflow and creation, where the contribution workflow can be either open or a hybrid. S3 is satisfied by creation of solution elements and they being able to be knowledge and distributed trough the software platform.

In order to firmly validate the reference model, the collection of more empirical data is needed from schools and companies. The usefulness of the model increases for every use of it, so the practical usability of the model relies on organizations and individuals attempting to implement it in their daily work.

5. Concluding remarks and Future Work

This paper presents three views of TEL ecosystems and usesthem to analyze the technology and software used in some Swedish schools and educational software companies. Based on a previous case study, we present abstractions of Business, Software and Content views and we identify problems, challenges and goals, and construct usage scenarios to demonstrate the gaps in functionality. We propose two extensions to an existing reference model to provide support for reuse and inter-organizational development processes. The extensions are merged with the original reference model. Future research includes an extended validation of the reference. Furthermore, the methodology must be enabledto companies and schools.Future efforts will be directed to this specific line of research.

References

Bosch, J. (2009). From software product lines to software ecosystems. *Proceedings of the 13th International Software Product Line Conference* (pp. 1–10). San Francisco, California: Springer LNCS.

Dowries, S. (2007). Models for Sustainable Open Educational Resources. *Interdisciplinary Journal of Knowledge and Learning Objects*, *3*, 29–44.

Frakes, W. B. (2005). Software reuse research: status and future. *IEEE Transactions on Software Engineering*, *31*(7), 529–536. doi:10.1109/TSE.2005.85

Greefhorst, D., & Proper, E. (2011). *Architecture Principles: The Cornerstones of Enterprise Architecture*. Springer Berlin.

Holloway, J. K. (2012). *Teachers' Concerns, Attitudes, Beliefs, and Pedagogical Practices as it Relates to Technology Integration*. Auburn University.

Jacobson, I., Griss, M., & Jonsson, P. (1997). *Software reuse: architecture, process and organization for business success*. New York, NY, USA: Addison-Wesley Publishing Company.

Jansen, S., Finkelstein, A., & Brinkkemper, S. (2009). A sense of community: A research agenda for software ecosystems. *31st International Conference on Software Engineering, New and Emerging Research Track*.

Josey, A. (2009). *TOGAF Version 9: A Pocket Guide*. Van Haren Pub.

Lam, S. (2012). *Collaborative Curation in Social Production Communities*. University of Minnesota.

Liu, S. (2010). Trends in Distributed Curatorial Technology to Manage Data Deluge in a Networked World. *The European Journal for the Informatics Professional*, *XI*(October), 18–24.

Moore, J. (1993). Predators and Prey: A New Ecology of Competition. *Harvard business review*, *71*(3), 75–86.

Pettersson, O., Davidsson, M., Andersson, J., & Milrad, M. (In preparation). Exploring a Selection of Swedish ICT Approaches from the Perspective of ICT-Teachers. *IEEE Transactions on Learning Technologies*, *To be subm*.

Pettersson, O., Svensson, M., Gil, D., Andersson, J., & Milrad, M. (2010). On the role of software process modeling in software ecosystem design. *Proceedings of the Fourth European Conference on Software Architecture: Companion Volume* (pp. 103–110). Copenhagen: ACM.

Pohl, K., Böckle, G., & Van Der Linden, F. (2005). *Software product line engineering: foundations, principles, and techniques*. Springer-Verlag New York Inc.

Portela, C., Vasconcelos, A., Silva, A., Silva, E., Gomes, M., Ronny, M., Lira, W., & Oliveira, S. (2012). xSPIDER_ML: Proposal of a Software Processes Enactment Language Compliant with SPEM 2.0. *Journal of Software Engineering and Applications*, *05*(06), 375–384. doi:10.4236/jsea.2012.56044

Ruth, S. (2010). Is E-Learning really working? The trillion-dollar question. *Internet Computing, IEEE*.

Tian, C. H., Ray, B. K., Lee, J., Cao, R., & Ding, W. (2008). BEAM: A framework for business ecosystem analysis and modeling. *IBM Systems Journal*, *47*(1), 101–114. doi:10.1147/sj.471.0101

Weiss, D. M., & Lai, C. T. R. (1999). *Software product-line engineering: a family-based software development process*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.

Zachman, J. a. (1987). A framework for information systems architecture. *IBM Systems Journal*, *26*(3), 276–292. doi:10.1147/sj.263.0276