

RESEARCH

Open Access

A study on dynamic aspects variability in the SOLAR educational software ecosystem



Emanuel F. Coutinho^{1,2,*}  and Carla I. M. Bezerra^{1,2†}

Abstract

A Software Ecosystem (SECO) refers to a collection of software products with some degree of symbiotic relationship. SOLAR is a Virtual Learning Environment (VLE) that enables the publication of courses and interaction with them among its various users. In this context, SOLAR SECO emerges, where diverse situations of software evolution and maintenance are part of its development process. The aim of this paper is to discuss the dynamic variability of SOLAR educational software ecosystem and software modeling. As an example, dynamic variability aspects of the feature model of SOLAR VLE discussion forum functionality were discussed, one of the most widely used services within SOLAR SECO. As a major conclusion of this work, we identified that the use of the contextual feature diagram allows the study of the dynamic aspects of a system, even more supported by tools to support automatic measurement collection .

Keywords: Software ecosystems, Virtual learning environments, Dynamic aspect, Variability, Feature diagram, Discussion forum

Introduction

According to the International Standard ISO/IEC/IEEE 42010, a software intensive system is a system in which software essentially influences the design, construction, deployment, and evolution of the system as a whole to cover individual applications, subsystems, system of systems, product lines, product families, entire companies, and other aggregations of interest [1]. The complexity of software and the complexity of software-dependent systems have grown at an intense pace [2]. In particular, software intensive systems have been rapidly evolved from standalone systems in the past, to be part of networked systems in the present, and to become system of systems in the near future.

Traditionally, a Software Ecosystem (SECO) refers to a collection of software products with some degree of symbiotic relationship [3]. A SECO can also consist of a set of actors acting as a unit that interacts with a distributed market between software and services, along with

the relationships between these entities [4]. Such relationships are often supported by a technological platform or a common market and carried out by the exchange of information, resources, and artifacts.

Manikas and Hansen [5] defined SECO as an interaction of a set of actors on a common technological platform that results in a number of software solutions or services. Each actor is motivated by a set of interests or business models, and he/she is connected to the other actors and SECO as a whole through symbiotic relationships. In turn, the technological platform is structured to allow the involvement and contribution of different actors. The ecosystem metaphor highlights external or unknown actors who are contributing to evolve a common technological platform by shifting the traditional organization-centric value chain to a software delivery network where multiple components developed on different platforms coexist and affect business from the buyer [6]. Another definition of SECO is the interaction of a set of actors on top of a common technological platform, resulting in a number of software solutions or services [7]. Finally, Manikas [8] updated the definition of SECO as “the software and actor interaction in relation to a common technological infrastructure, that

*Correspondence: emanuel.coutinho@ufc.br

[†]Emanuel F. Coutinho and Carla I. M. Bezerra contributed equally to this work.

¹Federal University of Ceará (UFC), Quixadá, Brazil

²Graduate Program in Computer Science (PCOMP), Quixadá, Brazil

results in a set of contributions and influences directly or indirectly the ecosystem.”

Regarding the concept of software ecosystem (SECO), the terms technological platform, central platform, and software platform will have the same meaning. All refer to products of interest, that is, the main software in the business model, as presented in Boucharas et al. [6]. We will use the “technological platform” expression to standardize the reading.

Virtual Learning Environments (VLEs) integrate Information and Communication Technology (ICT) to create Internet-based environments that enable the process of knowledge construction and autonomy by their stakeholders [9]. In VLEs, it is possible to use various learning mechanisms, such as class visualization, chats, message exchanges, and forums [10]. However, the use of these mechanisms does not imply real learning. Educators now have access to better and modern technologies [11]. However, despite the available technologies, there are a range of factors that influence resource utilization, such as processes, strategies, motivations, and culture.

Because of these complexities faced by educators today, designing any e-learning system should start with adopting a sustainable model [11]. To this end, an e-learning ecosystem model has been proposed. This model is comprehensive and capable of handling the use of new technologies and tools, incorporating new learning approaches, adaptable to a variety of learning styles, and being sensitive to learning conditions.

SOLAR (Online Learning System) is a VLE whose participation model is oriented to the teacher and the student, enabling the publication of courses and interaction with them. It currently has web, mobile (Android and IOS), and MOOC (Massive Open Online Course) versions. Additionally, there is the SOLAR educational SECO [12], with various integrations between development, society, research, and business. The terms educational software ecosystem and e-learning SECO will have the same connotation. We will use the “educational software ecosystem” expression (or educational SECO) to standardize the reading.

For SOLAR SECO, several evolution and maintenance situations are part of their daily lives. Motivated by a lack of documentation or updating, requirement changes impact all SECO technological platforms. These changes, whether due to identified defects or platform customizations to users or to fit development platforms, would have a beneficial effect on the development environment if they were modeled, documented, and analyzed for the impact of their implementation in different contexts.

Software Product Line (SPL) companies increasingly expand their platform outside their organizational boundaries, in effect transitioning to a software ecosystem

approach [13]. There are, at least, two reasons why a company would be interested in moving towards a software ecosystem. First, the company may realize that the amount of functionality that needs to be developed to satisfy the needs of their customers is far more than what can be built in a reasonable amount of time and with a research and development investment that offers an acceptable return on investment. Secondly, the mass customization trend drives the need for a significant research and development investment for successful software applications. Especially on the web and mobile devices, users demand a significant degree of customization or even compositionality that allows each user to create a potentially unique configuration that addresses her or his specific needs and desires. Extending the product with externally developed components or applications provide an effective mechanism for facilitating mass customization.

One way to model the dynamic aspects is to use the feature model of a Dynamic Software Product Line (DSPL). According to Pascual et al. [14], a DSPL is a single system that is able to adapt its own behavior at runtime. The feature model represents domain features and line variability. The features describe the features and quality characteristics of a software product [15]. One way to represent the dynamic aspects in a feature model is by modeling the context adaptations [16]. Thus, this work will use the feature model with context to model the dynamic aspects of SOLAR.

The idea of this work is to study variability to have a software that can adapt to user needs or resource constraints. This adaptation is possible by postponing the resolution of variability to the execution time. In this case, the resolution occurs at the moment the software or system is started or during its execution.

The main objective of this work is to discuss dynamic aspects of software modeling of SOLAR educational SECO. As a secondary objective, we intend to present an example focusing on the discussion forum functionality of SOLAR VLE, one of the most used services within SOLAR SECO.

In the context of SECO, some research opportunities arise. SOLAR SECO is currently static from an evolutionary point of view. This is due to the fact that for new functionalities, configuration or customization is required. The platform is changed, tested, and deployed in a static manner, and not at runtime. There are also no runtime adaptations, depending on user contexts. Thus, opportunities and benefits of dynamic changes become attractive to this context.

Considering that this work aims to study variability, we proposed the following research questions:

Research question 1 (RQ1) —How to model dynamic aspects in an educational software ecosystem?

Research question 2 (RQ2) —How to evaluate from measurements the modeling of dynamic aspects of an educational software ecosystem?

As a major conclusion of this work, we identified that the use of the contextual feature diagram allows the study of the dynamic aspects of a system, even more supported by tools to promote automatic measurement collection. Practitioners and researchers can use this work to evaluate the variability of situations that can be complex to study in the real world. Moreover, a feasibility study can be carried out through similar ideas, and to analyze different contexts for supporting in decision-making and systems improvement. This article is an extended version of the proposal presented by Coutinho and Bezerra [17], but with more details, analysis, and discussions of the results.

As answers to the research questions we have:

RQ1 —Initially, it is necessary to design or model the educational contexts to be analyzed, and then to standardize their description. Context constraints must be added to define contextual adaptations. Tools to support modeling, inclusion of constraints, and collection of metrics must be used to support the process. In our work, we used the feature diagram and the S.P.L.O.T and DyMMer tools. Once modeled, metrics are collected to measure the variability of contexts.

RQ2 —Once the educational contexts are modeled, metrics for the analysis of variability are collected and the results interpreted. In our work, we used the DyMMer tool to collect metrics. Eight specific metrics for dynamic product lines were analyzed, indicating the level of dynamics, which in our case was average.

In addition to this introduction, the article is divided into the following sections: the “**Background**” section introduces the key concepts related to work; in the “**Study design**” section, the methodology adopted in the paper is presented; the “**Results**” section presents the results of this work with modeling and data collection; in the “**Discussion**” section, general analysis is discussed; the “**Threats to validity**” section presents some threats to the validity of this work; the “**Related work**” section shows the related work; and finally, the “**Conclusions**” section presents the conclusions and future work.

Background

This section presents the concepts used in the study for supporting this work.

SOLAR Virtual Learning Environment

SOLAR is a Virtual Learning Environment (VLE) designed to enable the creation of a virtual space for face-to-face or semi-face courses, and it is based on the use of free software in an architecture integrable with other environments. Its major and most complete version is a web application. It currently has about 47,000 registered users, with an average of 2000 daily hits. It serves as a focal point for the creation of Blended Education, that is, the blend of characteristics of both modes of education, to form a new educational model that uses ICTs heavily.

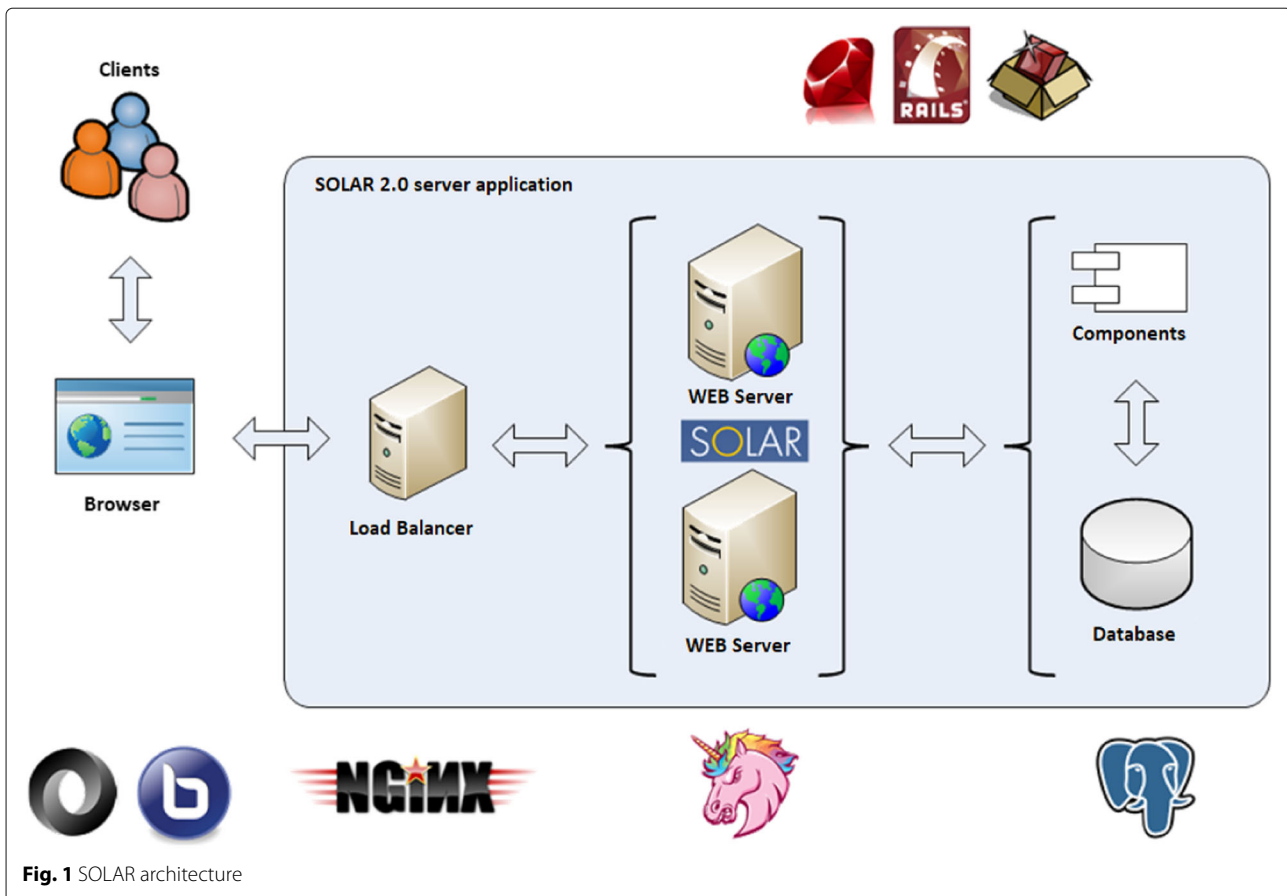
Blended Education or Blended Learning is a derivative of e-learning, and it refers to a training system where most content is transmitted remotely, usually by the Internet. However, it necessarily includes face-to-face situations, hence the origin of the designation “blended,” something mixed, combined. It can be structured with synchronous or asynchronous activities in the same way as e-learning, in situations where teacher and students work together at predefined times, or not, with each other fulfilling their own needs, with each one doing their tasks at flexible times.

SOLAR VLE uses a client server architecture, and it was designed with the philosophy of free software. Its main software platforms are Ruby on Rails (development technology), several auxiliary GEM (components) to automate development, PostgreSQL (database), Nginx (Web Server), Unicorn (Rack Web Server), and Linux (Operating System). The Client side basically consists of a web browser, HTML and Javascript. The Server side consists of the web server, software components, and database. Figure 1 shows the SOLAR architecture in its web version.

SOLAR VLE consists of the following modules: module integration core, user management and access control, VLE integrator with the university’s academic management system, interagent personal space, evaluation and monitoring, content authoring tool, communication tools, content tools, administrative tools, course tools, and collaborative and cooperative tools. Some features are shown in Figs. 2 and 3. Figure 2 shows the web version for SOLAR VLE. Figure 3 shows the Android mobile version for SOLAR VLE, only in Brazilian Portuguese version.

SOLAR VLE is mainly available in its web version, which is its most common use. However, two web versions exist. These two versions differ mainly by the amount of new features added and the completely redesigned graphical interface. The older web version was used by the Open University of Brazil (UAB) and is currently used only for historical data retrieval from old databases. The Open University of Brazil (*Universidade Aberta do Brasil - UAB*)¹ is a system integrated by public universities that

¹Open University of Brazil (Universidade Aberta do Brasil - UAB) <https://www.capes.gov.br/uab>



offers higher level courses for the population with difficulties in accessing university education, based on the use of distance education. The current web version is considered the official version of SOLAR, currently used in training, UAB, and classroom courses. It is the SOLAR VLE evolution in terms of technology and usability. There is also the mobile version of SOLAR, for iOS and Android, which has reduced functionality, as well as the MOOC (Massive Open Online Course) version of SOLAR.

Discussion forum

SOLAR VLE has many features. To perform an analysis of platform dynamic aspects, the discussion forum functionality was chosen. This choice was due to the fact that this is a feature common to any VLE, and it has a very high level of use by its users. In addition, it is possible to evolve the features of the discussion forum, often from user or research demands.

The SOLAR VLE discussion forum is one of the services most used by its users, both in the web and mobile applications. In addition to having traditional discussion forum features such as posting, replying to, chronologically organizing, and attaching files, it has some additional

functionality, such as text-to-audio transcription and vice versa.

Commonly, some elements of research and user requests/demands are registered. Several researches are carried out on the SOLAR VLE. Elements of research are new features that can be added to SOLAR VLE that have emerged from research, such as text-to-speech, usability studies, and virtual laboratories.

There is a functionality in the environment for the registration of user demands, so that he himself can suggest new features in the application. These new features, added to research ideas that can be incorporated into SOLAR VLE, may eventually compose new versions, according to the analysis of feasibility and benefits by the SOLAR VLE development team.

These elements are also desirable for distance learning users, such as tutors and course coordinators, and they are specific to the distance education context. These features would be automatic correction or analysis of discussion forums, tips, and accounting for metrics. In addition, general features such as language changing would also be interesting to make available.

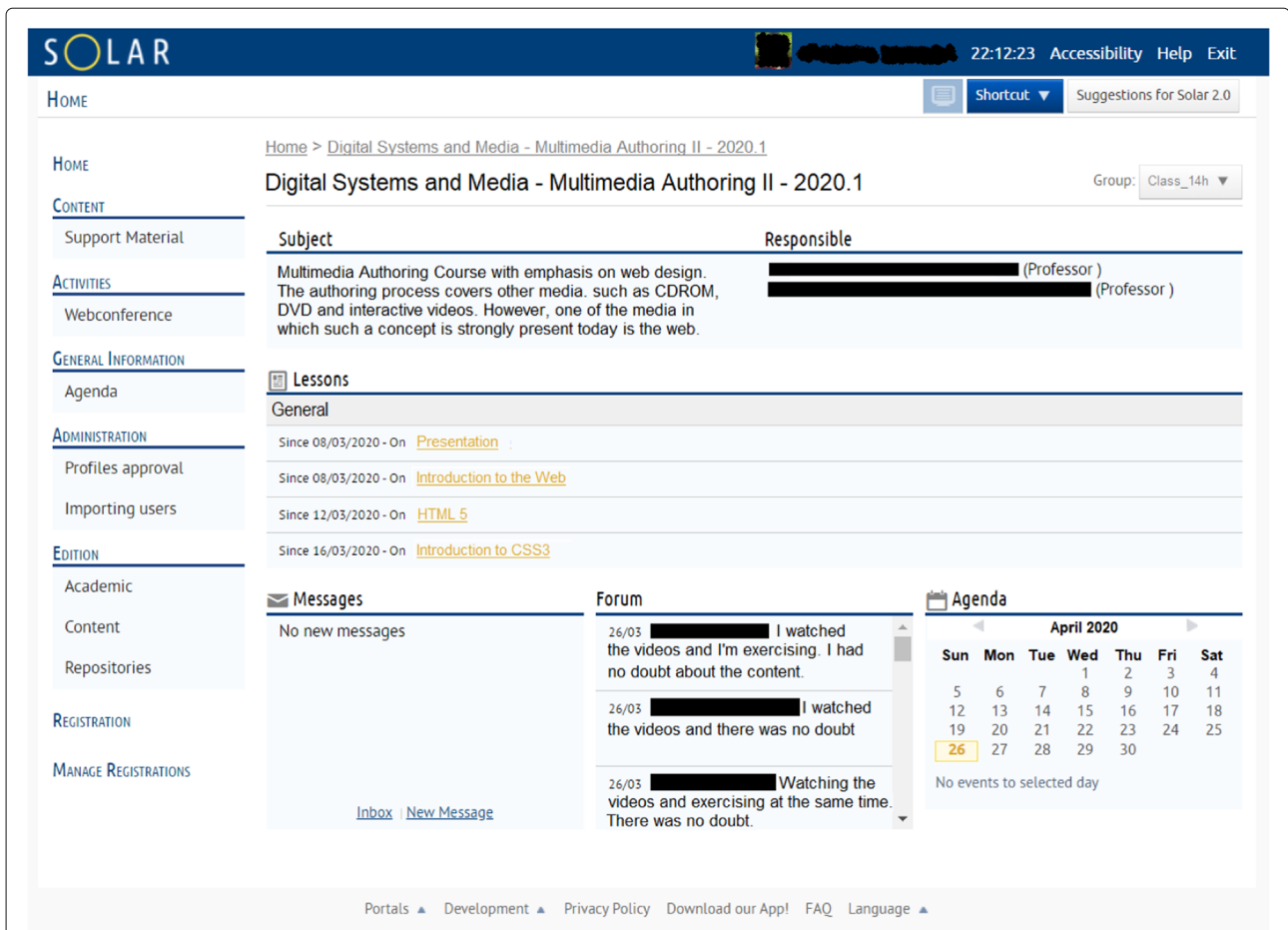


Fig. 2 Example of a screen with SOLAR VLE web functionality (translated from Brazilian Portuguese)

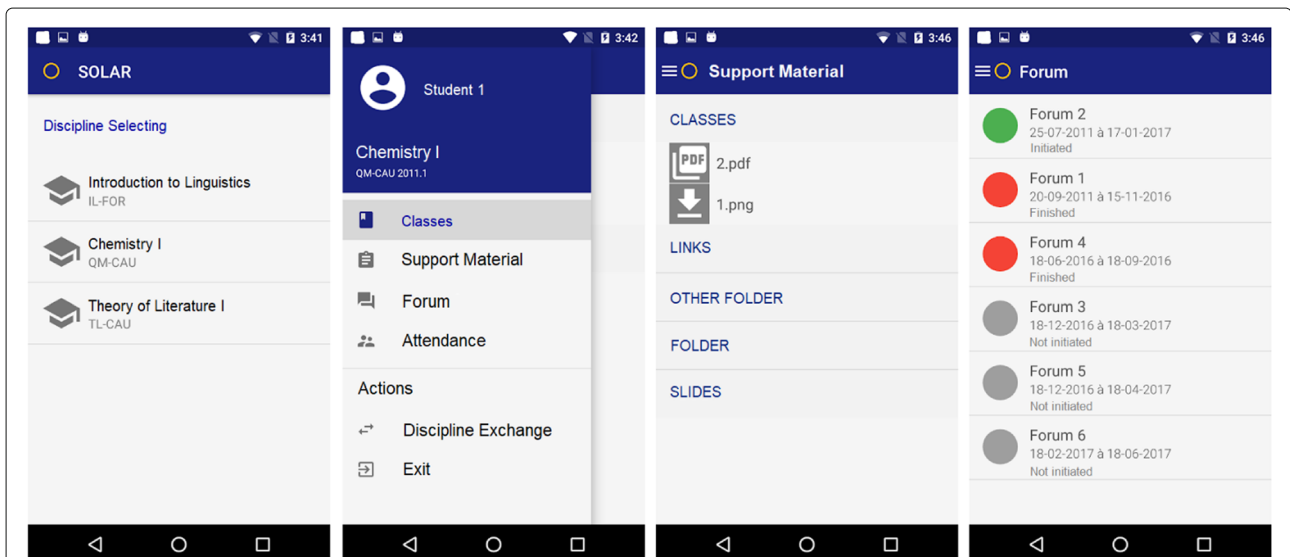


Fig. 3 Example of a screen with SOLAR VLE mobile functionality (translated from Brazilian Portuguese)

The impacts of evolutions in these features for SOLAR VLE are environment improvement, agility in operations and processes, faster feedback for users, and adequacy to customer needs. SOLAR VLE generates a lot of educational data. With these data, some aggregate systems were developed, such as class allocation systems, finance, and logistics. These aggregate systems use the data from SOLAR VLE to generate new information, used by other roles in SECO. The impacts on the evolution of SOLAR VLE functionalities also influence these aggregate systems.

SOLAR SECO

Around SOLAR, a set of relationships has been formed between users, technology vendors, solution developers, and business relationships [12]. Several systems were developed around the technological platform, and many versions and maintenance also emerged. Providing an API for building platform solutions also contributed to the integration and diffusion of the environment. In this context, SOLAR SECO emerged.

The SOLAR SECO is composed of a set of elements that communicate at different levels [12]. The SOLAR VLE is the basis of the ecosystem, being the technological platform that supports SECO. These elements involve different institutions, producing or receiving information, supported by different technologies.

A Software Supply Network (SSN) is a series of connected software, hardware, and services that cooperate to meet the demands of the market [18]. Figure 4 presents an SSN modeling for SOLAR SECO, based on Boucharas et al. [6] with the addition of the Aggregator element, proposed by Costa et al. [19]. SOLAR and its products are the Company of Interest, supported by various types of vendors: software, hardware, development teams, databases, and other systems. As Intermediaries, we have the SOLAR developer team, app stores, and a research committee. There are many types of Clients in SOLAR SECO, from institutions such as universities, platform users, researchers, management systems, to external developers, who in turn have their own clients. The extension of the proposed notation proposed the role of the

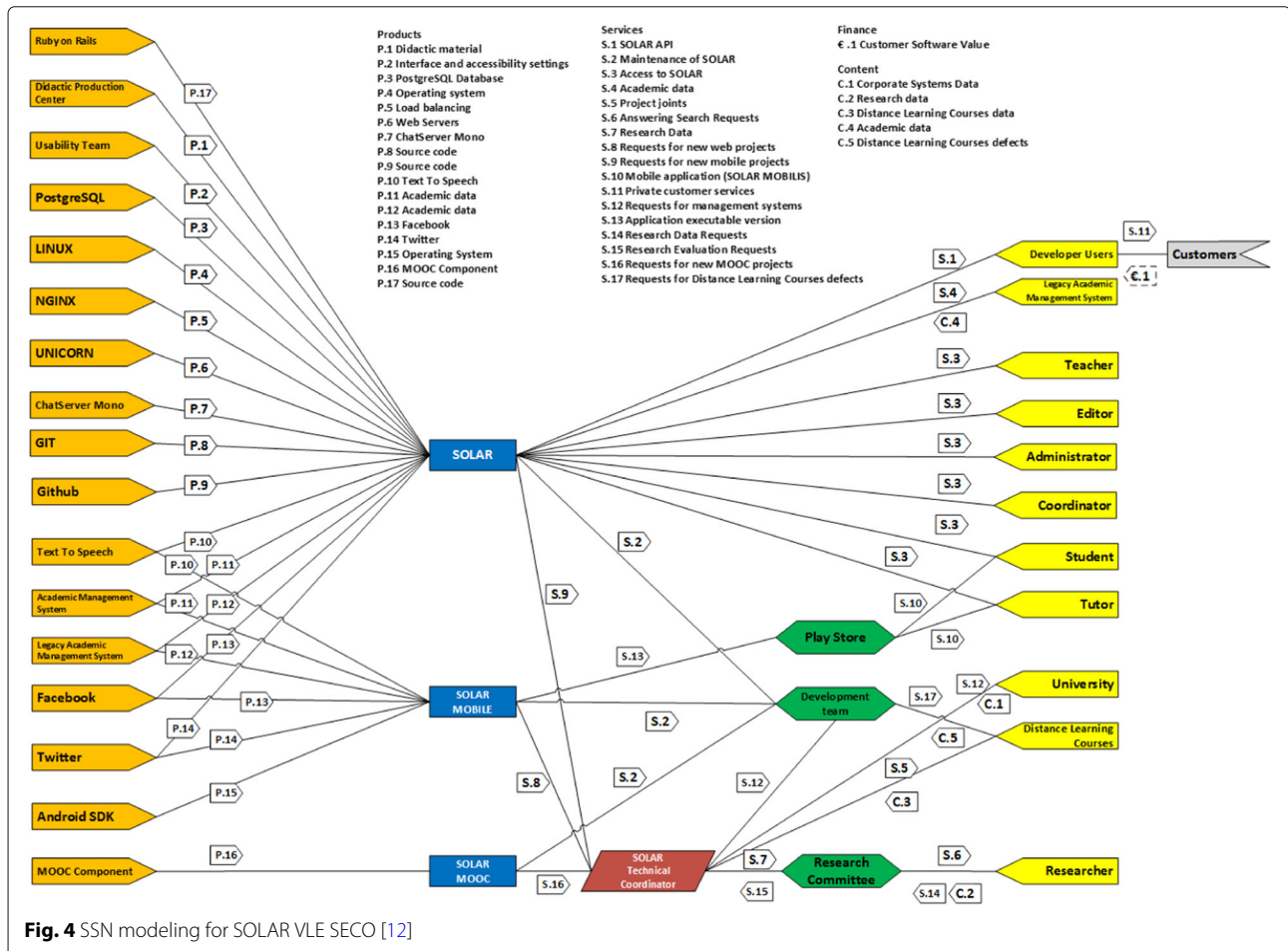


Fig. 4 SSN modeling for SOLAR VLE SECO [12]

Aggregator, which in the case of SOLAR SECO is represented by the SOLAR Technical Coordinator. In addition to managing SOLAR development, SOLAR Technical Coordinator is responsible for brokering new products and services, aligned with business needs, adding value and aggregating the opportunities that arise from using the platform.

Dynamic variability

With the SPL popularity in industry, new design and development challenges for complex, pervasive, autonomous, and embedded systems are demanding for new solutions [20]. Nowadays, systems with adaptive and contextaware architectures, such as autonomic and ubiquitous computing systems and software ecosystems, require dynamic capabilities to handle runtime needs. Many application domains require runtime capabilities to address self-management, autonomy, reconfiguration, and flexible configurable options that static variability models cannot handle. Thus, this trend requires SPLs to become more adaptable, in other words, dynamic.

Gurp et al. [21] characterized feature as a logical unit of behavior that corresponds to a set of functional requirements and quality requirements, with an *n:m* relationship between features and requirements. This means that a particular requirement can be applied to multiple features

and a particular feature can cover more than one requirement. Although feature models are studied in the domain engineering of a SPL, these information models can be used for other areas ranging from requirement gathering to data model structure. The structure of a feature model is represented as a hierarchical set of features composed of [15] (i) relationships between a parent feature and its daughter features (or subfeatures) and (ii) constraints (or cross-tree) that typically include or exclude statements of features.

Figure 5 presents a feature model example that represents context features for the systems domain Smart Home [22]. In a given current configuration of the feature model, the active and inactive features of a given context are represented, as well as the constraints, relationships, and optional features of the feature model.

In the example of Fig. 5, system control features can be enabled through the mobile phone. If the user wants to use video on demand feature, the current setting is not valid until video on demand enables the internet setting (which in the example is disabled). Consequently, a configuration explanation operation uses diagnostic techniques to determine which features should change their state to achieve a valid product configuration at run time. In this situation, it is possible to obtain four possible reconfigurations, one for each type of Internet connection: WiFi-b/g,

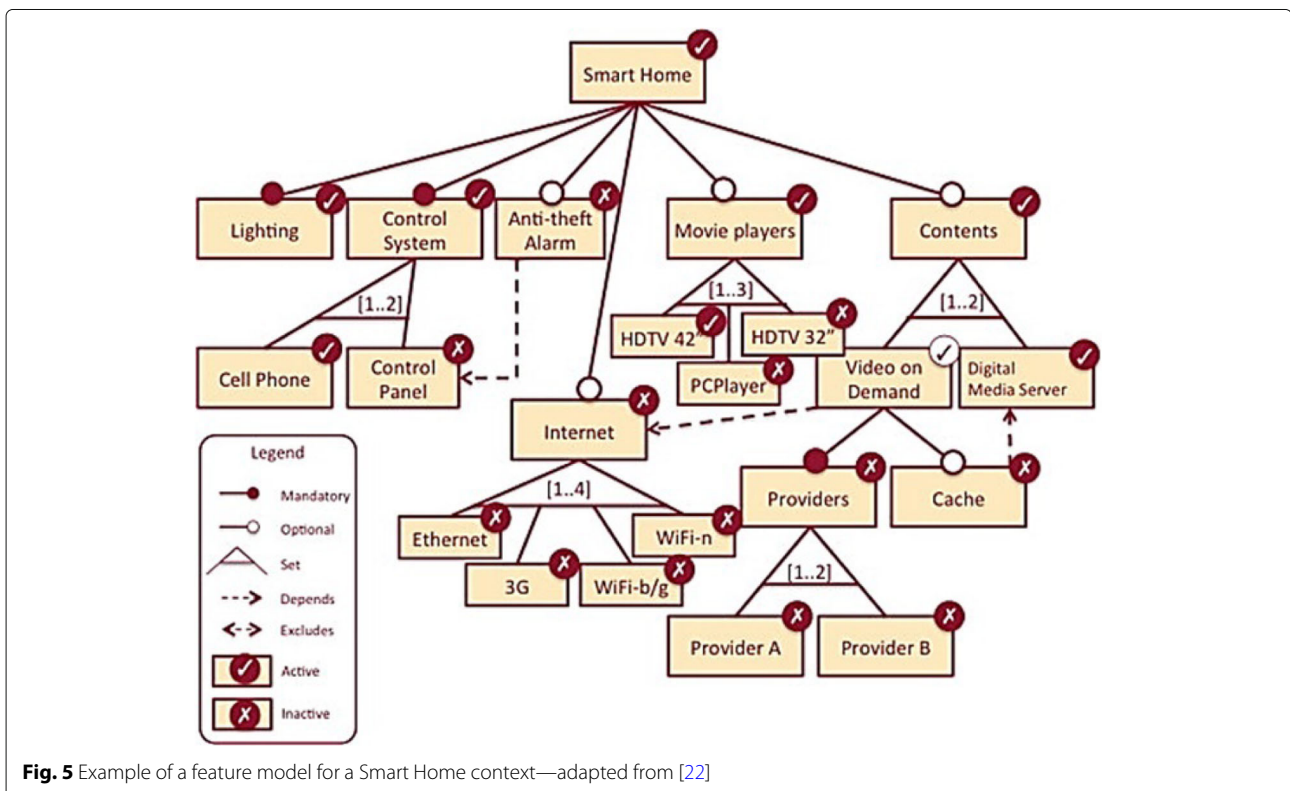


Fig. 5 Example of a feature model for a Smart Home context—adapted from [22]

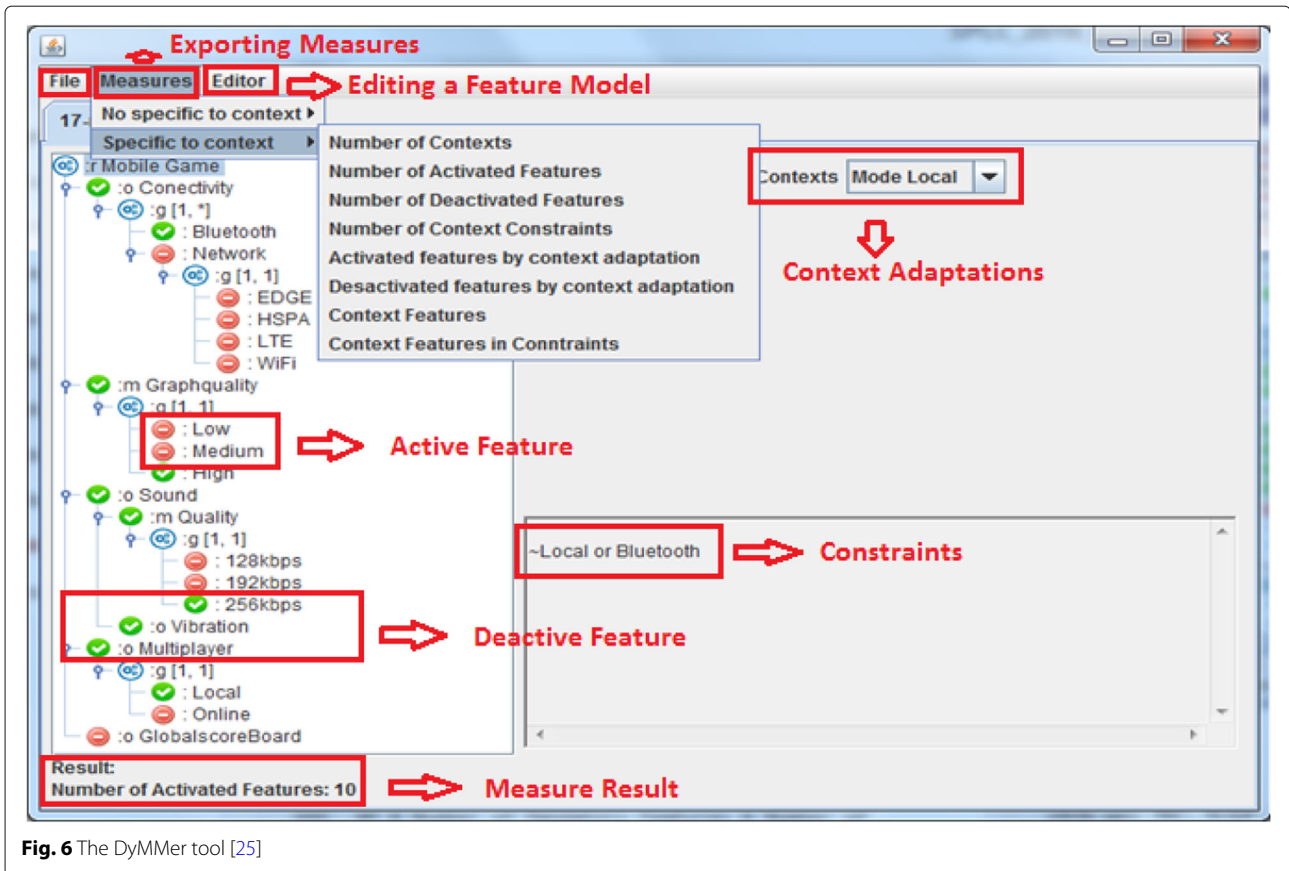


Fig. 6 The DyMMer tool [25]

Wi-fi-n, Ethernet, and 3G protocols. The tiebreaker criteria for choosing the best configuration may be based on quality attributes required by the desired user or system settings, such as available bandwidth rate. Since the algorithm determines the optimal or correct configuration, features can change their values to valid states once the constraints and dependencies have been satisfied.

There are several tools for feature modeling, but in the literature only three for dynamic feature modeling have been identified: MOSKitt4SPL [23], VariaMos [24], and DyMMer [25]. However, only DyMMer was available. Only one feature model repository has been identified, S.P.L.O.T. [26].

The DyMMer (Dynamic Feature Model tool based on Measures) tool was designed to extract measurements from feature models that can be described in S.P.L.O.T. format, and it supports feature model template editing and the automatic extraction of quality measures [25]. DyMMer² collects a large number of quality measures to support the evaluation of the maintainability of the feature model, using 40 measures in total. It automatically exports measurement values to spreadsheets in Microsoft Excel format. The tool is illustrated in Fig. 6.

The DyMMer tool evaluates dynamic feature models using measures that are associated with contexts. The eight measures that are specific to dynamic feature models are presented in Table 1.

Study design

The following subsections describe the research questions investigated in this paper and the activities performed to meet the objectives.

Research questions

This work aims to study the variability in SOLAR SECO. For this, we proposed two research questions:

Research question 1 (RQ1) —How to model dynamic aspects in an educational software ecosystem?

Answering RQ1 is the first step in identifying which features can have their dynamic aspects modeled on SOLAR VLE across their various platforms (web and mobile).

Research question 2 (RQ2) —How to evaluate from measurements the modeling of dynamic aspects of an educational software ecosystem?

By answering RQ2, we have a quantitative view of the dynamic aspects, making it possible to compare results.

²<https://github.com/DyMMerProject/DyMMerV2>

Table 1 Measures calculation formulas

Acronym	Measure name	Measure description
NC	Number of contexts	Number of contexts of a feature model
NAF	Number of activated features	Number of activated features in each context
NDF	Number of deactivated features	Number of deactivated features in each context
NCC	Number of context constraints	Number of constraints in a specific context
AFCA	Number of activated features by context	Number of activated features in a specific context/NC
DFCA	Number of deactivated features by context	Number of deactivated features in a specific context /NC
CF	Number of context features	Number of features that are always present in the feature model, regardless of the context that is enabled
CFC	Number of context features in constraints	Number of features that are related with constraints in a specific context

Functionality modeling

To enable data collection, initially, an activity to select functionality for analysis is required. For this, it was identified in SOLAR VLE a functionality that was widely used by users and that had several possibilities of extension. In this case, we selected the functionality of the discussion forum, which is a common activity for VLEs. This feature is usually the most used in these environments and promotes interaction between students and teachers.

Once functionality is selected, the next step is the definition and modeling of contexts, which basically consists of elaborating situations that simulate real use cases. For this work, three contexts were modeled, related to situations of SOLAR VLE use. The idea is that the three elaborated contexts make clear the use of the discussion forum in different platforms and user environments, in this case the student, considering network characteristics, connection, and mobility.

Finally, the feature model and the contexts must be inserted into a tool for manipulation. For the feature model, we used the S.P.L.O.T. tool, and for the contexts, the DyMMer tool.

Data collection procedures

After the activity modeling and the insertion of models and contexts in the tools, the study environment is ready for data collection. In this case, the DyMMer tool was used, as several metrics already have their automated collection in the tool itself. Table 1 displays a set of eight context-related metrics available in the DyMMer tool. All these metrics were collected.

Data analysis procedures

For data analysis and interpretation, taking the contexts as reference and the collected metrics, relationships between the results were analyzed. All three contexts will have the metrics analyzed and impact globally. Additionally, the research questions were answered, as well as an overview of impacts on SOLAR SECO, as well as observations on threats to the validity of work.

Results

In this section, the findings of the study are discussed and the answers of the research questions are presented. This section reports the activities of context definition, the discussion forum feature model modeling in tools, metric collection, analysis, and interpretation of data. In addition, research questions are answered.

Research question 1 (RQ1)—How to model dynamic aspects in an educational software ecosystem?

Context identification

One activity of this work is the context elaboration to exemplify changes or adaptations that must occur in the user environment or in the application, as the situations occur in the environment. These contexts were idealized in order to enable a variability analysis in tools. Since the idea is to focus on the discussion forum functionality, the elaborated contexts will aim at accessing SOLAR VLE in general and functional aspects of the discussion forum.

For contexts, we will consider using SOLAR VLE in a web application running on a notebook and in a mobile device application. This will allow us to make some adjustments regarding device power, audio quality, and feature limitation. We nominated the three contexts with the following nomenclature: (context 1) home, (context 2) mobility with quality, and (context 3) mobility without quality.

For the first context, initially consider a distance learning student at your home, accessing SOLAR VLE for classes, and plugged into your notebook. In this context, the student has high battery and good connectivity capabilities, due to ongoing power and connection speed services at home. This situation has no impact on the quality of the audio and video service.

Then, the student has to go to a classroom, eventual situation that occurs in distance courses in the semi-classroom mode. It will continue to be connected to SOLAR VLE in a vehicle used for its travel (bus or car). In this situation, for this study, we will consider that the notebook battery is at a below average charge, but still far from

discharging. Similar to this situation is when the student is using the SOLAR VLE for mobile version. In this case, the internet connection occurs through the mobile phone network, which consumes more energy resources. In this context, sound quality is reduced, and video resources will no longer be available, only images.

Finally, the connection to the mobile phone loses quality, impacting the quality of the connection, and the application must be adjusted. Audio and video features are disabled to conserve power, as downloading and uploading multimedia elements consume power resources, and only the basic functionalities of the discussion forum is possible, with only text being used.

These three scenarios are only intended to illustrate the study of variability within the context of VLE and SOLAR SECO. They were designed by a person who knows the environment, who has already used it in classes and activities, and who knows a little about the technical and business characteristics of SOLAR VLE. Additionally, another person with experience in SPL reviewed the models in the tools.

Discussion forum feature models

The three contexts were developed by a specialist in the educational context and with experience in SOLAR VLE. In addition, this specialist also had knowledge in SECO concepts. The feature model and the tools used to support the modeling were presented to the specialist, who created the models.

The model was reviewed by an SPL and DSPL specialist. This expert reviewed the content of the feature model in the tools and the constraints for contextual adaptations.

Once the contexts are defined, it is necessary to model them in a computational tool, in the form of a feature model. To model the discussion forum feature model, we used the S.P.L.O.T. tool, available online for use. There were a number of interesting features to model. We added to the model characteristics of the environment and infrastructure, which were sound quality, file types (image, sound, and audio), and connection types. It is worth mentioning that some features of the model do not exist. They are in the model only to provide an extension view of SOLAR VLE and their influence on its use in different contexts. Figure 7a shows the feature model in the S.P.L.O.T tool.

However, the S.P.L.O.T. tool only allows the inclusion of the feature model without contextual information. For this, we used the DyMMer tool. The DyMMer tool allows you to create contexts about a feature model, where each has an indication of which features are enabled or disabled. Then, the file generated by S.P.L.O.T. was exported to DyMMer, and in it, all three contexts were inserted. Figure 7b shows the model exported to the DyMMer tool [25].

SOLAR SECO has a very diverse set of developers and users. The images in Fig. 8 allow to view only three contexts of an aspect of SOLAR SECO, related to one of the most used features of SOLAR VLE, which is the discussion forum.

Constraints are modeled as a formula in the conjunctive normal form (CNF). Thus, two types of constraints were modeled: feature A requires feature B ($\neg\text{Feature-A} \vee \text{Feature-B}$), and feature A excludes feature B ($\neg\text{Feature-A} \vee \neg\text{Feature-B}$). All constraints can be seen in Fig. 9. S.P.L.O.T. does not allow the inclusion of constraints for contexts, only general constraints. In this case, only constraints 1 and 2 were modeled in S.P.L.O.T. and exported later to the DyMMer tool. In the DyMMer tool, it is possible to include context constraints. Thus, constraint numbers 3, 4, 7, and 8 belong to context 1, constraint 6 belongs to context 2, and constraint 5 belongs to context 3.

We concluded with these activities that, initially, we have to identify, design, and model the educational contexts to be analyzed and then standardize their description. Context constraints must be added to define contextual adaptations. Tools to support the modeling, inclusion of constraints, and collection of metrics must be used to support the process and provide flexibility in maintaining the model. In our work, we used the feature diagram and the S.P.L.O.T and DyMMer tools. Once modeled, metrics could be collected to measure the variability of contexts.

Research question 2 (RQ2)—How to evaluate from measurements the modeling of dynamic aspects of an educational software ecosystem?

Intensive software systems are usually centered on a technological platform so that increasing attention has been paid to influencing and promoting interdependence in the relationships between all involved parties [27]. The Software Engineering community has used the ecosystem metaphor to describe systems that are generally centralized on a technological platform. In this sense, the two concepts are related and promote discussion and research.

As previously mentioned, SOLAR SECO has a very diverse set of developers and users, and in this work only a few features of the discussion forum will be presented.

For example, suppose a student is at home using the web version of SOLAR, in a Google Chrome browser (context 1). The SOLAR web application has several features, implemented with javascript and access to a PostgreSQL database. The various active functionalities are implemented with different frameworks, and the VLE monitors the network resources to configure the sound quality and multimedia resources. With the change to the context of access to mobile resources, the platform now used is Android, with specific features of the operating system. The application no longer uses a

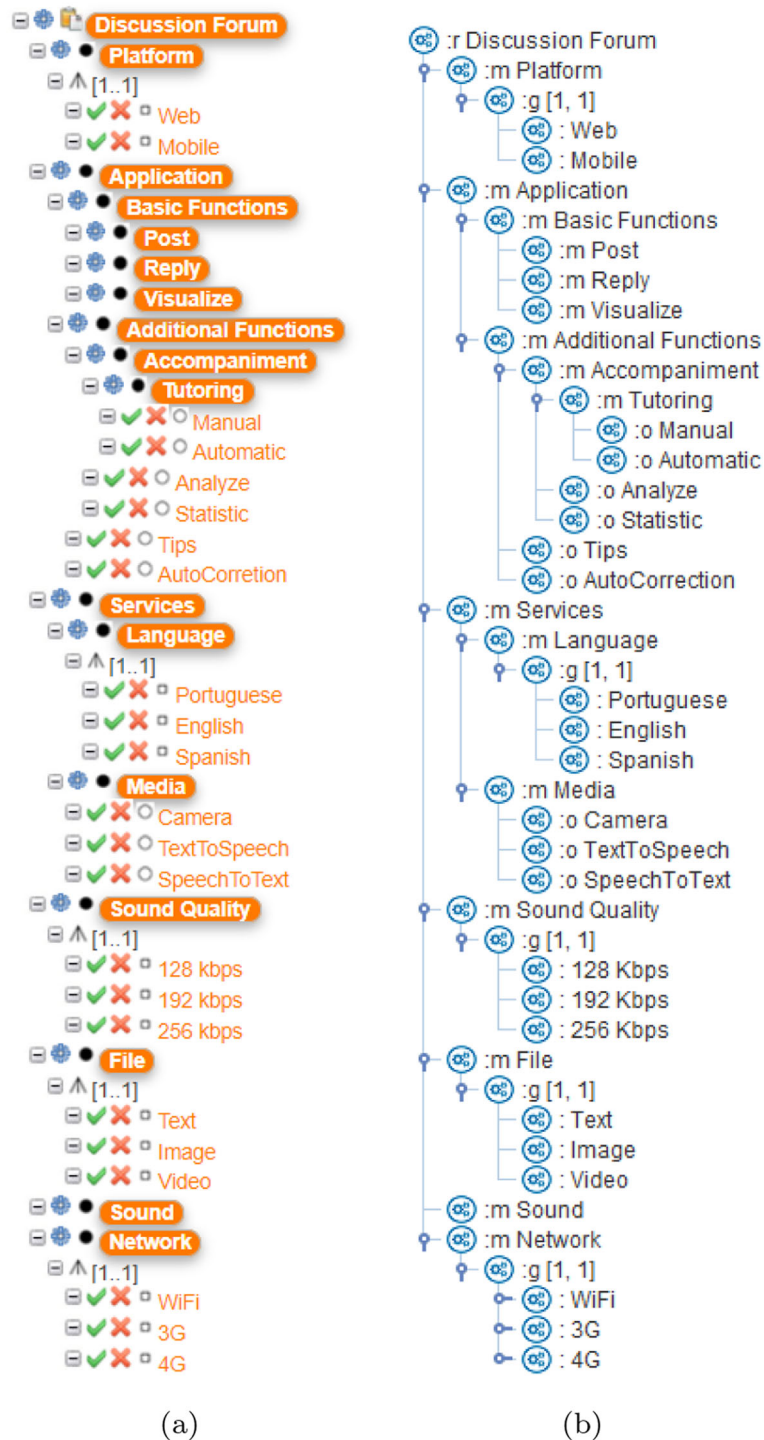
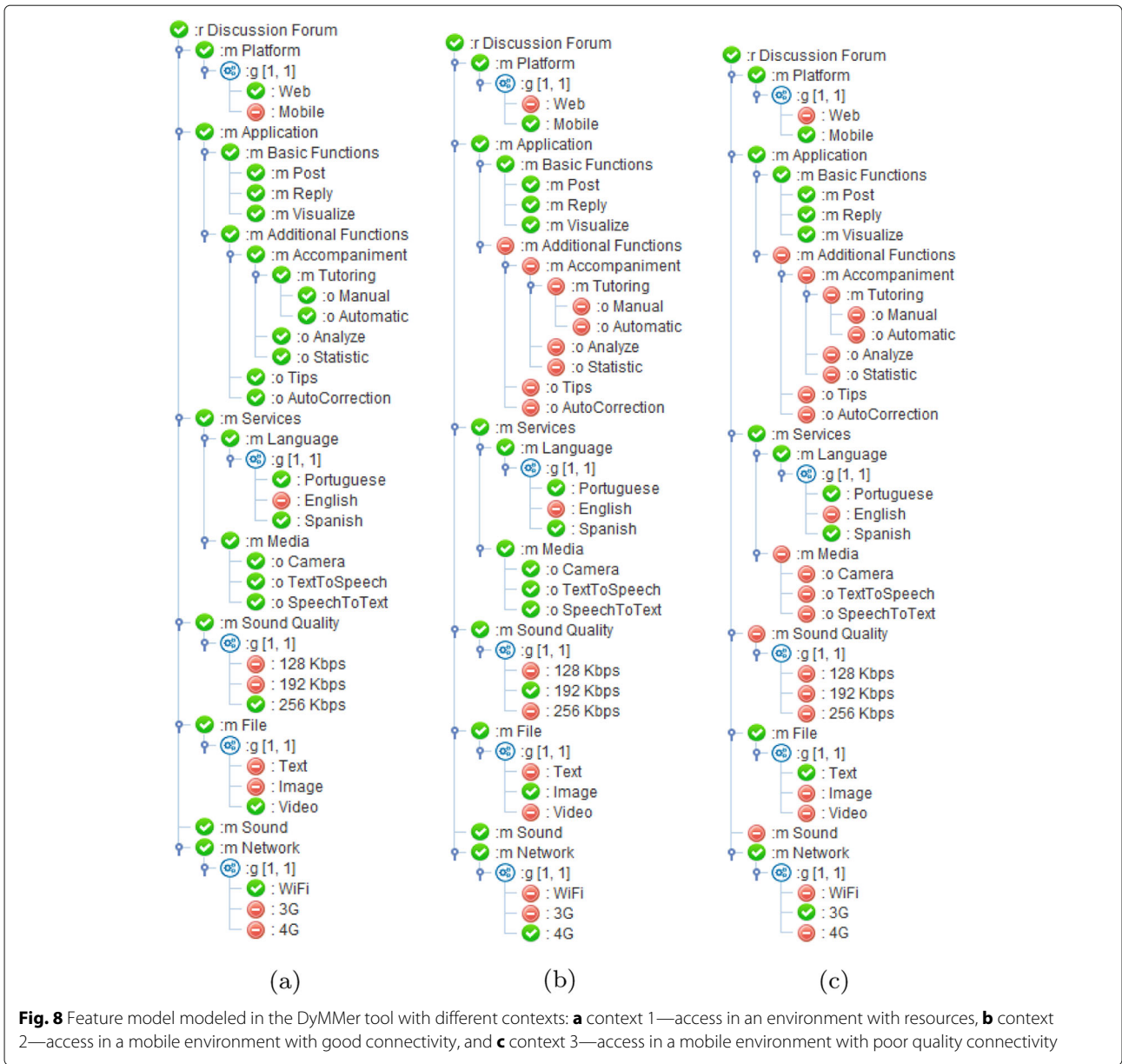


Fig. 7 a General feature model in S.P.L.O.T. [26]. b General feature model in DyMMer [25]

browser, but an application on the device. Interface adaptation features, depending on the device's screen size, are applied. Technologies for accessing the device's camera are accessed by the application. When a change to the context with few resources occurs, the minimum functionalities of the application must be active for a

minimum quality. Multimedia resources should no longer be available. All contexts must implement capabilities to identify network/internet resources, to better configure the application. In all of these transitions, the proper stages of development must be respected, such as requirements, analysis, design, implementation, and



tests, often in parallel. As there are several many possible technologies involved, a certain degree of complexity is involved in the integrations between the different platforms.

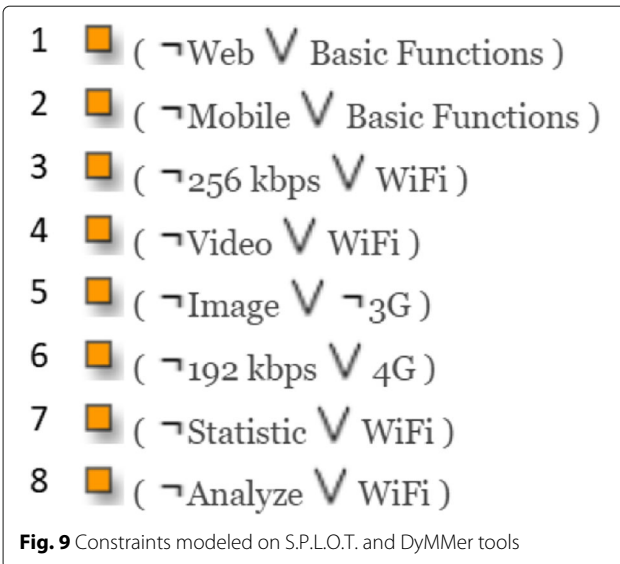
Some metrics were collected in the DyMMer tool [25], allowing some analysis. The number of contexts in the feature model is 3, just because we modeled 3 contexts for this analysis. Several other contexts could have been modeled.

The total number of features is 40. The number of activated features, respectively, for the three contexts, were 31, 22, and 16. It can be seen from the situation described as an example that, as the contexts occurred, the amount of activated features was decreasing. This makes sense, because due to this the situation, the resources really

were becoming scarce, limiting their use and adapting the application. Similarly, the number of disabled features has been increasing (9, 18, and 24).

Context features are the number of features that are always present in the feature model, regardless of the context that is enabled, and for all three scenarios were equal to 12. These are generally the features that must always exist for the base application to run fully.

AFCA is the number of features enabled in a specific context divided by the number of contexts. DFCA is the number of features disabled in a specific context divided by the number of contexts. AFCA resulted for the three scenarios in 10.3, 7.3, and 5.3. DFCA resulted in 3, 6, and 8, respectively. According, the features were deactivated



and AFCA decreased, similar to the number of enabled features. Similarly, DFCA and the number of disabled features increased. CF resulted in 12, which means there are 12 features always present in the feature model regardless of context.

Finally, some metrics are related to constraints. CFC resulted in 3 due to only 3 features in the 2 general

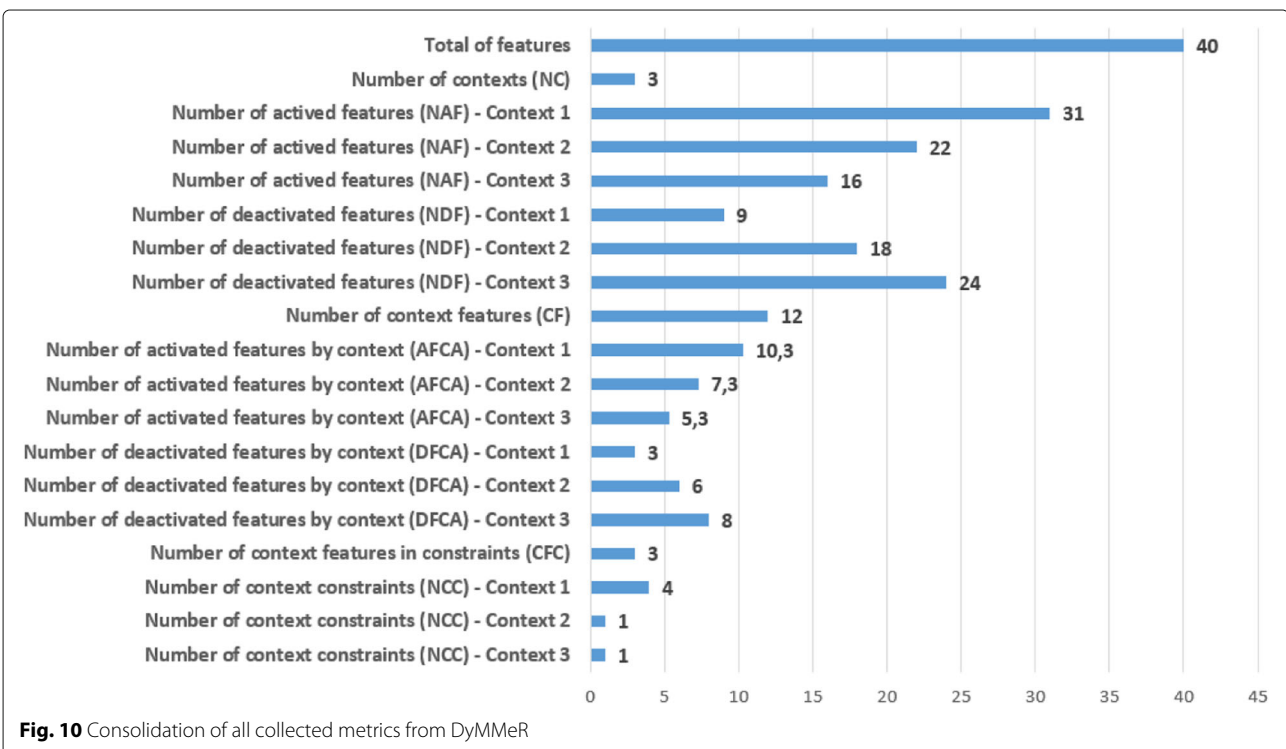
constraints. NCC resulted in 4 for context 1, and 1 for contexts 2 and 3.

Figure 10 presents an overview of all measurements collected by the DyMMer tool. As the measurements are arranged, the comparison is facilitative, making it possible to visualize some metrics that grow as others decrease, as features increase or decrease, such as NAF, NDF, AFCA, and DFCA.

The values of the presented metrics indicate the dynamics level of the feature model and the complexity of these models in relation to context changes. Note that for the evaluated model, the model has only 3 contexts and 12 context features. The model has a medium level of dynamics, as 12 out of 40 features are affected by the context change, representing a total of 33.33%. In addition, context constraints are also identified in the model, which increases the complexity of the model in relation to context changes. We concluded that the presented model has a medium level of difficulty in relation to dynamic modeling and a high level of complexity in relation to contextual changes.

Discussion

In this section, we discuss the research results in light of related work and present the implications for research and practice.



General analysis of the SOLAR educational ecosystem

As a consequence of the massive adoption of the Internet, different e-learning platforms have been incorporated into educational institutions, ranging from schools to universities [28]. As expected, a software market of auxiliary applications for these platforms has also emerged. For instance, Moodle platform has currently more than 700 complementary modules and plugins in its website. One of the challenges these auxiliary applications must face is how to deal with the variability inherent to these different, but similar platforms. All of them support the concepts of activity, assignment, deliverable, or grade. However, they present a wide range of differences among them. For example, their backend database schemas are different, but they refer to the same concepts. Therefore, the companies developing auxiliary products for e-learning systems must deal with this variability.

As advances in digital technologies increase, e-learning domain moves from being a purely “one-fits-all” online content delivery to an adaptive learning that fits a wide range of students [29]. This leads companies to uphold the productivity and the skills of their employees through more flexible and adaptive approaches.

In general, features that can be very useful for the community using SOLAR are related to tracking, especially if they are automated. Many of these demands originate from users, for necessity or suggestion. Thus, naturally, there is an improvement of features and consequently evolution. The designed feature diagram illustrates some features specifically for the discussion forum. For the other features of SOLAR, the same experiment could be repeated, generating many features when associated with different contexts. SOLAR WEB has specific features of this platform, just as SOLAR MOBILE has features of mobile applications. However, the same functionality may exist, with the exception of screen space, visualization, and use constraints.

Documentation and functionality modeling in a traditional manner is not easy to maintain in the SOLAR VLE development environment. And this is true for all related systems that evolve SOLAR VLE functionalities. Additionally, these systems have different clients and users, which also makes documentation difficult. Agile or DevOPS modeling strategies would be most welcome to accelerate the development process and put them into practice. The changes come from many different sources, either from users or from the development team itself. Management of these changes should be done carefully, as the impact on related systems can be serious. Diagram generation, such as UML, could contribute to both static and dynamic aspects, as well as sequence or activity diagrams, but with caveats. Some representation for dynamics and real-time adaptations would be very useful,

especially when related to varied contexts generated by environment characteristics.

The modeling of dynamic variability has proved to be a great challenge due to aspects such as the number of variants and their relationships, traceability of the model variants being aligned to the variants represented in the architecture, reconfiguration representation of the variants according to the context of the environment, representation restrictions between variants, and context restrictions. Thus, some studies have extended the feature model to meet these challenges of representing dynamic variability [30–32]. However, few tools are supported for representing these extensions, such as the Variamos [24] and the DyMMer [25] tools. In our study, we used the DyMMer tool because it is available and have the option of assessing the quality of dynamic aspects using specific metrics.

In relation to modeling dynamic aspects with UML, such as sequence diagrams and activities, the study of variability may not be as efficient. Since such diagrams tend to be very detailed, according to who modeled a certain functionality, they tend to be very technical, which can blur the study of system variability. And usually, when studying dynamic aspects, there is an impact between several features. In sequence and activity diagrams, usually only one case is modeled at a time, even with different alternative and exception flows.

In SOLAR SECO, UML models would only highlight features of SOLAR VLE or its associated systems. Thus, the study of variability would be impaired, because it needs in this context to be broader and global.

Some features of SOLAR, such as tracking and customizing features depending on the platform, are features that have attractive dynamics for SOLAR, especially in a variety of contexts. Such functions could be configured to suit the needs of each SOLAR user. Models that represent system dynamics in real time would also be very useful. Again, some representation for dynamics and real-time adaptations would be very useful, especially when related to varied contexts generated by environment characteristics. The feature diagram assists in modeling, behavior study, and changes. The DyMMer tool provided a good overview of the discussion forum functionality and some contexts. However, it is still a challenge how to model dynamic aspects of the system and implement them.

Assuming that SOLAR permeates web and mobile platforms, which have varied usage situations, and it has varying types and levels of users, several research opportunities may arise. Some of them are functionality identification strategies, variability modeling techniques, variability testing, change impact study, maintenance, and adaptations on SOLAR SECO. In addition, there are research situations regarding the use of SOLAR SECO itself by the various user profiles.

This study was the first step to identify which characteristics of SOLAR VLE can have their dynamic and aspects modeled. Several situations of variability in SOLAR VLE can arise, since in this work only the discussion forum was considered. This gives rise to a lot of work on SOLAR VLE. And in this context, the simulation of how the different functionalities could behave in the face of the most varied situations would be of great value to SOLAR VLE.

In SOLAR VLE, many situations could be combined for the study of variability, due to the large number of features and functionalities. When simulating situations of variability for testing, study or research, and later decision-making, an effort similar to that described in this work can be performed, with context identification, modeling in some feature model tool, and context and restriction modeling. With this exercise, situations of context variation can be analyzed and thus evaluate a cost-benefit relation.

Impacts on SOLAR SECO

Regarding the impacts of the study for SOLAR SECO, this study allowed an analysis from the technical and social point of view.

SOLAR VLE, not only in its web application, but also in the mobile and MOOC versions, generally has the same basic functionality, with the exception that adapted for their platforms. Checking the effects of the three analyzed contexts, even if only for web application in a notebook, the same situations and analysis could be performed for mobile and MOOC application contexts. Impacts due to different mobile devices would be quite visible as the SOLAR VLE running on the mobile device is widely used. Screen restrictions (adjusting for devices) and battery would be highlighted. The adaptation of the functionalities would also occur, because currently they are already restricted.

There are situations where software suppliers can modify their products, and this impacts the production environment. This situation can also occur in SOLAR SECO. The simulation of different contexts can collaborate to predict impact situations in SOLAR SECO if products used in certain situations have to be modified, either by replacing the component or by exchanging the complete product itself.

Focusing in the mobile application development in SOLAR SECO, it is possible some work refer to SECO and mobile applications. This could refer to a Mobile Software Ecosystems (MSECO) [33]. A MSECO can help the establishment and analysis for a base of users, developers, companies, and applications. The relationships can contribute to improve quality not only of mobile applications, but also of other related products in the MSECO, such as mobile devices, wearable devices, and the platform. These features could become new features in SOLAR SECO.

From a social point of view, the analysis of the three contexts could have a bad performance whether users did not have the devices and infrastructure necessary for the full use of SOLAR VLE. Many users may not be able to use SOLAR VLE at home or at work, and in these cases, using SOLAR VLE on mobile devices would be very important to complete the courses.

It is common in the context of SOLAR VLE users, typically students, who do not have the necessary infrastructure for the full use of the environment. In this situation, the study of variability for adapting to needs according to the available infrastructure of users is of great impact for society. Thus, students can have a minimum of quality to carry out the activities.

For SOLAR SECO, in terms of variability, the possibilities for research and work are also immense, due to the dynamic characteristic of SECO. This implies that customizations for clients, for users, and for environments and platforms can occur more frequently. And in this context, simulating the interactions between the different SECO elements would contribute a lot to SOLAR VLE.

Threats to validity

This research can be affected by different factors which, while extraneous to the concerns of our work, can invalidate its main findings [34].

Internal validity. Internal validity is concerned with the validity within the given environment and the reliability of the results. The modeling was performed by only one person and validated by just another. Modeling can result in different models depending on the experience and effort of each designer. This can impact internal validity. In addition, only the discussion forum functionality was analyzed.

External validity. External validity is the extent that the obtained results of a case study can be generalized to other relevant research scenarios. The threat to external validity refers to the possibility of different implemented versions of the discussion forum. And these different versions may not have the same analyzed features. That would not be generalizable for this feature. In addition, this work analyzed only one feature. Different contexts may arise as new features and users access the environment. To generalize to SOLAR educational SECO with all its features, more work needs to be done.

Conclusion validity. Conclusion validity is concerned with the relation between treatment and outcome. The manner in which the study was performed by analyzing the results of two tools (S.P.L.O.T. and DyMMer) may limit the conclusion because they had feature and context models modeled by only one domain expert. Even if the model has been reviewed

by an SPL expert, there are possibilities for subjective interpretations. In addition, no repetitions of the experiment were performed with varying models of the same functionality and generally obtaining the same values of the measurements.

Construct validity. Construct validity is concerned with the relationship between theory and observation. The manner in which the study was designed and modeled using the S.P.L.O.T. and DyMMer tools can influence the work. Not all aspects of dynamic variability were represented in the model. It was a representation based on the experience of the specialist in the educational field. Since feature and context models have been modeled by a domain expert, even with the SPL expert review, it is still possible that context failures, unclear elements in the models, and lack of functionality may have occurred. In addition, the lack of more modeling tools can be another threat.

Related work

Software Product Lines in e-learning environments

Some works in the literature have developed research in e-learning and VLE in conjunction with SPL.

Sanchez et al. [28] discussed in their work about the use of SPL engineering techniques, whose goal is the effective production of similar software systems. They argued this use can benefit in this particular knowledge area. The authors shown how to build a SPL for a family of applications, aiming to extract knowledge from log data stored in e-learning platforms. They also illustrated with an example how SPL engineering can help to the development of software auxiliary applications for e-learning platforms, using as case study a data mining application for analyzing log data from e-learning platforms, named E-learning Web Miner. They concluded the main advantage of adopting a SPL approach is the process of adapting the product, in order to fit it with a specific customer requirements, is carried out practically with no cost, since this process is performed automatically.

One factor for restricted technology usage in education today is mainly because of the huge effort and the ever increasing complexity of educational technologies [35]. Chimalakonda and Nori [35] presented a research on mining a SPL from nine e-learning systems, with different development teams and varied development processes. The goal of this family of e-learning systems is to address adult illiterates in India spread across many Indian languages. The authors shown a challenging situation, while the SPL arises from a societal context rather than a business context as in traditional SPL. The aim of this work is to present the application of SPL to the legacy e-learning systems. They explained the context of this domain and presented some key challenges of mining an SPL from e-learning systems.

According advances in digital technologies increase, the e-learning domain moves from being a purely “one-fits-all” online content delivery to an adaptive learning. This fits a wide range of students and leads companies to uphold the productivity and the skills of their employees through more flexible and adaptive approaches [29]. The adaptation takes place at different areas of adaptive e-learning systems. It includes the presentation by adapting the content or the used media, and navigation by adapting different ways of navigation for a flexible access to information. Dounas et al. [29] proposed a continuous requirement monitoring that uses a constraint program to check the conformity of adaptive e-learning systems to their requirements, reacting when deviations occur at runtime. They specified the system’s requirements with a DSPL. Their approach was guided by the graphical tool VariaMos, which allows the designer a performance modelling and verification of the specification through simulations.

With the growth in demand for software and its faster delivery, SPL could be a good solution for application development. In this context, a SPL approach is one of the best methods that can be used to develop an educational software family [36]. Zaki et al. [36] researched on the development of an online learning SPL by studying issues in online learning SPL. They identified an appropriate pedagogical approach to online learning environments and designed a constructivist learning environment based on the theory of constructivism and design principles. In their work, the authors compared 10 VLE. They were tabulated against a constructivist learning environment. The comparative evaluation is to discover commonalities and variabilities between online learning applications. The results could be useful as the process of domain analysis, which is to document the commonalities and variabilities between SPL members. Hence, it can be used to enhance the feature model for online learning applications.

All of these works used SPL to deal with e-learning. Some used modeling tools, such as VariaMos in [29]. Others worked on VLE extensions, such as Moodle in [28]. With the exception of [29], which used DPSL, the other related work used only SPL. In our work, we used the VLE SOLAR, DSPL, an SLP repository (S.P.L.O.T.), and a DYMMER tool for the modeling of the DSPL.

Variability in software ecosystems

There are some studies in the literature related to variability in software ecosystems.

Software ecosystems rich in variability need configuration features, as in any product line [37]. However, several additional resources are needed, taking into account the characteristics of the software ecosystem. Eichelberge et al. [37] developed the Integrated Variability Modeling Language (IVML) to describe

configurations of software ecosystems rich in variability. IVML is a variability modeling and configuration language. IVML has been applied in various industrial contexts, both as part of research projects and in industrial cooperation.

Nowadays, software industry established successful ecosystems around technological platform [38]. However, architectural knowledge of the existing ecosystems is implicit and fragmented among online documentation. Platform providers can learn from the existing ecosystems in order to make reasonable design decisions with respect to their business strategies to create their own ecosystems. Jazayeri et al. [38] identified a variability model for architectural design decisions of a store-oriented software ecosystem product line from an enterprise perspective, comprising business, application, and infrastructure views. They derived the variability model from fragmentary material of existing ecosystems, and they described real-world ecosystems from diverse domains using the variability model. This knowledge helps platform providers to develop customized ecosystems and contributes to an increase in designer and developer productivity.

Companies nowadays need to supply a market and handle at the same time with customers individual solutions [39]. Individual products for customers are derived and adapted by adding new features or creating new versions of existing features to meet the customer-specific requirements. Development teams typically use version control systems to track changes to product lines and products. However, it is difficult to relate such low-level changes to features and their evolution in the SECO. Hinterreiter [39] presented which workflows and additions to variation control systems are required to support feature-oriented development in an industrial SECO environment. He also investigated mechanisms that support feature-based monitoring to guide the evolution in SECOs.

Many companies in industrial automation domain nowadays need to serve a mass market while at the same time customers demand individual customer-specific solutions [40]. These customizations often apply to individual products. However, they may also be needed at the level of product lines for whole market segments. To handle this problem, development is frequently organized in SECOs. Hinterreiter et al. [40] proposed an approach for supporting feature-oriented, distributed development and evolution in industrial SECOs. Their approach allowed to share new or updated features by transferring them to other product lines in the ecosystem. This is useful when a feature developed in an individual customer project becomes relevant for another market segment or when updates of features need to be transferred to related products in the ecosystem.

In the literature, there is still a gap in studies of dynamic variability in software ecosystems, despite the challenges related to this topic. Thus, our study is an initial step in modeling dynamic aspects in software ecosystems.

Conclusions

If we consider a software intensive system as those in which software is a dominant and essential element, both in its structure and as a cross-sectional element of the production stages, having a substantial impact on the planning, development, and evolution of these systems, we have SOLAR VLE included into this category. Its development in its various modalities (web, mobile, and MOOC) permeates various levels of development, demanding institutions, personal and institutional users, partnerships, and business opportunities. Its evolution follows the needs of various online courses and alignment with supporting technologies such as accessibility services and virtual labs.

This work is important for the following reasons: (i) the educational domain is a very promising area that generates many applications being used by different types of users and varied platforms; (ii) for SOLAR VLE, maintenance is constant, as well as evolutions, so a study of dynamic variability would collaborate even more for its development; (iii) for SECO, it is a case study in a specific domain, SOLAR SECO; and (iv) there are few studies of dynamic variability in the educational area, according to the literature review.

This work, as well as similar work, can be used to evaluate the variability of situations that can be complex to study in the real world. A feasibility study can be carried out through similar ideas, as it is possible to evaluate different contexts and their alterations. Finally, similar work may be used to assist in decision-making and system improvement.

This paper discussed aspects of dynamic variability with an example of the discussion forum functionality, modeled by means of a feature diagram. It has been realized that the complexity of modeling dynamic aspects and variability is a complex task and requires a lot of effort. As a future work, we intend to study deeply the modeling of other dynamic aspects of SOLAR SECO, with different models.

Acknowledgements

This is an extended version of a paper published and awarded in the I Workshop on Modeling and Simulation of Software-Intensive Systems (MSSIS 2019). We thank Dr. Valdemar Vicente Graciano Neto and Dr. Elisa Yumi Nakagawa for being the guest editors that conducted the review process for this paper. The authors also thank the students who developed the DyMMer tool, which it was the course conclusion work of some students, under the guidance of Profa. Carla Bezerra.

Authors' contributions

Both authors participated in the writing of the work. The use of modeling tools was also performed by both, as well as literature review. Carla Bezerra is expert

in Software Product Lines. Emanuel Coutinho worked on the development and use of the SOLAR virtual learning environment, and he is expert in Software Ecosystems. The authors read and approved the final manuscript.

Funding

The authors declare that there is no funding.

Availability of data and materials

The original feature model used in this work is available on the S.P.L.O.T. tool website, which can be downloaded (Feature Template—SOLAR VLE Discussion Forum—<http://www.splot-research.org/>). The model adjusted in the DyMMer tool can be obtained by contacting the authors.

Competing interests

The authors declare that they have no competing interests.

Received: 10 December 2019 Accepted: 23 August 2020

Published online: 07 September 2020

References

1. Iso/iec/ieee systems and software engineering – architecture description (2011) ISO/IEC/IEEE 42010:2011(E)(Revision of ISO/IEC 42010:2007 and IEEE Std 1471-2000). pp 1–46. <https://doi.org/10.1109/IEEESTD.2011.6129467>
2. Oquendo F (2016) Formally describing the architectural behavior of software-intensive systems-of-systems with SosADL. In: 2016 21st International Conference on Engineering of Complex Computer Systems (ICECCS). IEEE, New York. pp 13–22. <https://doi.org/10.1109/ICECCS.2016.012>
3. Messerschmitt D, Szyperski C (2003) Software ecosystem: understanding an indispensable technology and industry. 1st edn. The MIT Press, The MIT Press
4. Jansen S, Brinkkemper S, Finkelstein A (2009) Business network management as a survival strategy: a tale of two software ecosystems. In: 11th International Conference on Software Reuse. CEUR Workshop Proceedings, vol. 505. CEUR-WS, Aachen. pp 34–48. <http://ceur-ws.org/Vol-505/iwseco09-5JansenBrinkkemperFinkelstein.pdf>
5. Manikas K, Hansen KM (2013) Software ecosystems - a systematic literature review. *J Syst Softw* 86(5):1294–1306
6. Boucharas V, Jansen S, Brinkkemper S (2009) Formalizing software ecosystem modeling. In: Proceedings of the 1st International Workshop on Open Component Ecosystems, IWOCE '09. ACM, New York. pp 41–50
7. Axelsson J, Skoglund M (2016) Quality assurance in software ecosystems: a systematic literature mapping and research agenda. *J Syst Softw* 114:69–81. <https://doi.org/10.1016/j.jss.2015.12.020>
8. Manikas K (2016) Revisiting software ecosystems research: a longitudinal literature study. *J Syst Softw* 117:84–103. <https://doi.org/10.1016/j.jss.2016.02.003>
9. Castro Filho JA, Loureiro RC, Paula PS, Sarmiento WWF, Peixoto LE, Pequeno HSL, Rocha BTS, Viana Jnior GS (2005) Portal humanas: Um ambiente colaborativo para criação de projetos e comunidades virtuais para a área de humanidades. In: XVI Simpósio Brasileiro de Informática na Educação (SBIE 2005). SBC, Porto Alegre
10. Coutinho E, Moreira L, Sarmiento W (2013) Maat - sistema de avaliação de alunos e tutores para um ambiente virtual de aprendizagem. In: IX Simpósio Brasileiro de Sistemas de Informação (SBSI2013). SBC, Porto Alegre
11. Gütl C, Chang V (2008) The use of web 2.0 technologies and services to support e-learning ecosystem to develop more effective learning environments. In: Data Engineering and Management, ICDM 2008 Proceedings. Bishop Heber College, Tiruchirappalli. pp 145–148
12. Coutinho EF, Santos I, Bezerra CIM (2017) A software ecosystem for a virtual learning environment: Solar seco. In: 2017 IEEE/ACM Joint 5th International Workshop on Software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems (JSOS). IEEE, New York. pp 41–47. <https://doi.org/10.1109/JSOS.2017.2>
13. Bosch J (2009) From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference, SPLC 09. Carnegie Mellon University, USA. pp 111–119
14. Pascual GG, Lopez-Herrejon RE, Pinto M, Fuentes L, Egyed A (2015) Applying multiobjective evolutionary algorithms to dynamic software product lines for reconfiguring mobile applications. *J Syst Softw* 103:392–411
15. Benavides D, Segura S, Ruiz-Corts A (2010) Automated analysis of feature models 20 years later: a literature review. *Inf Syst* 35(6):615–636
16. Capilla R, Ortiz, Hinchey M (2014) Context variability for context-aware systems. *Computer* 47(2):85–87
17. Coutinho EF, Bezerra CIM (2019) Um estudo sobre a variabilidade de aspectos dinâmicos no ecossistema de software educacional solar. In: I Workshop em Modelagem e Simulação de Sistemas Intensivos em Software (MSSiS 2019). SBC, Porto Alegre. pp 59–68. <https://doi.org/10.5753/mssis.2019.7560> <https://sol.sbc.org.br/index.php/mssis/article/view/7560>
18. Jansen S, Brinkkemper S, Finkelstein A (2007) Providing transparency in the business of software: a modeling technique for software supply networks(Camarinha-Matos LM, Afsarmanesh H, Novais P, Analide C, eds.). Springer, Boston, MA
19. Costa G, Silva F, Santos R, Werner C, Oliveira T (2013) From applications to a software ecosystem platform: an exploratory study. In: Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems, MEDES '13. ACM, New York. pp 9–16
20. Bosch RCJ, Hinchey M (2018) Cutting-edge topics on dynamic software variability(Hinchey MG, ed.). IEEE, Wiley-IEEE Press. <https://doi.org/10.1002/9781119174240.ch14> <https://ieeexplore.ieee.org/document/8471057>, Chap. 14
21. van Gorp J, Bosch J, Svahnberg M (2001) On the notion of variability in software product lines. In: Proceedings Working IEEE/IFIP Conference on Software Architecture. IEEE, New York. pp 45–54. <https://doi.org/10.1109/WICSA.2001.948406>
22. Capilla R, Bosch J, Trinidad P, Ruiz-Cortés A, Hinchey M (2014) An overview of dynamic software product line architectures and techniques: observations from research and industry. *J Syst Softw* 91:3–23
23. Alférez GH, Pelechano V, Mazo R, Salinesi C, Diaz D (2014) Dynamic adaptation of service compositions with variability models. *J Syst Softw* 91:24–47
24. Mazo R, Muñoz-Fernández JC, Rincón L, Salinesi C, Tamura G (2015) VariaMos: an extensible tool for engineering (dynamic) product lines. In: Proceedings of the 19th International Conference on Software Product Line. ACM, New York. pp 374–379. <https://doi.org/10.1145/2791060.2791103>
25. Bezerra CI, Barbosa J, Freires JH, Andrade RMC, Monteiro JM (2016) Dymmer: a measurement-based tool to support quality evaluation of DSPLE feature models. In: Proceedings of the 20th International Systems and Software Product Line Conference. ACM, New York. pp 314–317. <https://doi.org/10.1145/2934466.2962730>
26. Mendonca M, Branco M, Cowan D (2009) S.p.l.o.t.: software product lines online tools. In: Proceedings of the 24th ACM SIGPLAN Conference Companion on Object Oriented Programming Systems Languages and Applications, OOPSLA '09. ACM, New York. pp 761–762
27. Coutinho EF, Viana D, Santos RP (2017) An exploratory study on the need for modeling software ecosystems: the case of SOLAR SECO. In: 2017 IEEE/ACM 9th International Workshop on Modelling in Software Engineering (MiSE). IEEE, New York. pp 47–53. <https://doi.org/10.1109/MiSE.2017.3>
28. Sanchez P, Garca-Saiz D, Zorrilla M (2012) Software product line engineering for e-learning applications: a case study. In: 2012 International Symposium on Computers in Education (SIEE). IEEE, New York. pp 1–6
29. Dounas L, Mazo R, Salinesi C, El Beqqali O (2015) Continuous monitoring of adaptive e-learning systems requirements. In: 2015 IEEE/ACS 12th International Conference of Computer Systems and Applications (AICCSA). IEEE, New York. pp 1–8. <https://doi.org/10.1109/AICCSA.2015.7507210>
30. Jean-Baptiste L, Maria-Teresa S, Jean-Marie G, Antoine B (2013) Modeling dynamic adaptations using augmented feature models. In: Proceedings of the 28th Annual ACM Symposium on Applied Computing. ACM, New York. pp 1734–1741. <https://doi.org/10.1145/3107411.3107444> <https://doi.org/10.1145/2480362.2480690>
31. Baresi L, Quinton C (2015) Dynamically evolving the structural variability of dynamic software product lines. In: 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. IEEE, New York. pp 57–63. <https://doi.org/10.1109/SEAMS.2015.24>
32. Moritani BI, Lee J (2017) An approach for managing a distributed feature model to evolve self-adaptive dynamic software product lines. In: Proceedings of the 21st International Systems and Software Product Line

- Conference-Volume B. ACM, New York. pp 107–110. <https://doi.org/10.1145/3109729.3109743>
33. d L Fonto A, d Santos RP, Dias-Neto AC (2015) Mobile software ecosystem (MSECO): a systematic mapping study. In: 2015 IEEE 39th Annual Computer Software and Applications Conference vol. 2. IEEE, New York. pp 653–658. <https://doi.org/10.1109/COMPSAC.2015.121>
 34. Wohlin C, Runeson P, Höst M, Ohlsson MC, Regnell B, Wesslén A (2012) Experimentation in software engineering. Springer, Springer Heidelberg New York Dordrecht London
 35. Chimalakonda S, Nori KV (2013) What makes it hard to apply software product lines to educational technologies? In: 2013 4th International Workshop on Product Line Approaches in Software Engineering (PLEASE). IEEE, New York. pp 17–20. <https://doi.org/10.1109/PLEASE.2013.6608657>
 36. Zaki SM, Mohamad R, Halim SA, Ahmad NB (2017) Learning environment for software product lines online learning applications. In: 2017 8th International Conference on Information Technology (ICIT). IEEE, New York. pp 717–723. <https://doi.org/10.1109/ICITECH.2017.8079933>
 37. Eichelberger H, Schmid K (2015) IVML: a DSL for configuration in variability-rich software ecosystems. In: Proceedings of the 19th International Conference on Software Product Line, SPLC 15. Association for Computing Machinery, New York. pp 365–369. <https://doi.org/10.1145/2791060.2791116>
 38. Jazayeri B, Zimmermann O, Engels G, Kundisch D (2017) A variability model for store-oriented software ecosystems: an enterprise perspective. In: International Conference on Service-Oriented Computing. Springer, Cham. pp 573–588. <https://doi.org/10.1007/978-3-319-69035-342>
 39. Hinterreiter D (2018) Supporting feature-oriented development and evolution in industrial software ecosystems. In: Proceedings of the 22nd International Systems and Software Product Line Conference-Volume 2. ACM, New York. pp 79–86. <https://doi.org/10.1145/3236405.3236425>
 40. Hinterreiter D, Prähofer H, Linsbauer L, Grünbacher P, Reisinger F, Egyed A (2018) Feature-oriented evolution of automation software systems in industrial software ecosystems. In: 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA), vol 1. IEEE, New York. pp 107–114. <https://doi.org/10.1109/ETFA.2018.8502557>

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
