# Reusing a Geographic Software Product Line Platform: A Case Study in the Paleontological Sub-Domain

Fiorella Pesce[1], Sofia Caballero[1], Agustina Buccella[1,2]⋆, and Alejandra
Cechich[1]

[1] GIISCO Research Group
Departamento de Ingeniería de Sistemas - Facultad de Informática
Universidad Nacional del Comahue
Neuquen, Argentina
{fiorella.pesce,sofia.caballero,agustina.buccella,alejandra.cechich}@fi.
uncoma.edu.ar
[2] CONICET

**Abstract.** Developing Software Product Lines (SPLs) is a paradigm oriented to reusing software within particular domains. Key aspects within this paradigm are the inherent particularities of these domains and the techniques applied to systematize the ways to maximize reuse. In this article, we describe a process for creating SPLs by reusing through domain hierarchies, starting from the geographical domain and going deeper into the paleontological sub-domain. In particular, our process is based on standardizations and previous techniques already applied to another geographical sub-domain, which is marine ecology. Here we show how these techniques are applied in the paleontological sub-domain, improving the systematic reuse of software artifacts.

**Keywords:** Software Product Lines, Domain Reuse, Geographic Domain, Paleontology

## 1 Introduction

Software Product Line Engineering (SPLE) [6, 13, 18] proposes a software development process whose main objective is to maximize the reuse of artifacts to obtain faster-developed and high-quality software applications. Considering that the SPL development focuses on domain-oriented reuse, the success of the development depends on the identification, use and administration of the artifacts inside those domains; and therefore, the application of specific techniques for systematizing reuse becomes crucial.

The geographical domain [11] is broad in the sense it includes general aspects applicable to all products within this domain. Here, the creation of an SPL can be impracticable (the amount of variability that would have to be defined within

---

⋆ Corresponding author

each service will become unmanageable). So, it is logical to think about the division into different sub-domains, where each one has certain characteristics of the geographical domain, such as zooming in a map, obtaining coordinates, etc; and certain special characteristics such as knowing the depth of the ocean in certain areas. In this way, we build components designed to be part of a hierarchy of domains; that is, a set of general services is defined at the level of the geographical generic domain, and it is applicable to any SPL implemented within the included sub-domains.

In the literature, SPLE is a very active research area where there are many proposals for techniques or methods that improve the activities involved in SPL development. This can be seen in the number of literary or systematic reviews that exist today focused on techniques for modeling, implementation, validation, and so on [4, 5, 12, 16]. Similar approaches to our proposal addressing a hierarchy of domains can be those related to the development of software ecosystems or multiple software product lines [7, 15]. In general, these works focus on the problem of having developed components for different domains that must then work together to be integrated within the same development. In our case, these problems are minimized since the components are designed to be part of a hierarchy within the geographical domain sharing already defined standardizations.

A first approximation to our approach has been presented in [9, 10] where we have built an SPL within the marine ecology sub-domain, but taking common services from higher-level domains such as the oceanographic and the geographical. In these works, we have defined a methodology for the creation of SPLs that designs and implements a series of *software artifacts* necessary to communicate and model the domain based on standardizations and proper techniques. In this article we show the development process for the creation of SPLs, based on the previously presented methodology [9, 10], but focusing on reusing artifacts now created and adapted to the paleontological sub-domain. In particular, we focus on three of the software artifacts created as part of the domain engineering.

This article is organized as follows. In the next section we briefly describe our SPL development process oriented to the geographic domain and its sub-domains. In section 3 we show a case study where we apply this process to the paleontological sub-domain. Here, we model the information relevant to excavations to find paleontological pieces. Then, we performed a preliminary validation where we analyzed improvements due to reuse. Finally, we address conclusions and future work.

## 2 Development Process for Building Domain-Oriented SPLs

Figure 1 shows the four steps of our process for the domain engineering during SPL development. Recall that this engineering is responsible for identifying, capturing and organizing all the source information collected. As a result, it generates a software platform with a set of reusable and configurable artifacts that provide a common environment [18].
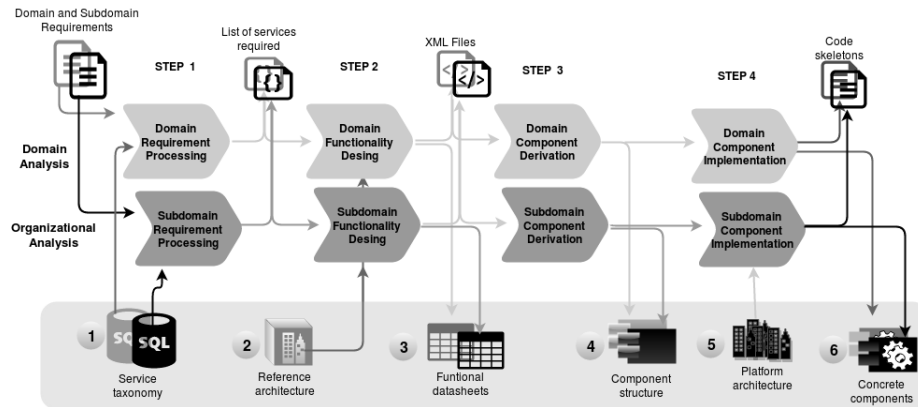
**Fig. 1.** Development process as part of the domain engineering, along with created and used software artifacts

In our development methodology, domain engineering is divided into two types of analyses: *domain* and *organizational*. The former involves the analysis and design of the information within a specific domain but focusing on a general view. Then, the organizational analysis uses the previous information to adapt it to the context of the SPL under development. In our application case, during the domain analysis the general geographical domain must be analyzed, obtaining common services that could be relevant to a set of sub-domains. Then, during the organizational analysis, the activities are focused on the specific sub-domain, in our case, on the paleontological sub-domain.

In Figure 1 we can see the six software artifacts that must be developed or used in the four steps of the domain engineering. In general, each artifact is built during the domain analysis (artifacts in gray color) and refined and completed in the organizational analysis (artifacts in black). The first artifact, which is one of the inputs of step 1, is the *service taxonomy* built as a hierarchical structure. It shows the categories that allow classifying the different services. In previous works, we have built this taxonomy for the geographical domain considering the Architecture Services standard (defined in the OpenGIS Service Architecture) [3], and the ISO/DIS 19119 std. [4] as a basis for defining services and their categories [10].

The second artifact is a *reference architecture* used as input of step 2 – *functionality design*. This architecture must specify a preliminary structure for the interaction of the services defined in the taxonomy. As we can see, there is only one instance of it in gray because it is reused from the one specified in the ISO 19119 std. In addition, in this step we must create the *functional datasheets* that specify each of the functionalities of the geographical domain (in gray) and the particular sub-domain (in black). They are designed through the interaction of

---

[3] The OpenGIS Abstract Specification: Service Architecture, 2002.

[4] Geographic information. Services International Standard 19119, ISO/IEC, 2005.

the necessary taxonomy services. Also, each functional datasheet is represented by a set of XML files that allow to automatically analyzing these models in search of inconsistencies or incompatibilities when specifying the variability [8, 20]. Then, in step 3, the *domain component derivation* should be performed to create reusable components based on the information defined in the functional datasheets [1, 19]. As part of this step and the following one (*component implementation*) we have defined, in previous works, initial mechanisms that assist in the creation of the fourth artifact (*component structure*) [2].

Finally, the refined artifacts determine the structure of the software that implement the fifth artifact – the *platform architecture*. It models the way in which each refined functional datasheet (based on the taxonomy refinement) is implemented as software components. This platform is then used in the configuration of the products to create the architecture of a specific application.

## 3   A Case Study in the Paleontological Sub-Domain

First of all, it is necessary to introduce some particular aspects of this domain. In Figure 2, we show part of the conceptual model where we focus on the classes that help us to understand the defined functionalities. To build the model, in addition to the elicited information of the expert users of the paleontological sub-domain, we have extracted information from the ISO standards 19109 and 19107 for geographical data, the ISO standards 21127:2014[5], LIDO[6], and CIDOC[7]. These three latest standards are specific to the paleontological sub-domain since they define and classify services and protocols required for the management of collections inside museums and the representation of paleontological pieces.

Figure 2 shows in gay color those classes that have been defined by the expert users and in black color those extracted from the three latest standards. According to the ISO 21127 std., any action that is carried out on a physical object is represented as an activity. In particular, the activity of *prospecting* is the one that reveals areas in which the *extracting* can be carried out by a group of professionals. In the case of finding some *specimen*, the activity of setting up a *jacket* (through the activity of *jacketing*) initiates. A jacket can contain *biological objects – specimen* or *piece*. These are subjected to mechanical and/or chemical procedures to correctly remove the object found. In addition, the geographical information, such as areas and points, has been represented using the MADS (Modeling of Application Data with Spatio-temporal features) [17] approach.

Then, based on this conceptual model and the functional requirements of the sub-domain, we built the *paleontological service taxonomy*. It was defined following the basis described in [10] through the realization of an iterative process that involved all the stakeholders – expert users, software engineers and developers,

---

[5] Information and documentation – A reference ontology for the interchange of cultural heritage information

[6] LIDO – Lightweight Information Describing Objects Version 1.0 – `http://network.icom.museum/cidoc/working-groups/lido/what-is-lido/`

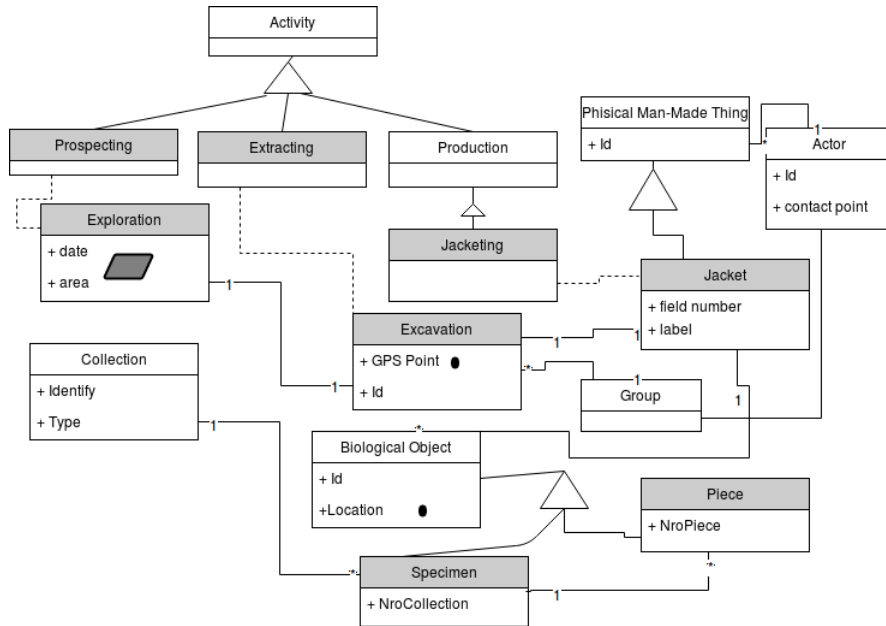[7] CIDOC Conceptual Reference Model Version 6.0 – `http://www.ccc-crm.org`

**Fig. 2.** Part of the Data Model for the Paleontological Sub-Domain

and used the geographical taxonomy defined by the ISO 19119 std. Also here, we used the specific rules of the sub-domain (already mentioned previously) and the reference architecture of ISO 19119 already applied for the SPL in the marine ecology sub-domain [9].

The reference architecture is a three-layer architecture in which the services of the taxonomy [10] are defined and classified. The first is the *human interaction* layer grouping together the services used to manage the user interface, graphics, and visualization aspects. The second is the *processing* layer, responsible for coordinating the functionalities required by the sub-domain. At the same time, it coordinates the services of the upper layer of human interaction and the lower layer, called *model administration*, which is responsible for handling data and storage.

The taxonomy, which is defined based on the layers of the architecture, has a set of main categories in which the corresponding services should be included. In Figure 3 we can see some services of the category *human interaction* (HI) already defined previously [10]. These services belong to the geographical domain (in gray and italic), and to the marine ecology sub-domain (in black and italic). Also, the services of the paleontological sub-domain (in black and bold) are defined. We can observe some of them that deal with different ways of information visualization. In this case, we see the attributes of explorations and excavations that can be displayed as tables or by means of labels directly on a map.
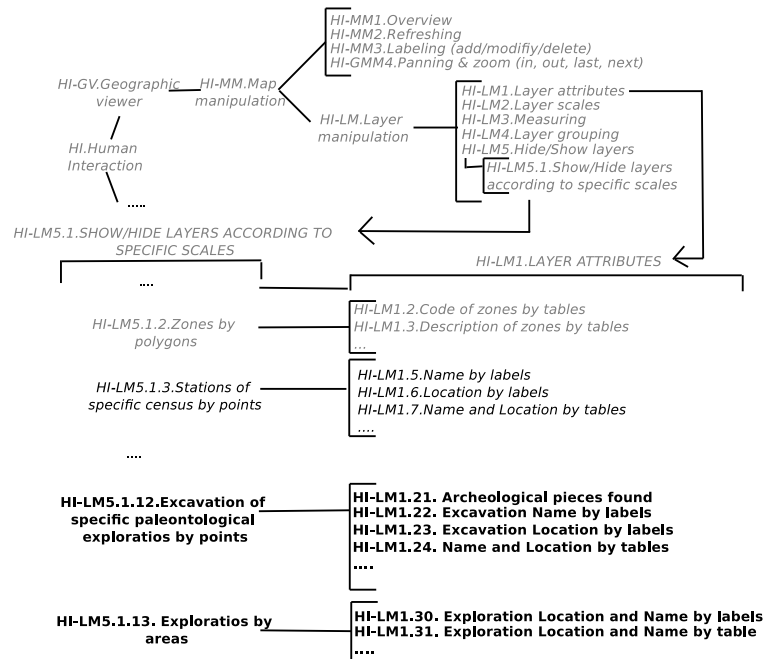
**Fig. 3.** Some services of the HI category defined by the domain hierarchy

In this way, as a result for each category, the complete taxonomy for the paleontological sub-domain is obtained with reused and inherited services from the geographical domain.

The next step is the creation of the functional datasheets (third artifact) with the functionalities of the sub-domain based on the services of the paleontological taxonomy and the reference architecture. The elaboration of these datasheets follows the same guidelines defined in previous works [1, 14] and uses our supporting tool called *Datasheet Modeler* [14]. It allows developers to create the datasheets by using variability [1] and taxonomy services, and translating to XML files. Figure 4 shows the functionality *Load excavations*, which allows registering a new excavation of an exploration already existing in the system. As we can see, an interaction is carried out between the services of the taxonomy already defined in order to carry out the functionality. The variability represented in this case defines that the data load must contemplate that it can be done in places without internet connections or GPS devices. Therefore, excavation data (such as geographical coordinates and found objects) must be allowed by manually entering information. At the same time, if it is required by any product, the options of loading data that come from spreadsheets files and/or through the devices that contain GPS, are also contemplated.

Then, the XML files of each of these datasheets are created automatically as outputs of the tool. These files allow us to process datasheets automatically, for
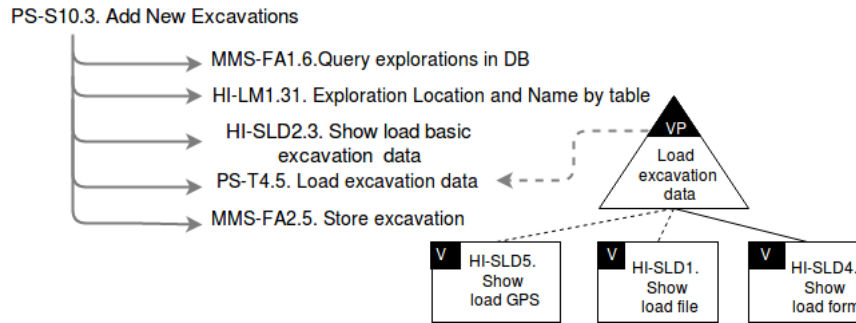
**Fig. 4.** Functionality representation of *Load excavations* in a functional datasheet

instance for posterior validation [19, 20]; and create a preliminary structure of reusable components (artifact 4) [2, 3], which will be later part of the architecture of the platform (artifact 5). These components (artifact 6) must be implemented in the last step.

## 4  A Preliminary Evaluation of Reuse

In this section we performed an evaluation for analyzing the degree of reuse achieved in two of the artifacts created and described in the previous section.

Firstly, we have analyzed the reuse of the service taxonomy considering the number of services that we have defined for the paleontological sub-domain. For this evaluation we have analyzed three types of services:

- *Completely Reusable Services (CRS)* are those services of the geographical domain or some sub-domain, which are used without modification by the functionalities of the paleontological SPL. Examples of these services are those that are black and red in Figure 3 and that are related to this sub-domain.
- *Reusable Services for Specialization (RSS)* are new services generated from services already defined for other domains. Examples of these services are the blue ones in Figure 3.
- *Unreusable Services (URS)* are services created completely new without relationship with the services of the previously developed taxonomy.

Based on these types of services, and from a total of 120 services defined as useful, we identified a 25% as CRS, 58.3 % as RSS and 16.6% as URS. Figure 5a) shows these percentages graphically.

Then, for analyzing the functionalities, we simulated the creation of a product in which we chose a set of ten functionalities (defined in the functional datasheets) as part of the SPL platform, and we have randomly instantiated them in order to generate a new product. To illustrate this process, in Figure 6 we show the *Load Excavations* datasheet, in which we have instantiated the

variability for loading information by using forms or by means of GPS devices. In this way, the resulting product will offer both options for loading an excavation.
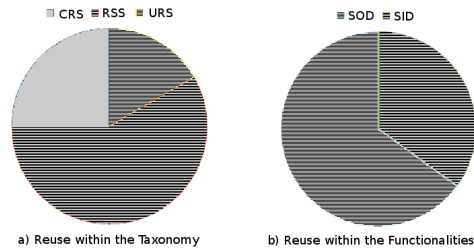


**Fig. 5.** Analysis of the reuse achieved in the taxonomy and instantiated datasheets
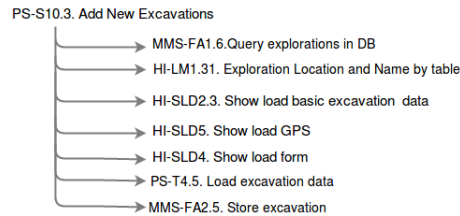


**Fig. 6.** *Load Excavations* functionality instantiated for a particular product

A similar process was carried out for each of the ten functionalities used for the creation of the product. For analyzing the reuse, we have analyzed the number of *Services of Other Domains (SOD)* used in the functionalities and the number of *Services Included in the Domain (SID)* of the taxonomy. The SODs are those that we have classified as the sum of the CRSs and RSSs, and the SIDs are the URSs. From this analysis, the results show, in Figure 5b), a percentage of approximately 65% of SOD and 35% of SID. This result is reasonable considering that many functionalities use RSS services (specialized for the sub-domain).

## 5  Conclusion and Future Work

In this work, we have shown a process for SPL development oriented to the reuse of software artifacts within a hierarchy of domains. Based on previous work, we have generated artifacts in a new sub-domain - the paleontological - illustrating a possible reuse of our platform. At the same time, we have performed a preliminary analysis of the reuse achieved in the construction of some of these artifacts showing promising results.

As future work, we will extend the process for building the artifacts that are involved in the application engineering of the SPL.

# References

1. A. Buccella, M. Pol'la, A.C., Arias, M.: A variability representation approach based on domain service taxonomies and their dependencies. In: Internationl Conference of the Chilean Computer Science Society (SCCC). Talca, Chile (2014)
2. Arias, M., Buccella, A., Cechich, A.: Smooth transition from abstract to concrete spl components: a client-server implementation for the geographic domain. In: Proceedings of the IEEE ARGENCON'16. GRSS: IEEE GRSS, Buenos Aires, Argentina (2016)
3. Arias, M., Buccella, A., Cechich, A.: Un marco de trabajo para la gestión de requerimientos de software en líneas de productos de software. In: CLEI 2017 : XLIII Conferencia Latinoamericana en Informática. SLISW - Simposio Latinoamericano de Ingeniería de Software. To appear (2017)
4. Bashroush, R., Garba, M., Rabiser, R., Groher, I., Botterweck, G.: Case tool support for variability management in software product lines. ACM Comput. Surv. 50(1), 14:1–14:45 (Mar 2017), `http://doi.acm.org/10.1145/3034827`
5. Benavides, D., Segura, S., Ruiz-Cortés, A.: Automated analysis of feature models 20 years later: A literature review. Information Systems 35, 615–636 (September 2010)
6. Bosch, J.: Design and use of software architectures: adopting and evolving a product-line approach. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA (2000)
7. Bosch, J.: From software product lines to software ecosystems. In: Proceedings of the 13th International Software Product Line Conference. pp. 111–119. SPLC '09, Carnegie Mellon University, Pittsburgh, PA, USA (2009)
8. Braun, G., Pol'La, M., Cecchi, L., Buccella, A., Fillottrani, P., Cechich, A.: A dl semantics for reasoning over ovm-based variability models. In: 30th International Workshop on Description Logics (DL17). To appear, France (2017)
9. Buccella, A., Cechich, A., Arias, M., Pol'la, M., Doldan, S., Morsan, E.: Towards systematic software reuse of gis: Insights from a case study. Computers & Geosciences 54(0), 9 – 20 (2013), `http://www.sciencedirect.com/science/article/pii/S0098300412003913`
10. Buccella, A., Cechich, A., Pol'la, M., Arias, M., Doldan, S., Morsan, E.: Marine ecology service reuse through taxonomy-oriented SPL development. Computers & Geosciences 73(0), 108 – 121 (2014), `http://www.sciencedirect.com/science/article/pii/S0098300414002155`
11. Burrough, P., McDonnell, R.: Principles of Geographical Information Systems. Oxford University Press (1998)
12. Capilla, R., Snchez, A., Dueas, J.C.: An analysis of variability modeling and management tools for product line development. In: Proceedings of the Software and Services Variability Management Workshop: Concepts, Techniques, and Tools. pp. 32–47 (2007)
13. Clements, P.C., Northrop, L.: Software Product Lines : Practices and Patterns. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (2001)
14. Mancuso, M., Buccella, A., Cechich, A., Arias, M., Pol'la, M.: Datasheet modeler: Una herramienta de soporte para el desarrollo de funcionalidades en líneas de

productos de software. In: Proceedings of the CACIC'15: XXI Argentine Congress of Computer Science. Junin, Buenos Aires, Argentina (2015)

15. Manikas, K., Hansen, K.M.: Software ecosystems a systematic literature review. Journal of Systems and Software 86(5), 1294 – 1306 (2013)

16. Munir, Q., Shahid, M.: Software Product Line: Survey of Tools. Master's thesis, Linkopings University, Linkopings, Sweden (2010)

17. Parent, C., Spaccapietra, S., Zimányi, E.: Spatio-temporal conceptual models: data structures + space + time. In: Proceedings of the GIS '99: 7th ACM International Symposium on Advances in Geographic Information Systems. pp. 26–33. ACM Press, New York, NY, USA (1999)

18. Pohl, K., Böckle, G., Linden, F.J.v.d.: Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2005)

19. Pol'la, M., Arias, M., Buccella, A., Cechich, A.: Un sistema de anotaciones para la especificación de componentes de una línea de productos de software. Revista Tecnología y Ciencia de la Universidad Tecnológica Nacional 6(1), 116–122 (2015)

20. Pol'la, M., Buccella, A., Arias, A., Cechich, A.: Sevatax: Service taxonomy selection & validation process for spl development. In: XXXIV International Conference of the Chilean Society of Computer Science (SCCC'15). IEEE Computer Society Press, Santiago, Chile (2015)