Computers & Geosciences 54 (2013) 9–20

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo

Towards systematic software reuse of GIS: Insights from a case study

Agustina Buccella [a,*], Alejandra Cechich [a], Maximiliano Arias [a], Matias Pol'la [a], Maria del Socorro Doldan [b], Enrique Morsan [b]

[a] GIISCO Research Group, Departamento de Ingeniería de Sistemas—Facultad de Informática, Universidad Nacional del Comahue, Buenos Aires 1400, Neuquen 8300, Argentina
[b] Instituto de Biología Marina y Pesquera "Almirante Storni", Universidad Nacional del Comahue—Ministerio de Producción de Río Negro, San Antonio Oeste, Argentina

ARTICLE INFO

Article history:
Received 30 May 2012
Received in revised form
8 September 2012
Accepted 19 November 2012
Available online 29 November 2012

Keywords:
Geographic information systems
ISO 19119 std.
Software product lines
Reused services
Marine ecology
Geographic open source tools

ABSTRACT

With the development and adoption of geographic information systems, there is an increasingly amount of software resources being stored or recorded as products to be reused. At the same time, complexity of geographic services is addressed through standardization, which allows developers reaching higher quality levels. In this paper, we introduce our domain-oriented approach to developing geographic software product lines focusing on the experiences collected from a case study. It was developed in the Marine Ecology Domain (Patagonia, Argentina) and illustrates insights of the process.

1. Introduction

Nowadays, it is easy to see the great expansion of geographic information systems (GIS) used to represent common instances of the real world. Then, as any software engineering discipline, also engineering GIS is constantly developing new technologies, methods, tools and approaches to increase ability of software engineering professionals to improve cost-effectiveness, predictability of quality and time-to-market. Probably the most promising approach to achieve this is the use of software artifacts in multiple contexts, i.e. software reuse.

Software product lines (SPL) is one of the forms through extensive reuse of software that has evolved during the last decade. A software product line (Clements and Northrop, 2001; Pohl et al., 2005; van der Linden et al., 2007) consists of a product line architecture, a set of shared components and a set of products. Each product derives a product architecture from the product line architecture; selects, instantiates and configures product line components; and adds, if necessary, product specific code. The introduction or adoption of a product line approach is a process that requires technical, process, organizational and business changes and requires a rather fundamental rethinking of the whole approach to software development. In particular, every SPL project should identify commonalities and variabilities of a particular problem domain. The definition and analysis of which requirements will be part of the line and which others will be product-specific are not easy tasks and strongly depend on the domain.

Researchers, developers and users of GIS agree on different particularities must be taken into account when the geographic domain is involved. GIS are considered as members of an area emerging from general-purpose information systems but taking aspects from other areas such as cartography, topology, etc. Another important aspect to be considered is that GIS include a group of more specific domains or branches, each of them focused on its own particularities. We can find a first classification in which the geographic domain is divided into three main branches[1] (Bonnett, 2008): *human geography*, focused on the study of patterns and processes of the human society; *physical geography*, focused on the productions and interactions of organisms, climate, soil, water, and landforms, over the nature environment; and *environmental geography*, combining the physical and human phenomenon to analyze the interactions between the environment and humans. In addition, within the physical geography we can find other areas or domains including the oceanography and climatology domains; and at the same time, the oceanography domain includes other subdomains such as marine geology, marine ecology, marine fishery, etc.

---

* Corresponding author. Tel.: +54 299 4490300x435; fax: +54 299 4490313.
E-mail addresses: agustina.buccella@fai.uncoma.edu.ar, agustina_buccella@yahoo.com.ar (A. Buccella), alejandra.cechich@fai.uncoma.edu.ar (A. Cechich), ariasmaxi89@gmail.com (M. Arias), matias_polla@hotmail.com (M. Pol'la), msdoldan@gmail.com (M.d.S. Doldan), qmorsan@gmail.com (E. Morsan).

[1] http://en.wikipedia.org/wiki/Geography.

0098-3004/$ - see front matter © 2013 Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.cageo.2012.11.014

The previous characterization perfectly fits an SPL development. The great number of implemented products share a set of common features[2] that are in general available to be used by any of them. In addition, we can find another great number of features that are product-specific and are only implemented by some products. In this way, the geographic domain requires a different analysis and its particularities must be carefully analyzed. First of all, we must take into account aspects that are not usual in general-purpose information systems such as special data types and operations, analysis and visualization procedures, long transactions, huge storage space, etc. Secondly, the range of different disciplines in which the geographic area is branched should help to organize the different domains and products that can be developed.

At this point, it should be interesting to analyze the possibility of building reference models that guide SPL instantiation. To do so, some standardization efforts reveal that the development of a GIS might be guided from a special set of defined rules that help designers and developers to build this type of systems and define some quality requirements such as interoperability, modifiability, etc. The Open Geospatial Consortium (OGC)[3] and the ISO Technical Committee 211 (ISO/TC 211)[4] are two different entities working together on the definition of such rules. Here, we focus on the services defined by the Service Architecture standard (OpenGIS Service Architecture)[5] and the ISO/DIS 19119,[6] as specializations of a taxonomy of geographic services.

This paper is organized as follows. The next section presents related work in the literature taking into account GIS reuse as well as approaches to SPL development. Section 3 introduces knowledge and design decisions that constitute basic background. Section 4 describes the methodology for creating a product-line for the geographic domain and the marine ecology subdomain as well as our experiences from applying the approach. Future work and conclusions are discussed afterwards.

2. Related work

From the point of view of GIS applications, systematic reuse has been approached from pattern-oriented techniques and many times applied to particular domains. This is the case of analysis patterns for reusing geographic database design of urban area planning (Lisboa Filho et al., 2002); analysis and design patterns for improving catalogues and libraries (Câmara et al., 2000; Sodré et al., 2005); and design patterns for developing GIS applications with objects (Gordillo et al., 1999). However, only some efforts have recently incorporated reuse through components as a way of integrating domain functionality (Dobrica et al., 2010). From the point of view of reusing through SPL, GIS applications are treated as instantiations of multiple product lines (MPL) characterized by ultra-large-scale systems, software ecosystems, or product populations. Unfortunately, there are many approaches to deal with MPL modeling; and even MPL capabilities are not clearly established yet (Holl et al., 2012).

In this way, it is necessary to analyze other set of paradigms for software reuse coming from the software engineering area. One of them is the Software Product Line Engineering (SPLE) (Bosch, 2000), which is focused on the definition of domain knowledge that can be reused in many systems sharing a domain (Schäfer et al., 1993). There exists several proposals in the literature describing different methodologies for developing software product lines (Bosch, 2000; Clements and Northrop, 2001; Pohl et al., 2005; van der Linden et al., 2007). All of them propose a division into common and variable aspects of the product line, and a set of tasks or activities that must be done to specify and implement these aspects. In addition, several approaches propose different ways to model variability in SPL (Czarnecki and Eisenecker, 2000; Kang et al., 1998; Pohl et al., 2005). For example in Pohl et al. (2005), van der Linden et al. (2007) authors define a meta model and a graphical notation for representing a variability model.

Other set of proposals (Czarnecki and Eisenecker, 2000; Kang et al., 1990, 1998) define variability models based on the feature oriented paradigm within the domain analysis area (Arango, 1994). In general, several of these proposals provide extensions of the Feature-Oriented Domain Analysis (FODA) proposed by Kang et al. (1990). FODA defines a graphical notation for representing Feature Models (FM) at the requirement level; however other work, presented in Kang et al. (1998), extends FODA to the software design phase supporting the definition of reusable domain artifacts; and proposes mapping FM to an architectural design.

Also, there exist the possibility of integrating feature diagrams to UML models in order to provide more semantics and a familiar development environment (based on a standard modeling language) (Czarnecki et al., 2005a; Possompes et al., 2011; Vranic and Snirc, 2006). Finally, several works propose extensions of UML diagrams to represent the variability of the domains (Dobrica and Niemelä, 2008; Gomaa, 2004; Pohl et al., 2005; Rumpe and Robert, 2011; Ziadi and Jézéquel, 2006).

By comparing feature-oriented to UML-based models (Reinhartz-Berger and Tsoury, 2011; Chen and Babar, 2011), empirical evaluations arrive to the conclusion that all the methodologies analyzed have limitations to represent and evaluate variability. Therefore, so far there is no standard methodology and/or tool to be used in a software product line development.

In sum, reusing GIS is still waiting for the advantages of using SPL modeling. In general, creating SPL for GIS is seen only as another application of product line development; however, both research areas might be combined taking advantage of their most promising aspects (such as the existence of standards) for guiding GIS reuse.

3. Background

This section outlines the basic decisions we have made when developed the SPL. Firstly, we looked at existing standards that might help us build a reference architecture. Secondly, we decided to model a component-based SPL considering that support for instantiating architectural layers involves implementing black-box components for reuse. And finally, we decided to implement products by using open source code in order to increase spreading and possibility of use.

3.1. Standards for GIS development

GIS standards are important to coordinate exchange and use of spatial data. Standards are meant to reduce the cost of data collection and to promote the reuse of existing information. GIS standards encompass several issues relating to data and process. Data standards address issues of classification, content, transfer, and usability. Classification and content describe data theme development; attributes and features important to a dataset or

---

[2] In this work a feature describes the functional and quality characteristics of a system (Bosch, 2000; Pohl et al., 2005).

[3] http://www.opengeospatial.org/.

[4] http://www.isotc211.org/.

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

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

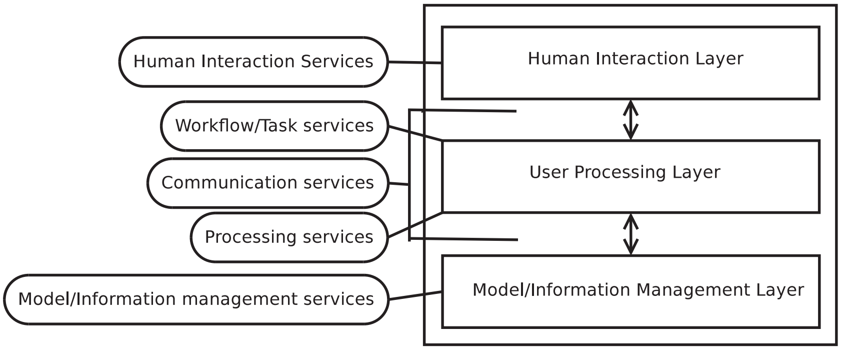


Fig. 1. Types of services in the three-tier architecture.

user community. They may describe the appropriate scale or resolution of a dataset. Data transfer standards address how to format data products for use and reuse. One aspect of data usability is the definition of metadata, which allows users to discover, review, and retrieve existing datasets. For instance, many federal agencies in United States have developed content standards for GIS datasets and most available federal datasets include some form of agency standard particularly to develop the content of a dataset. For example, the National Hydrography Dataset[7] uses existing standards for data transfer, and metadata description, but has developed agency standards for the features and attributes of the dataset, as well as the standards of spatial resolution, quality control, etc. As another case, the Federal Information Processing System (FIPS) Standards are developed to standardize data and processes among federal agencies. Their goal is to gain efficiency and economy through widespread use. An example of a widely used FIPS standard is the Geographic Names Information System which is a database of geographic names (FIPS55-DC3, 1998). On the other hand, the Federal Geographic Data Committee (FGDC) Content Standard for Spatial Metadata (FGDC-STD-001-1998, 1998), defines how document the contents of a spatial dataset, and is used as input to GIS enabled search methods. The FDGC is also developing the Framework,[8] a collaborative effort to create a widely available source of basic geographic data. It provides the most common data themes geographic data users need, as well as an environment to support the development and use of these data. More widely used, industry standards are very popular in the GIS domain. The OGC is an international industry consortium of more than 220 companies, government agencies and universities participating in a consensus process to develop publicly available geo-processing specifications. Standards include: abstract data models, data transfer standards, and web-enabled GIS data accessibility. And of course, the International Organization for Standardization (ISO) is the primary international standards organization for information technology. The ISO/TC 211 develops GIS standards such as those relating to data models, metadata, spatial referencing, and quality.

Within the wide range of defined standards, we are particularly interested in the Service Architecture standard (defined in OpenGIS Service Architecture) and the ISO/DIS 19119 std. They define a classification of geographic services according to the requirements of spatial and temporal information organized into a reference architecture using a multi-tier architecture model. Thus, it is accepted for the GIS community that every geographic system should be built by using at least a three-tier architecture. It contains a *human interaction tier*, responsible for the interaction with the user; a *user processing tier*, responsible for the functionality required by the user; and a *model/information management tier*, responsible for physical data storage and data management. The main advantage of this architecture is the required separation of the functionality into three different independent layers that interact only through their well-defined interfaces. This enables a developer to work over each one of these layers without interfering on the others; therefore, this architecture provides increased modifiability and scalability. In addition, the standard defines a classification and a taxonomy of the set of generic services that each layer should provide. It is important to analyze how these generic services fit in a simple three-tier architecture. Fig. 1 shows one simple reference model proposed by the standard, based on three main layers together with the services that each layer should implement.

3.2. Component-Based Software Development

Component-Based Software Development (CBSD) advocates the use of pre-fabricated pieces, perhaps developed at different times, by different people, and possibly with different uses in mind. One of the main goals of CBSD, once again, is the reduction of development times, costs, and efforts, while improving quality of the final application due to the (re)use of software components already developed, tested and validated. Of course, such achievements usually do not depend only on following a CBSD, but also on some other organizational and environmental factors. However, adopting a component model allowed us to operate at two levels: firstly, the component model defines how to construct an individual component; and secondly, it can enforce global behavior on how a set of components in a component-based system will communicate and interact with each other. Particularly, the component model specifies how interfaces should be defined and the elements that should be included in an interface definition. We adopted Enterprise Java Beans (EJB) as component model since its mechanisms handle common concerns such us persistence, integrity and security in a standard way, leaving us free to focus on the SPL modeling problem. Besides, EJB perfectly support our main intent: building components as much as encapsulated as possible and reusing existing ones (off-the-shelf) only by adapting their interfaces.

3.3. SPL implementation

We have decided to instantiate common features of the SPL by using open source tools.[9] The main reason to do so was

---

[7] http://nhd.usgs.gov/.

[8] http://www.fgdc.gov/framework.

[9] *Tools* here, are considered as coarse-grained components and treated as units of composition.

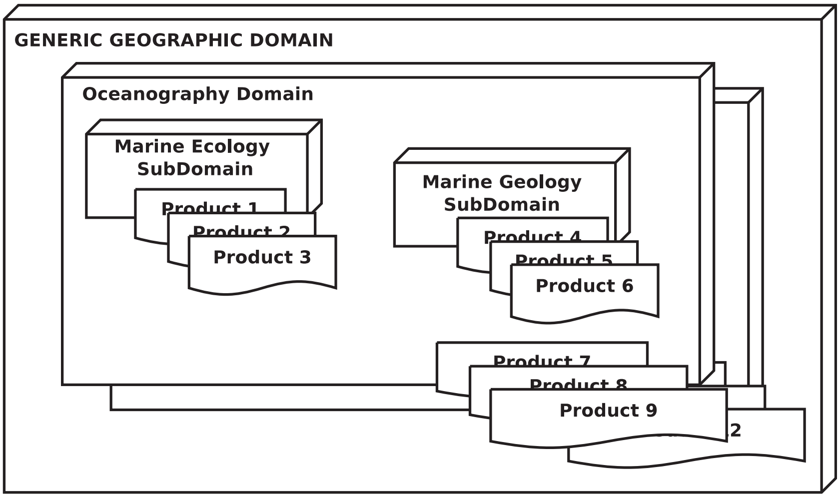


Fig. 2. Software product line within the geographic domain.

considering that during the last 15 years there was a great explosion of open source geographic tools, allowing multiple possibilities of reusing lower-cost fine- and coarse-grained components (Fils et al., 2009; Diviacco, 2005). By simply searching on the Web, we can obtain hundreds of pages offering tools that implement different solutions for GIS development. It is true that the use of proprietary software is a common practice in industry, however open tools are very useful too. All these possibilities should not be ignored. For instance, PostGIS[10] and MySQL Spatial[11] are only two examples of open source tools capable of satisfying the set of services defined in the *model/information management services* of the standard (Section 3.1).

However, the huge number of geographic tools makes really complex the decision about which specific tools and components might be reused in a specific product. We had to carefully evaluate options to build the common part of our SPL. Selecting suitable tools was a complex task; and particularly for our approach it was very important that the set of tools were compatible to work together and allow developers to implement the architecture defined by the SPL. Then, the implementation of the line paid special attention to component integration. Several important aspects were taken into account, such as, the tools must present continuity in their developments and provide a good documentation; the possible information extracted from the forums must be extensive and useful; they must be flexible enough, that is, they must be easily extensible, etc.

4. Developing and reusing an SPL for the marine ecology subdomain

By considering the main aspects and characteristics of geographic systems described previously, and the needs of reusing geographic services, the GIS domain becomes suitable for following a software product line approach through the development of software components. However, although the ISO standard is useful to understand the wide range of services every GIS is able to offer, these services are defined in a very generic and abstract way. Then, the task of creating an SPL should consider these particularities and provide mechanisms to implement them. In this way, we propose an hybrid approach, combining top-down and bottom-up mechanisms in order to consider requirements depending on the domain-level being analyzed. When more specific domains are analyzed, they must include their own requirements plus those defined for more general domains in which they are included. When these general domains are considered, the process is as the opposite one. Fig. 2 shows a framework for our hybrid approach, named as *generic geographic domain*. For example, we can see that the *marine ecology* subdomain is implemented as a product line in which a set of products can be built. The same is applied to the *marine geology* subdomain. In addition, these two product lines, can be part of a more generic domain, named *oceanography*, in which another set of products will also be implemented.

We have defined a development methodology, which combines advantages of several methodologies widely referenced in academy and industry (Bosch, 2000; Czarnecki et al., 2005b; Kang et al., 1990; Pohl et al., 2005) and we have extended them in order to apply our level-domain view. Fig. 3 shows the main activities of the methodology concerning the domain engineering phase (as defined in Pohl et al., 2005). This phase is divided into two types of analyses: *domain* and *organizational*. At the domain analysis we define three processes which impact directly on the activities defined by the organizational analysis. The gray rectangles of the domain analysis denote the main processes and the black arrows denote their relations to the activities of the organizational analysis. Another important aspect of the figure is the big gray arrow on the left side. It denotes the influence of the ISO 19119 std. over the three domain levels (generic, domain and subdomain) through a service taxonomy. In the figure we show the subdomain level in which the common services will use the more specific services defined in the taxonomy.

On the other hand, at the organizational analysis, the domain-level information is applied to the processes defined in the figure. The information modeled and implemented at the organizational analysis will be a subset of the information captured at the domain-level.

Next subsections describe particularities of the subdomain, and the software product line implemented by following our leveled approach.

---

[10] http://postgis.refractions.net/.

[11] http://dev.mysql.com/doc/refman/5.0/en/gis-introduction.html.

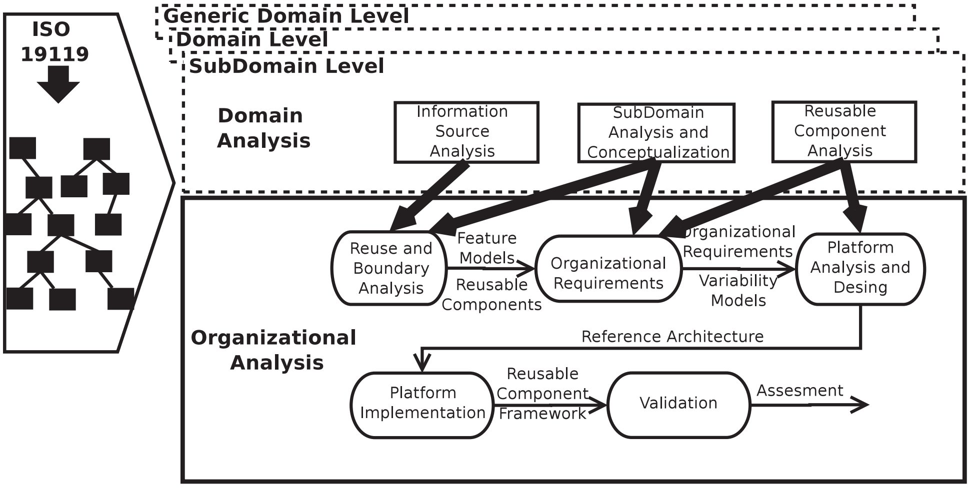


Fig. 3. Activities of the domain engineering process by applying a domain-level view.

4.1. The marine ecology subdomain

In general terms, the marine ecology subdomain involves the scientific study of the interdependence of all organisms living in the marine-life habitat, and their interactions with each other and the surrounding environment. Here, it is important to analyze the abiotic factors such as temperature, salinity, light, etc., and biotic factors involving the relationships between the organisms of one particular species, that is, how the behavior of one organism influences another. Also, the subdomain involves analyses about how the human activity impacts on ecosystems and populations. In this way, a very large set of activities and analyses are performed by experts (mainly biologists) in order to arrive to conclusions about life and conservation of the organisms. Our work, along with two expert organizations in this subdomain – the Instituto de Biología Marina y Pesquera "Almirante Storni"[12] (IBMPAS) and the Centro Nacional Patagónico[13] (CENPAT-CONICET) – has allowed us to define an interesting set of activities and goals, and to abstract them in order to be applied to the whole marine ecology community. Both organizations are responsible for storing and analyzing information about checklist of species (i.e. censuses) in three gulfs of the Argentinean Patagonia (*San Matías, San Jorge and Nuevo Gulfs*). Each census, performed once a year, collects information about the population of benthic species living in this area. This information is then used for spatial processing in order to obtain information about spatial distribution of the species, population variation patterns in different scales, etc. In particular, bivalve species, such as the native oyster *Ostrea puelchana* are studied at the IBMPAS and *echinoderms* species are studied at CENPAT-CONICET. Three main aspects of these organisms are encompassed by the investigations of both organizations:

- *Distribution and classification*: Distribution of species is addressed by taking samples of fauna and sediment at different stations in the three Gulfs. Organisms are classified according to age or size in order to follow the temporal variation of the population structure. Thus, changes on the population and distribution of the organisms can be evaluated.
- *Fishing*: Some bivalve species have been exploited in the Gulfs for many years. At present time, some fishery devices are forbidden because of the disturbances they cause on the benthic ecosystem. However, other new ones are being used, new species are targeted and new areas are being fished. Thus, it is necessary to analyze the impact of fishing on populations and distributions. To do so, maps of the zones in which fishing is allowed are overlapped with maps of the zones in which the organisms are distributed in order to analyze the possible impacts.
- *Impact of other organisms*: Zones in which inhabits invasive kelp such as the *undaria pinnatifida* must be registered and mapped in order to identify the colonization of new areas. Also, the *Bonamia* sp. bivalve parasite presence and the distribution of infected bivalves must be monitored and mapped in order to control the spread of epizootic events. Thus, it is possible to find relationships between these zones and environmental factors that may allow these species to establish and to determine how much native populations and their distribution are affected.

By working together (informatics and biologists), we could define a set of activities needed to perform the analyses required by each of the main aspects aforementioned. The main line of work was focused on defining which of the activities might be supported by computer systems to arrive to better and faster conclusions. For example, modern graphical data analyses are fundamental to understand the interaction among species, between species and disturbance factors, and the response of the populations. This information has great importance and must be considered when planning for conservation.

In the next section, we analyze the set of defined activities and transform them as a set of services of a GIS. Our contribution produces standard and reusable services, so they can be (re)used by the marine ecology community.

4.2. Building a software product line at the subdomain level

First of all, let us describe the domain analysis steps (from Fig. 3).

- *Information source analysis*: This process involved three sources to be considered: standards, existing applications and domain experts. In the geographic domain, the standard information was obtained from the taxonomy defined in the ISO 19119 std. and specialized according to the requirements of the subdomain. For instance in Table 1[14] we show only some of the

---

[12] http://ibmpas.org/.

[13] http://www.cenpat.edu.ar/.

[14] In the table we can see that the term "feature" has a different meaning. In the ISO 19119 std. a "feature" is defined as an abstraction of real world phenomena. We refer this term as "geographic feature" in order to differentiate it from the term "feature" defined by the domain engineering process.

Table 1
Part of the specific geographic services required by the marine ecology subdomain.

Categories of the ISO 19119	Service	Specific features
Geographic human interaction	S1. Geographic viewer	(S1.1) (a) Show zones. (b) Show stations within a zone. (c) …(S1.2) (a) Show/hide the zone layer. (b) Show/hide the density layer. (c) …
	S2. Geographic feature editor	(S2.1) (a) Show a map with the location of zones. (b) Show a map with the abundance of species…
Geographic model/information management services	S3. Feature access	(S3.1) (a) Query zones of density of species. (b) Query zones in which the population of species are higher than a specific value…
	S4. Map access	(S4.1) (a) Query the gulf area image in a map. (b) …
Spatial processing services	S5. Proximity analysis	(S5.1) (a) Obtain the location of stations within a specific zone. (b)…
Temporal processing services	S6. Temporal proximity analysis	(S6.1) (a) Obtain the number of specimens of specific species in a zone at different times. (b) …
Thematic processing services	S7. Change detection	(S7.1) (a) Find changes among densities of species on different censuses. (b) …

specific services required for this subdomain and classified by the standard.

Secondly, the existing applications[15] were analyzed from the geographic tools that are currently used by the organizations in this subdomain. In our analysis, we observed that few organizations had applications involving geographic information. They had used only office software tools in which almost all the tasks were made manually. However, as we described in Section 3, there exist several systems and open source tools providing services that can be useful for the requirements in the geographic domain. In this way, we decided to apply an evolutionary approach (Bosch, 2000) by considering a set geographic open source software tools and combining them in a suitable way. In Pernich et al. (2010) we have classified and analyzed some of these tools.[16] This analysis allowed us to define a set of general requirements included in the generic geographic and oceanography domain levels (Fig. 2) that must be implemented by all products. For example, within the *geographic human interaction services* (extracted from the ISO 19119 std.) we defined general visualization services such as pan and zoom, select and hide layer menus, etc. Then, within the oceanography domain level we also defined another set of general requirements but only applicable to its products; for instance, showing the different sea depths in specific zones (in our case in the gulfs). Then, within the lowest level, marine ecology subdomain, we defined another set of requirements that are specific for the products created in this level (as described in Table 1).

Finally, the project team was composed by domain experts (people experienced in the marine ecology subdomain) and software engineers and developers.

- *Subdomain analysis and conceptualization*: Here, the information recovered in the previous process was used to analyze and organize the features or services that the subdomain should offer together with the general features derived from the upper domains. Thus, Table 1 was refined by considering the information provided by domain experts and the software tools available. Next, in this process the subdomain must be conceptualized by different software artifacts (such as class models, process models, etc.) when it is possible. In our work, we firstly defined the conceptual model which was part of the last layer of the architecture of the SPL (*model/information management layer*). This model was then used by all the functional features in order to implement their functionalities. In addition, we defined a *feature template* in order to describe the way each feature is designed. Thus, for each feature defined by the software product line (Table 1), a new template must be created. All these documents conform the detailed design of the features that will be implemented. The template contains the id of the feature, its name, the type (inherited, when it is a general feature belonging to upper domains; or specific, when it is specific of the domain-level in which it is defined), the set of open source tools that can implement it, and the software artifact used to represent its functionality. Table 2 shows the feature template filled in for two features that we defined in our work; one of them is an inherited feature from the oceanography domain, and the other is an specific one for the marine ecology domain. In this case, the table only shows sequence diagrams[17] for the *models* item, but we also used another UML software artifacts (specifically, use cases and collaboration diagrams) to represent different aspects of the features.[18] For example, firstly the table shows the feature *Find changes among densities of species on different censuses*, which is *specific* of the marine ecology subdomain and can be implemented by using all those geographic *open source tools*.
- *Reusable component analysis*: This process identifies the set of reusable components that could be used to implement the features defined in the last process.[19] By considering the particularities of geographic systems, described in Section 3, we defined a layered architectural style in order to promote modifiability and scalability. Apart from the usual advantages of modular software with well-defined interfaces, the layered architecture is intended to allow any of the layers to be upgraded or changed independently in response to changes in requirements or technology. Thus, we firstly defined the reference architecture with three main layers, *geographic model*, *geographic processing*, and *user interface*. This design decision follows the decisions rules defined in the standard in which a *n*-tier architecture is proposed. In addition, for each layer, we specified the generic components according to the

---

[15] *Existing applications* are the software tools that the organizations used for doing their daily works.

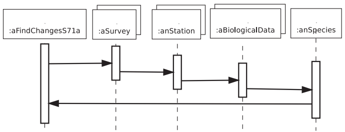
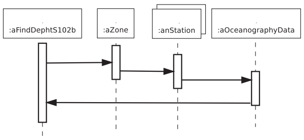
[16] In our work we analyzed open source tools due to they fulfill the main characteristics described in Section 3.3. The availability of the source code and the right to redistribute modifications and improvements to the code are two of the most important advantages prioritized in this work.

[17] The blurred sequence diagram in the figure is included just for illustration. In the next sections, we will show a sequence diagram and a feature development in detail.

[18] We chose UML to represent the analysis and design of the product line because it was a common language for all members of the project and its characteristics in terms of expressiveness are similar to other proposals. (See Section 2).

[19] OD acronym is used to denote services belonging to the oceanography domain.

Table 2
Two different features according our feature template.

Id	(S7.1) (a)
Name	Find changes among densities of species on different censuses
Type	Specific
Open source tools	OpenLayers, Ka-Map, MapServer GeoServer, GeoTools,…
Models	
Id	OD3.2b[19]
Name	Query the depth of a specific area of the ocean
Type	Inherited from oceanography domain
Open source tools	OpenLayers, Ka-Map, GeoServer GeoTool, MapServer,…
Models	

features defined in the last processes. For instance, the second layer defines features involving processing—part of them are features S5–S7 described in Table 1.

4.2.1. *Organizational analysis steps*

- *Reuse and boundary analysis*: This activity defines the organizational boundary, commonality, and variability features. Thus, by considering the features specified in the *subdomain analysis and conceptualization* process and the information from domain experts, the scope of the product line must be defined. Then, this activity analyzes which of the features can be implemented by geographic open source tools. In our proposal, we modified and added some items of the feature template defined in the *subdomain analysis and conceptualization* process. Firstly, we refined the conceptual model and the models used to represent each functional feature. In addition, we added the *variability model* item to the template in order to represent, when necessary, the variability included in each feature. Fig. 4 shows the variability model associated with the sequence diagram of the feature S7.1 (Table 2). We used the notation of variability models proposed by Pohl et al. (2005), called *orthogonal variability model*, due to its clarity on defining variability over UML software artifacts. In the figure, we observe different objects (part of the subdomain model) used to show information about the distributions of species in different censuses. The feature is associated to a variability model in order to allow different representations of the returned data. In the variability model, data are always shown in a tabular way and histograms and labels are alternative choices.
- *Organizational requirements*: In this activity, we used the information of the commonality and variability identified in the last activity and the information provided by the *subdomain analysis and conceptualization* and *reusable component analysis* processes. The main goal here is to define the range of products and features that the line is able to implement. We defined a product/feature matrix indicating which subset of features will be part of the product-line and which subset of features will be product-specific. In this work, we followed a minimalist approach (Bosch, 2000), that is, only the features used in all products are part of the product line. Thus, our software product line is then seen as a *platform* (van der Linden et al., 2007).
- *Platform analysis and design*: This activity builds the reference architecture based on the features defined in the previous activities and processes. The preliminary structure of reusable components defined in the *reusable component analysis* process is reorganized and refined. In our work, we performed two tasks. Firstly, we made decisions about feature allocation into software components. In addition, we refined components in order to add variability, and we decided, for each feature and variability, the way they must be implemented as software components.[20]

  Secondly, as part of this activity, the three-layered reference architecture was restructured. We decided not to add any new layer, but we reorganized and defined the specific components for each layer. In this way, we designed the software line by following a component-based development. We had to analyze the different design decisions in order to improve reusability. For instance, we decided to design features of different type in different software components. Fig. 5 shows components of each layer of the SPL. The darker components implement services belonging to the generic geographic domain, for instance the *visualization features* component implement several GD1 services[21] involving pan and zoom, scale, refresh tools, etc. The gray components implement services of the oceanography domain, such as the *oceanography proximity service* component, involving query attributes of zones of the ocean. Finally, white components implement the specific services of the marine ecology domain (some of them were detailed in Table 1). For example, the *Change Detection* component of the geographic processing layer implements the feature S7.1 and its variability as we have described in the detailed design represented by Fig. 4. Finally, the components

---

[20] Notice that software components developed for reuse are the result of domain knowledge, design decisions, and guesstimates about the future use of the component. For brevity and clarity reasons, we have omitted here a deeper discussion for our case.

[21] GD acronym is used to denote services belonging to the generic geographic domain.

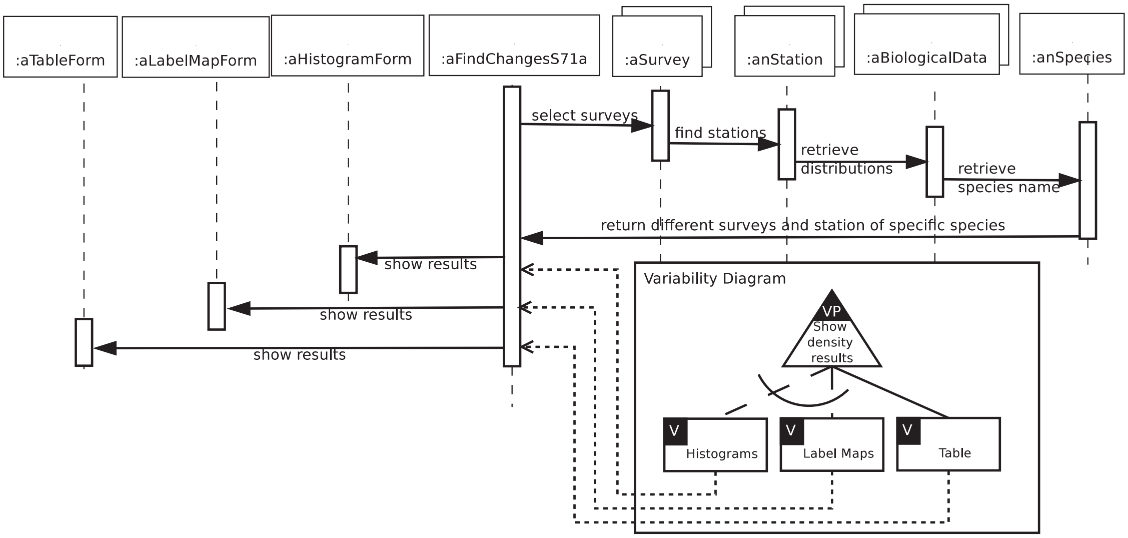


Fig. 4. Variability model item associated with the sequence diagram of the feature S7.1: *Find changes among densities of species.*

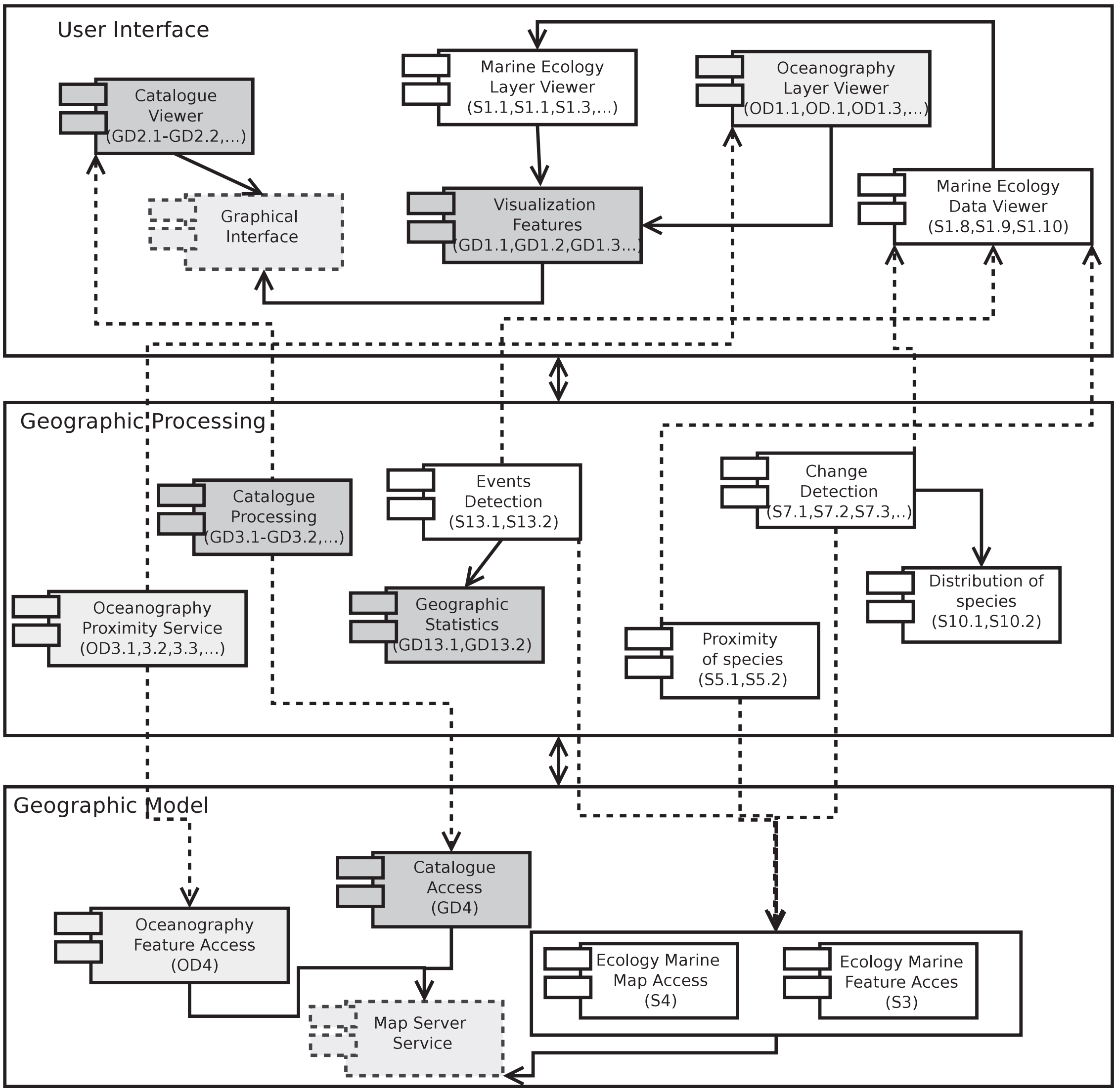


Fig. 5. Reference architecture of the marine ecology product line.

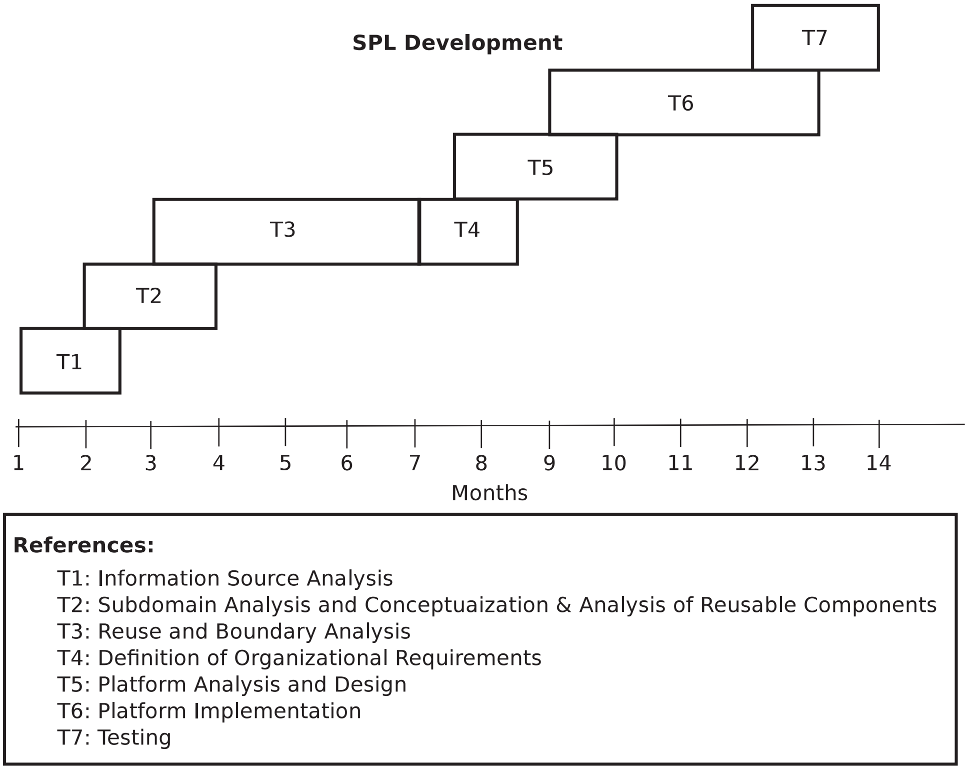


Fig. 6. Time required to perform the activities involved in the SPL development.

represented by dash lines indicate external components used to implement our services.

- *Platform implementation*: In this activity components that are common for all products, that is, components of the line, are implemented. In particular, we firstly used the information of the open source tools used to implement each feature (defined in the feature template) and we defined the software components used to implement the whole architecture. Although the information in the templates was useful to provide an idea of the specific software tools that we could use, one of the most important challenges was to define the set of tools that could work together and, at the same time, being more suitable for implementing our three-level architecture.

  In a first prototype (Pernich et al., 2010) we had built a software platform with a set of 64 common services by using PostGIS,[22] GeoServer[23] and OpenLayers[24] for each level respectively. However, although these tools were suitable to work together and the first prototype worked fine, they presented several limitations with respect to our component-based architecture. The use of OpenLayers allowed us to create the set of designed services, but only in a static way. Everything was on the client-side and it was really difficult to separate processes from interfaces. Then, in a second stage, we changed the implementation of components as EJBs[25] and strictly followed a component-based development approach. GeoServer was then running as a software component (the *Map Server Service* component in Fig. 5) and the interfaces were implemented by using the Google Web Toolkit[26] (GWT) (the *Graphical Interface* component). The database was implemented in PostGis, and the whole system runs over a JBoss Application Server.[27] This new set of tools allowed us to implement the restrictions of our architecture making them more suitable for modifiability and scalability.

- *Validation*: There are several aspects to analyze within this activity. Firstly, some test cases must be defined in order to test the framework and the specification of the product line. Secondly, when a product is developed we must test this new instantiation. In addition, the creation of new products within the platform will allow us to verify the quality of the line by taking into account several aspects. According to the Software Engineering Institute (SEI) in its report named "A Framework for Software Product Line Practice, Version 5.0",[28] a software product line must be analyzed from three main perspectives: *reusable component development (or core assets)*, *product development*, and *management of the overall product line*. Then, in order to analyze and measure the line, we applied the two first perspectives to the development of two products. Each product belongs to a different organization within the Marine Ecology domain: the IBMPAS and the CENPAT-CONICET. The evaluation is based on some indicators proposed by SEI and tries to show a first analysis of the effectiveness of our product line.

  On the first perspective, *reusable component development*, we analyzed time required to develop the components and efficiency of the reusable components in order to reduce the amount of work needed for product development. From 64 services defined in our platform, we implemented 16 different components according to the design decisions described in the construction of our reference architecture (Fig. 5). From these 16 components, 5 of them contained a variability of no more of

---

[22] http://postgis.refractions.net/.

[23] http://geoserver.org/.

[24] http://openlayers.org/.

[25] http://www.jboss.org/ejb3.

[26] http://code.google.com/intl/es-AR/webtoolkit/overview.html.

[27] http://www.jboss.org/.

[28] http://www.sei.cmu.edu/productlines/frame_report/meas_tracking.htm.

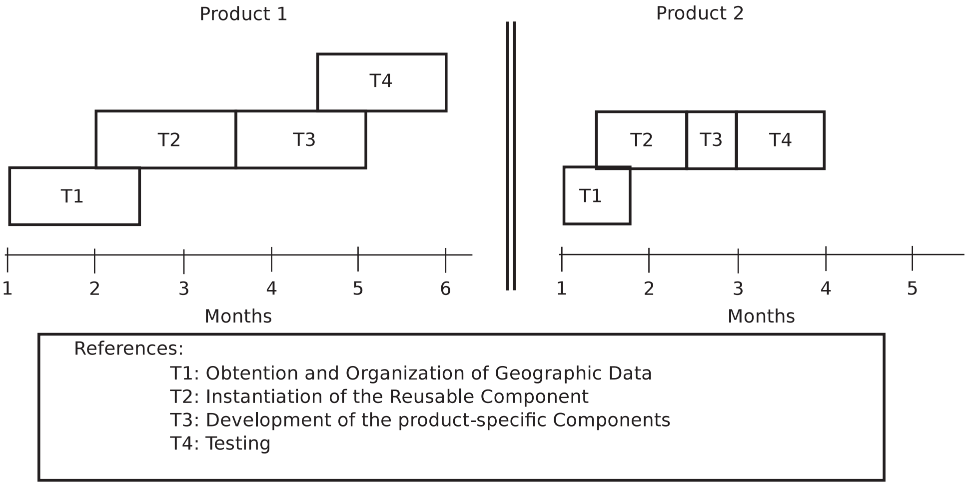


Fig. 7. Time required to perform the activities involved in the development of the two products.

three or four choices (such as the specification showed in Fig. 4). The addition of variability and the way of managing for instantiating future products were the more difficult tasks and demanded more time than expected. Thus, the total time required to implement our platform from the beginning of the project was 14 months, taking four months for the definition of the variability and the platform implementation. Fig. 6 shows the domain analysis and organizational analysis steps (Section 4.2) as well as time needed by each of them.

In this first work, we did not use a supporting tool to manage variability because selecting suitable tools is not easy and they have some disadvantages that we had to control.[29] However, as the variability was not really complex, we could manage well when we had to instantiate a new product. Next, considering the effectiveness of the reusable components, we should say that as we implemented only the components that are part of the line, all components were reused and instantiated when new products had to be implemented. Obviously, we have to be careful with this advantage because this can generate the implementation of many new product-specific components in order to satisfy the special requirements of a new product.

From the second perspective, *product development*, we evaluated three main aspects: time required for the construction of each product, percentage of bugs that were found in reusable and specific components, and time required in the construction of product-specific components. In order to do so, we divided the main steps involved in the development of the products according to the tasks denoted in Fig. 7.

We can see that time required for obtaining and organizing the geographic data was larger in Product 1. This happens because the IBMPAS organization had data only on papers and they were not digitalized. On the contrary, in the other organization (CENPAT-CONICET), data were well organized on spreadsheet files. Then, we analyzed the particular requirements of each organization in order to instantiate the reusable components with specific variability choices. As we aforementioned, all reusable components were used in both products and we did not have to add any other variability. However, time required for this task was shorter in Product 2 because the design engineers had gathered previous experience by working with these components during the development of Product 1. Following, the next task (T3) demanded the development of two product-specific components for Product 1 and only one for Product 2. For instance, in Product 2 we implemented a component that analyzes the advance of the *undaria pinnatífida* seaweed and its relation with the population of echinoderms. In Product 1, one of the specific components implements services for the analysis of influence of fishing on different aspects of species. Finally, in T4 we analyzed the testing activity taking into account the percentage of bugs found in reusable and specific components. Again, Product 1 took more time because it was the first product developed and we had to implement two specific components. Thus, in Product 2 we found fewer errors than in Product 1 (around 20% reduction).

In conclusion, by comparing times (Figs. 6 and 7), we can see that the development of Product 1 required only 42% of the time required by the SPL development, and it was a 28% for Product 2. Therefore, for these two products, saving is really interesting denoting also saving with respect to costs.[30]

5. Conclusion and future work

In this paper, we have described a software development oriented to improve the reuse of geographic services. To do so, firstly we analyzed the main characteristics of the geographic domain and the use of standards in order to face a software product line development. Then, we applied these aspects to the construction of the SPL and the instantiation of two products within a specific subdomain—the marine ecology subdomain. Based on the experiences of this work, we might say that:

- The nature of geographic information and the branches in which we can divide the discipline make the definition of a software product line directly over this domain a very complex task. Dividing the generic geographic domain into several domains and subdomains allowed us to decrease the initial complexity by considering the creation of several product lines according to the domain level in which we are working on. The definition of a domain-level approach allowed us to focus

---

[29] SEI—"A Framework for Software Product Line Practice, Version 5.0". Tool Support Section. http://www.sei.cmu.edu/productlines/frame_report/tool_support.htm.

[30] Product 1 is available at http://geoserver.ods.org/geoserver/www/webgissao/index.html.

on general aspects that must be applied to the domains and subdomains included in the generic geographic domain. At the same time, we could focus on specific aspects of the subdomains and add those aspects that are common for all of them.

- The use of the ISO 19119 std. as starting point to define the generic and specific services within the domain levels has generated two main benefits. Firstly, it allowed us to delimit the wide range of possible generic services by providing also a classification of them. And secondly, the taxonomy defined and specified within each domain level and within each particular product line, improved the capability of reuse among the products being built.
- There are several important aspects to be taken into account when geographic open source tools must be selected. Aspects as continuity in their developments, good documentations, complete and active forums, flexibility on the extension of provided services, and more, must be carefully analyzed. In addition, and the most important aspect in this domain, is that the set of tools should be compatible to work together and allow developers to implement the architecture defined in the standard, which follows a layered approach based on component reuse. In this work, we selected three main tools that fulfill all these requirements, but further work is needed to develop specific selection methods.
- In our two experiences of instantiating the SPL we could see important improvements on time and cost. This happened because all reusable components were applied and the variability was easy to manage. The rate of bugs was minimized when the second product was developed because all reusable components had already been tested (only the new variability and the new components had to be analyzed). In addition, the experience of the software developers in the use of the SPL also generated better time results.

Coming back to the main aspects of the species analyzed at both organizations (Section 4.1), we should remark that:

- The geographic area of the Gulfs constitutes one of the main South Atlantic areas for assisting governmental organizations to understand linkages among root causes of degradation and integrating needed changes in economic and wild-life preservation activities. Analyzing distribution changes, fishery, and organisms relationships serve to initiate capacity building and for bringing science into pragmatic uses. Reuse-based software development (using standard information) has allowed jointly undertaking strategic processes that would not been able to be addressed otherwise. Due to standardization of information and services, now a set of scientific analyses is treated as a building block (“the line”) allowing incremental construction of procedures. These common set provides a useful mechanism to foster participation at all levels scientific as well as operational and political (notice that one of the agencies is a joint endeavor between State and Academy). Now, everybody uses a common vocabulary facilitating understanding and addressing priorities. Faster software developing (“instantiating the line”) helped to draw diverse scenarios by incorporating variable settings, which have become the basis for decision-making in order to support the ecosystem preservation.
- From the strictly scientific side, our work is part of a Conservation Project of Bivalves of North Patagonic Gulfs. We carried out studies to analyze the response of exploited populations, where graphical data analyses are fundamental to understand the interaction among species, between species and fishery, and the response of the populations. These analyses involve the description of spatial/temporal patterns and processes of the dynamic of the native Ostrea Puelchana during the last 80 years, and include the estimation of a combination of parameters, such as demographic structure, growth, and recruitment. The nature of these processes is particularly adequate to treat them as pieces to be combined and reused. For instance, time series data of oyster distribution and density is currently extending. This is unusual and required analysis of combined factors (i.e. a set of common factors of the line) that were extended with assumptions (i.e. variable factors), such as the *closure of artisanal fishery in 1975*; *the presence of a motile stage the larvae condition*, etc. At the same time, as another (re)use of the line, parasitological studies were carried out simultaneously. There were also time series data with this kind of information, which required analysis of common factors as well as variable ones such as *parasite spread direction*. Results from both instantiations of the line contributed to analyze the response of the species. Although using traditional GIS for supporting the analysis was a possibility, the use of our product line facilitated building models and verifying assumptions faster. Besides, classification and separation of services helped to overcome the barrier of facing parallel analyses, showing that complexity was easier to handle.
- Of course, the line is not only about the oyster. All bivalve populations of the Gulfs will be included. At this point, it is interesting to notice that working with our approach stimulated “what-if” analyses. Participants, engaged with the development of the line, realized that the approach inherently invited them to scale the picture. In this sense, using the line became a motivating factor. However, we should also remark some pitfalls, such as the lack of specific training in GIS in some cases, which made us to organize specific sessions. Support from management of both organizations was the key point to achieve significant results.

As future work, we are working on a more formal validation of the product line for the marine ecology subdomain, through modeling quantitative quality indicators. At the same time, we are designing and implementing a supporting tool for dealing with domain-oriented variability, and including some intelligence for dealing with reuse patterns. Regarding extending the SPL, we are currently collaborating with organizations from the oceanographic domain, in such a way that the SPL be extended to another level. And of course, we continue reusing the line through the instantiation of several products required by IBMPAS and CENPAT-CONICET.

Acknowledgments

We would like to thank the two organizations that are collaborating in this project: IBMPAS and CENPAT-CONICET.

This work is partially supported by the UNComa project 04/E072 “Identificación, Evaluación y Uso de Composiciones Software” and by the PAE-PICT 2007-02312 “Métodos y Herramientas para Software Masivamente Distribuidos”.

References

Arango, G., 1994. Domain analysis. In: Marciniak, J. (Ed.), Encyclopedia of Software Engineering, vol. 1. Wiley, pp. 424–434.

Bonnett, A., 2008. What is Geography?. SAGE Publications Limited, London, Great Britain.

Bosch, J., 2000. Design and Use of Software Architectures: Adopting and Evolving a Product-Line Approach. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

Câmara, G., Souza, R.C.M., Pedrosa, B.M., Vinhas, L., Monteiro, A.M.V., Paiva, J.A., Carvalho, M.T., Gattass, M., 2000. TerraLib: technology in support of GIS innovation. In: II Brazilian Symposium on Geoinformatics, GeoInfo2000. São Paulo.

Chen, L., Babar, M.A., 2011. A systematic review of evaluation of variability management approaches in software product lines. Information and Software Technology 53 (4), 344–362.

Clements, P.C., Northrop, L., 2001. Software Product Lines: Practices and Patterns. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Czarnecki, K., Eisenecker, U.W., 2000. Generative Programming: Methods, Tools, and Applications. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA.

Czarnecki, K., Favre, J., Gogolla, M., Mens, T., 2005. Proceedings of the essentials of the Fourth UML/models workshop in software model engineering. In: MoDELS Satellite Events, pp. 151–158.

Czarnecki, K., Helsen, S., Eisenecker, U.W., 2005b. Formalizing cardinality-based feature models and their specialization. Software Process: Improvement and Practice 10 (1), 7–29.

Diviacco, P., 2005. An open source, web based, simple solution for seismic data dissemination and collaborative research. Computers & Geosciences 31 (5), 599–605.

Dobrica, L., Ionescu, T., Dobrica, L., Colesca, S.E., 2010. Solutions based on GIS technology in components of urban management applications. U.P.B. Scientific Bulletin 72 (2), 37–48.

Dobrica, L., Niemelä, E., 2008. A UML-based variability specification for product line architecture views. In: Proceedings of the Third International Conference on Software and Data Technologies, ICSOFT. Porto, Portugal, pp. 234–239.

FGDC-STD-001-1998, 1998. Content standard for digital geospatial metadata technical report csdgm version 2. Federal geographic data committee. 〈http://www.fgdc.gov/metadata/contstan.html〉.

Fils, D., Cervato, C., Reed, J., Diver, P., Tang, X., Bohling, G., Greer, D., 2009. Chronos architecture: experiences with an open-source services-oriented architecture for geoinformatics. Computers Geosciences 35 (4), 774–782.

FIPS55-DC3, 1998. Codes for named populated places, primary county divisions, and other locational entities of the United States, Puerto Rico, and the outlying areas. Federal information processing standards. 〈http://www.itl.nist.gov/fipspubs/fip55-3.htm〉.

Gomaa, H., 2004. Designing Software Product Lines with UML: From Use Cases to Pattern-Based Software Architectures. Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA.

Gordillo, S., Balaguer, F., Mostaccio, C., Das Neves, F., 1999. Developing GIS applications with objects: a design patterns approach. Geoinformatica 3 (March (1)), 7–32. URL 〈http://dx.doi.org/10.1023/A:1009809511770〉.

Holl, G., Grünbacher, P., Rabiser, R., 2012. A systematic review and an expert survey on capabilities supporting multi product lines. Information and Software Technology 54 (August (8)), 828–852. URL 〈http://dx.doi.org/10.1016/j.infsof.2012.02.002〉.

Kang, K., Cohen, S., Hess, J., Nowak, W., Peterson, S., 1990. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Software Engineering Institute, Carnegie Mellon University Pittsburgh, PA.

Kang, K., Kim, S., Lee, J., Kim, K., Kim, G.J., Shin, E., 1998. FORM: a feature-oriented reuse method with domain-specific reference architectures. Annals of Software Engineering 5, 143–168.

Lisboa Filho, J., Iochpe, C., Borges, K.A.V., 2002. Analysis patterns for GIS data schema reuse on urban management applications. CLEI Electronic Journal 5 (2).

Pernich, P., Buccella, A., Cechich, A., Doldan, S., Morsan, E., 2010. Reusing geographic e-services: a case study in the marine ecological domain. In: Cellary, W., Estevez, E. (Eds.), Software Services for e-World. IFIP Advances in Information and Communication Technology, vol. 341. Springer, Boston, pp. 193–204. URL 〈http://dx.doi.org/10.1007/978-3-642-16283-1-23〉.

Pohl, K., Böckle, G., van der Linden, F., 2005. Software Product Line Engineering: Foundations, Principles and Techniques. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Possompes, T., Dony, C., Huchard, M., Tibermacine, C., July 2011. Design of a UML profile for feature diagrams and its tooling implementation. In: Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE'2011). Miami Beach, Florida, USA.

Reinhartz-Berger, I., Tsoury, A., 2011. Specification and utilization of core assets: feature-oriented vs. UML-based methods. In: Proceedings of the 30th International Conference on Advances in Conceptual Modeling: Recent Developments and New Directions. ER'11. Springer-Verlag, Berlin, Heidelberg, pp. 302–311.

Rumpe, B., Robert, F., 2011. Variability in UML language and semantics. Software and Systems Modeling 10 (4), 439–440.

Schäfer, W., Prieto-Diaz, R., Matsumoto, M. (Eds.), 1993. Software Reusability. Ellis Horwood, New York, NY, USA.

Sodré, V.D.F., Lisboa Filo, J., Vilela, V.M., Vinícius, M., Andrade, A., 2005. Improving productivity and quality of GIS databases design using an analysis pattern catalog. In: Proceedings of the Second Asia-Pacific Conference on Conceptual Modelling, vol. 43. APCCM '05. Australian Computer Society, Inc., Darlinghurst, Australia, Australia, pp. 107–114.

van der Linden, F., Schmid, K., Rommes, E., 2007. Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Vranic, V., Snirc, J., 2006. Integrating feature modeling into UML. In: Proceedings of the NODe/GSEM'06, pp. 3–15.

Ziadi, T., Jézéquel, J., 2006. Software product line engineering with the UML: deriving products. In: Software Product Lines, pp. 557–588.