Web-based Geographic Information Systems SPLE: Domain Analysis and Experience Report

Alejandro Cortiñas
Universidade da Coruña
Database Lab
A Coruña, Spain
alejandro.cortinas@udc.es

Miguel R. Luaces
Oscar Pedreira
Ángeles S. Places
Universidade da Coruña
Database Lab
Enxenio
A Coruña, Spain
{luaces,opedreira,asplaces}@udc.es

Jennifer Pérez
Universidad Politécnica de Madrid
ETSISI - CITSEM
Madrid, Spain
jenifer.perez@etsisi.upm.es

ABSTRACT

Geographic Information Systems (GIS) play a critical role for supporting the development of Cyber Physical Systems (CPS), since they allow geolocating users and the "things" or smart objects that constitute a CPS, providing a realistic vision in quasi real-time. This has increased the demand of developing web-based GIS applications to be deployed in the different devices and wearables of the CPS with short time-to-market. This demand and the fact that web-based GIS applications of CPS share many features and known variability justifies why they present the perfect setting to apply software product-lines engineering (SPLE). In this paper, we present the experience of developing a web-based GIS product line in the SME Enxenio, and the methodology applied to define the product line. In addition, we present the results obtained providing the GIS community with a reference SPL that is ready for its evolution and enrichment.

CCS CONCEPTS

• **Software and its engineering → Software product lines**; • **Information systems** → *Geographic information systems*;

KEYWORDS

Software product line engineering, web-based geographic information systems, scaffolding

ACM Reference format:
Alejandro Cortiñas, Miguel R. Luaces, Oscar Pedreira, Ángeles S. Places, and Jennifer Pérez. 2017. Web-based Geographic Information Systems SPLE: Domain Analysis and Experience Report. In *Proceedings of SPLC '17, Sevilla, Spain, September 25-29, 2017,* 5 pages.
https://doi.org/10.1145/3106195.3106222

The work of the authors from the Universidade da Coruña has been funded by MINECO (PGE & FEDER) [TIN2016-78011-C4-1-R, TIN2016-77158-C4-3-R, TIN2015-69951-R, TIN2013-46238-C4-3-R, TIN2013-46801-C4-3-R]; CDTI and MINECO [Ref. IDI-20141259, Ref. ITC-20151305, Ref. ITC-20151247].

SPLC '17, September 25-29, 2017, Sevilla, Spain

ACM ISBN 978-1-4503-5221-5/17/09…$15.00
https://doi.org/10.1145/3106195.3106222

1 INTRODUCTION

Society is currently demanding services and applications that implement smart cities, buildings, energy grids or water networks, in such a way that they are more intelligent, efficient and comfortable. They are well-known examples of Cyber Physical Systems (CPS) [16]. Geographic Information Systems (GIS) appear as the suitable candidates to faithfully manage this information [7], and even more, web-based GIS are the most appropriate applications to be deployed in the different devices and wearables that constitute the CPS. Although the GIS of CPS share many features, other features vary widely depending on: (i) the kind of "things" or smart objects that the CPS monitors and controls; and (ii) the physical place in which the CPS is deployed. This commonality and variability present the perfect setting to apply Software Product Line Engineering (SPLE) [20] in the development of web-based GIS systems.

The SME (small and medium-sized enterprise) Enxenio[1] has been a leading provider of web-based GIS systems in the Galicia region and an important contributor at the Spanish national level for years. Enxenio has developed more than twenty GIS-based products and services for many industrial sectors (e.g., urban planning, land consolidation, expropriation management, logistics, mobile workforce management, or tourism promotion). Enxenio has already made some attempts to define a product line architecture for web-based GIS applications [4, 5]. However, their previous attempts have failed due to the lack of formalism in the definition of the architecture (e.g., there was no connection between the SPL features and the product architecture), caused in turn by not using a proper methodology for the definition of the SPL.

In this paper, we illustrate the experience of the SME Enxenio developing a product line for web-based GIS systems, in terms of the methodology that has been defined and followed, and the results obtained, in terms of the SPL constructed and its flexibility and reuse properties. This GIS SPL may serve as a SPL of reference for other GIS developers with the guarantee of acquiring a benefit on reuse and flexibility of software.

2 SPL METHODOLOGY

Defining a SPL within an IT company allows us to benefit from its expertise of the domain and also from the products of the domain

[1] http://www.enxenio.es/

that they have already developed. Identification of features, choosing the right technology or planning the evolution of the products are some of the processes that can use this kind of knowledge and resources. To address this construction in a formal way, we have defined a SPL methodology.

Our SPL methodology is based on two methodologies that are complementary one to each other: Magro et al. [17] and Nakagawa et al. [18]. Magro et al. [17] defines six steps to address the construction of a SPL and it has been put into practice for the construction of a SPL of Validation Systems. However, it does not consider specific issues of the Product Line Architecture (PLA) design such as taking into account reference or standard PLAs or prioritizing and selecting tasks at the architectural level. Since the methodology of Nakagawa et al. [18] is focused on the PLA design, we adapted the methodology of Magro et al. by decomposing the steps related with the PLA design into those suggested by Nakagawa et al. In addition, we also considered the work of Diaz et al. [9], since the traceability among the requirements and the PLA is a key issue for guaranteeing the SPL maintenance [1]. This issue was also considered by Iida et al. [12] for the construction of an automotive braking system SPL.

We have added some extra tasks to the resulting methodology in order to exploit the advantages discussed above, such as the related work analysis or taking into account both internal and external products in the Product Planning step (see Sect. 2).

The defined methodology can be seen in Fig. 1. The extensions to existing methodologies that are critical in the construction of the web-based GIS SPL are represented as white boxes. Domain Engineering stage is divided into three phases (see the top part Fig. 1). The first phase is the Requirements Analysis, which is composed of two steps, the Domain Analysis and the Product Planning. The Domain Analysis consists in analysing both the requirements and the related work (see step 1, Fig. 1). The related work analysis extends the work of Magro et al., which only focused on analysing the domain to determine the feasibility of constructing a SPL and extracting the requirements of the products to identify the commonalities and variabilities of the domain. Product Planning step considers all kinds of requirements associated to the different products that can result from the SPL deployment (see step 2, Fig. 1). But not only those that belong to the company (see step 1.2.1, Fig. 1), it is important to also consider other projects that are known or relevant in the domain (see step 1.2.2, Fig. 1). Once the phase is complete, from these complete analyses of requirements, a feature model must be constructed.

The second phase consists in the Architecture Design, which is divided into five steps (see Fig. 1). Step 1 is based on the methodology of Nakagawa et al. and it pursues to identify existing reference architectures or standards in the field in order to not start from scratch. We added step 2 because if a company has previously developed products of the SPL domain, it is interesting to analyze their architecture, since it may enrich the architecture of the SPL. Steps 3 and 4 are proposed by Nakagawa et al. In these steps the elements of the architecture are identified and selected, and the architectures is design, respectively. And finally, the methodology adds a new step 5 to analyze the technology requirements in order to determine needs and interoperability problems, even technology evolution needs. As a result of this second phase, a Product Line Architecture is obtained.

Finally, the third phase consists in mapping the features and architectural elements, in order to guarantee that all features of the SPL are supported by the PLA. In addition to this new first step of the evaluation phase. The common step of architecture evaluation proposed by the rest of methodologies was also considered in step 2. In these steps, it is important to check there are no inconsistencies or drawbacks between previous stages and the results obtained. If something is missing, this should be solved in previous stages and the process starts again from phases 1 and 2, checking again all steps (see feedback arrows, Fig. 1). Once this Domain Engineering stage is ready, it is possible to start with the deployment of products, that is, the stage Application Engineering and its specific stage of deriving products by selecting the variability feature that the product should provide.

3 DEFINITION OF THE SPL

This section illustrates how the methodology was applied to construct the SPL for the development of web-based GIS applications. A complete description of the results of each step of the methodology are available as supplementary material at the website of the Database Lab of the Universidade da Coruña[2].

3.1 Requirements Analysis

Since applications produced by the SPL will be used in Enxenio, it makes sense that experts of the company are the ones deciding the targeted product family and the requirements the platform must comply. Therefore, during the *domain analysis* step we conducted discussions and interviews with project managers from Enxenio with expertise on web-based GIS in order to extract the requirements of the platform (see the step 1.1.1,Fig. 1). We classified these requirements in four different groups: R1) Data Management, which includes every requirement related to how the data, both alphanumeric and geometrical, is stored, introduced in the system and internally processed; R2) Graphical User Interface requirements, related to the way the standard alphanumeric data is accessed through the web interface, and which data is provided to the final user; R3) Map Viewer requirements, how the geographic data must be shown in the web application; and R4) User Management requirements, since there are some required functionalities regarding authentication of users and management of their roles, as in any other web application. A table summarizing the requirements can be found as supplementary material[3]. From these requirements we derived the set of features that the SPL platform must provide in order to generate the desired products. The features were identified and described, as well as associated with the requirement from which they are derived, thus keeping the traceability. The full table is available as supplementary material[4]. At this point, the features are just listed but we still do not have the information regarding to whether they are mandatory or optional. This information is obtained in a further step by analysing existing products.

[2] http://lbd.udc.es/
[3] Web-based GIS requirements: http://lbd.udc.es/webgis-spl/requirements.pdf
[4] Web-based GIS feature list: http://lbd.udc.es/webgis-spl/featurelist.pdf

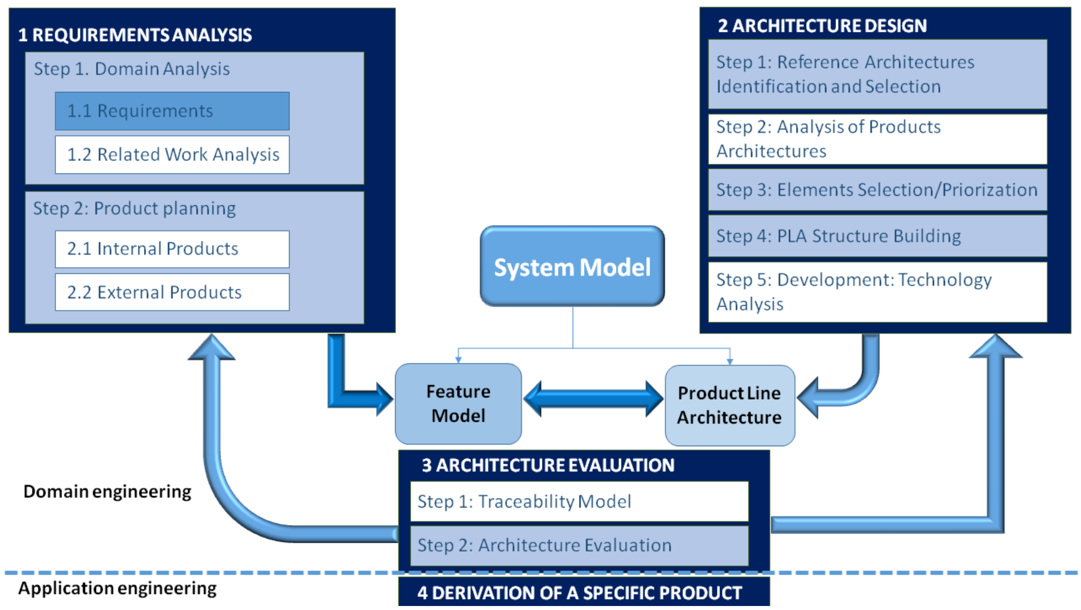

Figure 1: Methodology

Related work about SPLE applied on GIS is very scarce (see the step 1.1.2, Fig. 1). In [6], a SPL for GIS in the marine ecology is presented, but there is a non-exhaustive list of features. Therefore, this work was not used as reference for our generic GIS products. In [8], a tool for the automatic generation of web-based GIS from a data model is shown. However, in this case the design of the data model is the only variability of the generated applications, having all of them the same features.

During the *product planning* step we contrasted the set of features identified and described in the domain analysis step with existing products (see the step 1.2.1, Fig. 1). Specifically, we used the three products developed by Enxenio which are representative examples of applications that our SPL should be able to produce. Therefore, the set of features identified and the ones existing in these products must match: (i) webEIEL[5], a web-based GIS developed for the Provincial Council of A Coruña (Spain); (ii) Galician Cultural Heritage[6], a web-based GIS to promote cultural and touristic heritage; and (iii) Via Maps[7], a web-based GIS to promote cultural heritage focused on mobile devices in low-bandwidth environments such as limited public Wi-Fi networks. These three products cover a wide range of user needs in many different concerns. For example, the user expertise expected from the users varies from experts (webEIEL) to casual users (Galician Cultural Heritage and Via Maps); the tools required to customize the map appearance and use the map vary from as much as possible (webEIEL), to some of them (Galician Cultural Heritage), to almost none (Via Maps); the expected bandwidth of the user ranges from a broadband connection (webEIEL) to a limited bandwidth one (Via Maps); and finally, the device used to browse the map ranges from desktop computers (webEIEL) to mobile devices (Via Maps).

[5]WebEIEL: http://webeiel.dicoruna.es/gl/
[6]Galician Cultural Heritage: http://www.patrimonioculturalgalego.org/ViaxeVirtual
[7]Via Maps: https://madrid.via1101.pv.enxenio.net/Servicios/cartovia/

As a second iteration we also contrasted our decisions against three well-known external products (see the step 1.2.2, Fig. 1): namely the tool ESRI ArcGIS Online[8] because it is the leading product on commercial GIS development tools, Google Maps[9] because it is the reference on web-based map browsing and querying application, and OpenStreetMap[10] because it is the collaborative counterpart to Google Maps.

Apart from validating the coherence of our identified features, with this step we are also able to set the priority of each feature, and to decide which ones are common to every product, therefore mandatory, and which appear only in some of them, being then optional features. A table summarizing the result of the process is available as supplementary material[11]. To calculate the priority of the features we have used the number of times that each one appears in the products. If the feature appears in any of the products, the priority is the sum of the number of appearances in the products, 5 being the maximum priority that a feature can achieve. In case the feature is not included in any of the analysed applications, we manually set the priority between 0 and 1 to determine how important a feature is for the SPL of the web-based GIS of Enxenio.

Finally, the feature model of our SPL is provided as supplementary material[12]. It can be seen that features appearing in every product are mandatory, features appearing in some of the applications are optional and the ones not appearing at all have been removed from the feature model. We also decided the types of aggregation (between XOR and OR) depending on the appearance of the features.

[8]http://www.esri.com/software/arcgis/arcgisonline
[9]http://maps.google.com
[10]https://www.openstreetmap.org/
[11]Web-based GIS feature planning: http://lbd.udc.es/webgis-spl/featureplan.pdf
[12]Web-based GIS feature model: http://lbd.udc.es/webgis-spl/featuremodel.png

3.2 Architecture Design

There have been many attempts to define a reference architecture for web-based geographic information systems. ESRI, the leading company of the sector, has been proposing system architectures that include their products for decades. A constant aspect of these architectures is that they are layer-based (3-tiered or n-tiered) [10]. However, in addition to depending heavily on ESRI products, the architecture does not provide enough detail for the components. Another major driving force has been the European Commission through the INSPIRE directive [13] that establishes an infrastructure for spatial information in the European Community. Even though INSPIRE provides much detail on specific services, and it defines a service bus-based system architecture [14], it does not describe the way in which the services are expected to interact in the architecture and it does not provide any detail on the services that are out of the scope of the directive (e.g., the functionality of the web client).

The stakeholders that have defined a large collection of standards for GIS that are currently followed by most software libraries are the Open Geospatial Consortium (OGC) and the International Organization for Standardization (ISO). Particularly, the OpenGIS Service Architecture Version 4.3 [19], which is the same document as ISO19119:2005, defines a geographic services architecture identifying architecture patterns for service interfaces and the relationships among them, and providing guidelines for the selection and specification of geographic services from platform-neutral and platform-specific perspectives. ISO19119:2005 is revised in ISO19119:2016, but without any changes that would affect our SPL work. In the *Reference Architectures Identification and Selection* step we selected ISO19119:2005 as the reference architecture given the high level of detail provided. The set of selected services from [19] in the Elements selection/priorization step is available as supplementary material at the website of the research group[13].

After a deep analysis of the architectures of the products described in Sect. 3.1, carried out in the Analysis of products architectures step, we have identified that the reference architecture must be enriched with the following characteristics: i) the products must work both on desktop browsers and mobile browsers, therefore, the client must be developed using responsive web design and a JavaScript web framework to ensure cross-browser support; ii) some clients require that the product conforms to the INSPIRE directive and/or international standards, thus, the components of the architecture must follow these standards as closely as possible.

The final PLA resulting from the *PLA Structure Building* step of the SPL can be seen as supplementary material[14]. We have maintained the same logical architecture from [19], and we have classified every new service within this architecture, showing also the relations among services. It provides all the services selected from the reference architecture, but also two other services added afterwards (namely *Geoportal* and *Access control*). The new services are motivated by inconsistencies discovered in the stage of Architecture Evaluation, as mentioned in Sect. 3.3.

Regarding the *Development: technologies analysis* step, we have four different types of technologies and tools in function of the part of the architecture where they are used: *DBMS*, *Map Server*, *Data Server* and *Web Client*. The selection of the technologies for the DBMS and the Map Server is already in the requirements of our SPL, so it belongs to the selection of features from the Feature Model for each product. Regarding the Data Server, the SME is expert in Java technologies, which are also used in most of the applications developed by the company. Therefore, we use Spring as the most known alternative for Java Data Server and its set of libraries (such as Spring MVC, Spring Security, etc.). In the part of the Web Client, current web applications use the SPA (Single Page Application) pattern to simulate the feeling of a desktop application. We also apply this pattern, and to do so we use a de facto standard library: AngularJS. For the web map viewer, most current GIS with web map viewers use OpenLayers or Leaflet as map viewer library. We choose Leaflet as it is more lightweight and one of our characteristics is to build products that work both on desktop and mobile devices.

3.3 Architecture Evaluation

Once we have defined both the Feature Model and the PLA, the next stage in our methodology is the evaluation of the whole SPL platform. We have already presented the traceability model between Requirements and Features, and now we complete it by linking features to components from the functional architecture. The resulting table is available as supplementary material[15].

Although we started from a reference architecture, in the first iteration of this step we found that many of the features of the SPL platform were not provided for any of the services of our PLA structure, such as "Access control". This reflects the importance of following a serious methodology and even more, the importance of the step of validation within it. When we found the inconsistencies described, we started a new iteration of the process, solving these problems in the first stages of the methodology with the feedback provided in this step by adding the required components (see Sect. 2). We considered the end of this iterative process when we do not found inconsistencies between the features and the set of components providing them.

3.4 Derivation of a Specific Product

The derivation process is the actual generation of a product from a subset of features. Our derivation process follows an annotation-based approach for the product generation. Using this approach the variability within the code is represented with annotations that the derivation engine can interpret [2]. Although annotation-based approaches in SPL derivation have always been criticized [11, 21], other approaches (e.g., aspect-oriented or compositive, which are the most popular) cannot handle the vast collection of languages used in a standard web application (e.g., Java, Javascript with its variants, HTML, CSS, etc.), and they cannot guarantee the evolution of the platform (for example, in case a new language appears). In fact, many works have appeared lately analysing the usage of annotative approaches in industry and trying to improve their usage by solving their deficiencies [3, 15].

4 LESSONS LEARNT AND DISCUSSION

In order to give preliminary statistics about the benefits of the SPL, we have analysed the source code of the three products developed by

[13] Web-based GIS service selection: http://lbd.udc.es/webgis-spl/serviceselection.pdf
[14] Web-based GIS PLA structure: http://lbd.udc.es/webgis-spl/plastructure.pdf
[15] Web-based GIS feature - services: http://lbd.udc.es/webgis-spl/featureservices.pdf

Enxenio and used during our process (Sect. 3.1). Since they took part on our methodological process, the features in their products are supported by our platform and derivating these products is feasible. We measured the percentage of source code files and source code lines that can be reused among the different products due to the fact they are common assets of the SPL.

The percentage of reused files represents the files, from the total, that can be derived directly from the specification of the product without having to manually code them. Since the projects have a predominant Java nature, we decided to calculate individually this percentage for Java classes and for the rest of file types. The average of reusability for Java classes is 87%, achieving 94% and 93% in WebEIEL and Via Maps, respectively. Regarding the rest of the files, their percentage of reusability descends until 53%. This means that, on average, half of the Javascript, HTML, CSS or other files are generated directly by the SPL.

The second measure is equivalent but providing data about the lines of code (LOC) instead of the number of files. In this case, the average of lines that are derived from the SPL is 56%. The product with more LOC, which is Via Maps with 67 348, reuses a 44% of them, which is the minimum percentage of the three products. Nevertheless, it represents 29 806 LOC that do not have to be implemented manually thanks to the SPL.

Another point to remark is that the three products under analysis are totally different, and our SPL gives support to create of all of them. The main consequence of this is the reduction of time-to-market. Even if a developer team has to modify the part of the product not reused, they can start their work on an existing project instead of having to set it up from the beginning. Besides, having a single common SPL platform for all our web-based GIS facilitates the evolution of these products, safeguarding the quality and reducing the number of errors. To finish up, having such a platform it is possible to update our outdated products, like the ones analysed, to use state of the art technologies and architectures with minimum costs for the SME.

5 CONCLUSIONS

In this work we have shared the experience of applying SPLE in the domain of web-based GIS by a SME called Enxenio. We have presented a new methodology that joins several existing ones and extends them in an effort to complement their advantages. We have dealt with every step of the methodology to illustrate our experience in applying the methodology. We also have made an initial estimation of the benefits of deploying a web-based GIS SPL in the context of an SME. Besides, we have reviewed current state of art in GIS applications and architectures to design our own architecture, describing every feature of importance for current GIS. As a result, this work takes a firm step forward in the GIS domain, since these requirements, features and PLA may serve as reference models for the development of web-based GIS and the SPL defined can be extended and adopted by other companies.

As future work, we plan to automate the SPL, and specifically, the derivation process and the data domain provisioning. Once this automation is completely developed, the empirical validation of the SPL in terms of costs, code generation, quality and reduced time-to-market will be possible.

REFERENCES

[1] S.A. Ajila and A.B. Kaba. Using traceability mechanisms to support software product line evolution. In *Proceedings of the 2004 IEEE International Conference on Information Reuse and Integration, 2004. IRI 2004.* IEEE, 157–162. https://doi.org/10.1109/IRI.2004.1431453

[2] Sven Apel, Don Batory, Christian Kästner, and Gunter Saake. 2013. *Feature-Oriented Software Product Lines.* http://www.springer.com/us/book/9783642375200

[3] Thorsten Berger, Ralf Rublack, Divya Nair, Joanne M. Atlee, Martin Becker, Krzysztof Czarnecki, and Andrzej Wąsowski. 2013. A survey of variability modeling in industrial practice. *Proceedings of the Seventh International Workshop on Variability Modelling of Software-intensive Systems - VaMoS '13* (2013), 1. https://doi.org/10.1145/2430502.2430513

[4] Nieves R Brisaboa, Alejandro Cortiñas, Miguel R Luaces, and Oscar Pedreira. 2016. GISBuilder: a framework for the semi-automatic generation of web-based geographic information systems. *Proceedings of the 20th Pacific Asia Conference on Information Systems (PACIS 2016)* (2016).

[5] Nieves R Brisaboa, Alejandro Cortiñas, Miguel R Luaces, and Matías Pol'la. 2015. A Reusable Software Architecture for Geographic Information Systems based on Software Product Line Engineering. In *Proceedings of the 5th International Conference on Model & Data Engineering (MEDI 2015)*, Vol. 9344. Springer, 320–331. https://doi.org/10.1007/978-3-319-23781-7_26

[6] Agustina Buccella, Alejandra Cechich, Matías Pol'la, Maximiliano Arias, Maria Del Socorro Doldan, Enrique Morsan, Maria del Socorro Doldan, and Enrique Morsan. 2014. Marine ecology service reuse through taxonomy-oriented SPL development. *Computers and Geosciences* 73 (2014), 108–121. https://doi.org/10.1016/j.cageo.2014.09.004

[7] Peter A. Burrough and Rachael A. McDonnell. 1999. Principles of Geographical Information Systems. (1999), 422 pages. https://doi.org/10.2307/144481

[8] Paule-Annick Devoine, Bogdan Moisuc, and Jerome Gensel. 2012. GENGHIS: an Environment for the Generation of Spatiotemporal Visualization Interfaces. In *Innovative Software Development in GIS.* 121–150.

[9] Jessica Díaz, Jennifer Pérez, and Juan Garbajosa. 2014. A model for tracing variability from features to product-line architectures: A case study in smart grids. *Requirements Engineering* 20, 3 (2014), 323–343. https://doi.org/10.1007/s00766-014-0203-1

[10] ESRI. 2017. Architecting the ArcGIS Platform: Best Practices. (2017). https://www.esri.com/{~}/media/Files/Pdfs/products/ar

[11] J.-M. Favre. 1997. Understanding-in-the-large. *Proceedings Fifth International Workshop on Program Comprehension. IWPC'97* May (1997). https://doi.org/10.1109/WPC.1997.601260

[12] Takahiro Iida, Masahiro Matsubara, Kentaro Yoshimura, Hideyuki Kojima, and Kimio Nishino. 2016. PLE for automotive braking system with management of impacts from equipment interactions. *Proceedings of the 20th International Systems and Software Product Line Conference on - SPLC '16* (2016), 232–241. https://doi.org/10.1145/2934466.2934490

[13] INSPIRE. Directive 2007/2/EC of the European Parliament and of the Council of 14 March 2007 establishing an Infrastructure for Spatial Information in the European Community. (????).

[14] INSPIRE Network Services Drafting Team. 2008. INSPIRE Network Services Architecture. (2008), 29 pages. http://inspire.ec.europa.eu/reports/ImplementingRules/network/D3

[15] Christian Kästner, Sven Apel, and Martin Kuhlemann. 2008. Granularity in software product lines. In *Proceedings of the 30th international conference on Software engineering - ICSE '08.* ACM Press, New York, New York, USA, 311. https://doi.org/10.1145/1368088.1368131

[16] Edward A Lee. 2008. Cyber physical systems: Design challenges. *Object Oriented Real-Time Distributed Computing (ISORC), 11th IEEE International Symposium on* (2008), 363–369. https://doi.org/10.1109/ISORC.2008.25

[17] Belén Magro, Juan Garbajosa, and Jennifer Pérez. 2009. The Development of A Software Product Line for Validation Environments. In *Applied Software Product Line Engineering.* Taylor and Francis, 173–200.

[18] Elisa Yumi Nakagawa, Martin Becker, and Jose Carlos Maldonado. 2013. Towards a Process to Design Product Line Architectures Based on Reference Architectures. *Proceedings of the 17th International Software Product Line Conference* (2013), 157–161. https://doi.org/10.1145/2491627.2491651

[19] George Percivall. 2002. OpenGIS Service Architecture. *Architecture* 02 (2002), 78. file:///C:/Users/Anzabandora/Downloads/02-112

[20] Klaus Pohl, Günter Böckle, and Frank Van Der Linden. 2005. *Software Product Line Engineering: foundations, principles and techniques.* Vol. 49. 467 pages. https://doi.org/10.1007/3-540-28901-1

[21] Henry Spencer and Zoology Computer. 1992. # ifdef Considered Harmful , or Portability Experience With C News. *Usenix* (1992), 185–198.