Discovering Geographic Services from Textual Use Cases

Viviana E. Saldaño
Software Engineering Research Project
Unidad Académica Caleta Olivia – Universidad Nacional de la Patagonia Austral
vivianas@uaco.unpa.edu.ar

Agustina Buccella, Alejandra Cechich
Grupo de Investigación en Ingeniería de Software del Comahue (GIISCo)
Departamento de Ciencias de la Computación
Universidad Nacional del Comahue
{abuccel, acechich}@uncoma.edu.ar

ABSTRACT

Component Based Software Development (CBSD) relies upon utilization of previously developed software components OTS (Off-The-Shelf), which are appropriately merged to satisfy particular system requirements. However, wide acceptance of this paradigm at industry requires efficient component identification and selection, aspects which are being investigated until now.

In this context, this paper further explores the use of a geographic services taxonomy, which facilitates component identification, and is used by analysts in charge of developing a Geographic Information System (GIS) employing a CBSD approach.

In this article, different knowledge extraction techniques are evaluated and a methodology is proposed to standardize resultant vocabulary in order to allow automatic tools to support GIS services search.

Keywords: DSBC, OTS, GIS services, taxonomies, textual use cases.

1. INTRODUCTION

Component Based Software Development (CBSD) employs prefabricated pieces, developed at different times by different people and possibly with distinct goals of use [24][26]. In this context, processes for searching and selecting OTS components [4] are quite important. However, they have got serious limitations, such as dealing with documentation not expressive enough to guarantee an effective selection of the components; and not counting with mediator processes which allow speeding up the search of components that offer the required services.

In this context, a client who requires a specific component's service may interrogate a mediator service for the references to those components which supply the required service category. Besides, with the rise of component based software development, a number of GIS software companies have begun marketing distinct software components oriented to GIS software developers needs.

To achieve a more efficient development, analysts concentrate on reusability and interoperability attributes. However, it takes a lot of time and effort to find those components that meet intended functionality. Key to accomplish this task is counting with components' standard information, which allows speeding up software composition search. In this way, services supply could be standardized so that compositions are stored in an easy access repository. The same should happen for services demand, which should also be expressed in standard terms to make search easier. All this topics determine the final success of a selection process.

This article is an extension of the article presented in [22], in which a geographic service taxonomy has been created to make components identification easier. Furthermore, this article is presented in contrast to the supply model presented in [8][9] where a publication service is defined, to facilitate selection of requested components. In this work, a methodology is defined for standardizing vocabulary used by analysts in charge of developing GIS applications, to help in specifying requirements of components which provide the corresponding GIS services.

This article is organized in the following way. Firstly, we briefly introduce the whole view of our approach. Next in Section 3, related work is described. Then, in Section 4 we briefly describe the geographic services taxonomy and we define the proposed method to find required services, starting from a textual use case specification. In Section 5 an application of the proposed method to an actual use case is shown. Finally, in the last two sections we discuss identified lessons and we draw some conclusions and future work.

2. OUR APPROACH IN A NUTSHELL

Selecting OTS (Off-The-Shelf) components involves a complex process that relates component developers and application developers. The former are responsible for supplying information to be used when searching, understanding and selecting components. For instance, as shown in Figure 1 from [14], component developers' activities constitute the "Publication Process", which consist of (1) classifying the component recently created; (2) documenting the component; and (3) storing information in a repository. On the other hand, the application developers' activities constitute the "Selection Process", which consist of (1) searching candidates by matching some quality criteria, including functionality; (2) understanding information of candidates; and (3) making decisions about selection and adaptation.

One of the problems of both processes is the lack of standard documentation to describe OTS components. Among the different documentation proposals, there is a common understanding about the needs of defining a conceptual framework to classify and describe components from a repository or marketplace. However, there is a lack of such similar understanding to the way components should be characterized.

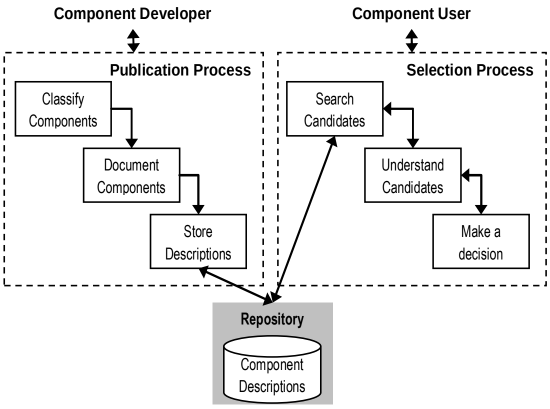

Figure 1. A component selection process

In [4], we have identified some key elements to support a standardized framework towards a knowledge-based process for COTS component identification. Firstly, when establishing a component marketplace, one of the specific demands is to provide well-structured information about components, i.e. a well-structured catalogue. This issue leads us to questioning about how information should be structured to be considered useful. It means not only gathering information from third parties but also setting basic elements that categorize COTS components, allowing us to assess quality properties – perhaps by using metrics or some testing mechanisms. On the other hand, matching provided and required services requires not only standardizing information from vendors but also standardizing requirements for searching. In analyzing different trends on component classification and matching, we have found several interesting aspects, which might constitute a basis for improving COTS component identification. Particularly, the use of domain-specific standard information might set a common vocabulary to support both processes – publication & selection.

With this aim, in [8][22] we have adapted a general component specification framework [19] to build a more suitable scheme for classifying GIS components. In order to normalize the classification categories, we firstly analyzed available information on web catalogues for GIS components, and we tailored the geographic service taxonomy provided by the ISO/IEC 19119 standard [16].

Our main concern is about how to build a useful description repository to automate selection. Therefore, from the suppliers' view, we suggest building a wrapper for information available on the Web, in such a way that search engines may access a normalized information structure when selecting candidates. From the composers' view, we suggest building another wrapper for components' requirements, in such a way that required services are expressed by using the same normalized information structure. In this way, selection becomes a matter of mapping two models represented by a wrapper for information on the web and a wrapper for component's requirements.

The main focus of this paper is on the description of this last wrapper: requirements from use cases are encapsulated so identification and selection of component candidates are treated uniformly.

3. RELATED WORK

3.1 Use Cases

During last twenty years, use cases are being used to describe the main functionalities of systems and their relations with the environment. In particular, the Unified Modeling Language (UML) [18], in which one specific use case diagram is proposed, became an accepted standard for the analysis and design of systems. In UML, use cases are identified and structured by using diagrams and textual descriptions. However, in the literature there exist several different proposals to specify use cases. The main difference among them is related to the formalism level used to perform the specification. For instance, there are approaches proposing formal notations [12][23][27] that provide mechanisms to inspect and analyze use case specifications automatically. However, in general these notations are not understandable enough for non-technical users. In this way, other approaches [2][5][13] have emerged to solve this problem proposing the use of a restricted natural language. The most popular proposal within this line is presented by Cockburn [5], in which templates are applied to specify the behavior within use cases. In addition, in the work presented in [10], author proposes the use of a controlled natural language structuring sentences in a particular way. Here, the SVDPI (Subject, Verb, Direct object, Preposition, Indirect object) pattern is applied as follows: "Sentence structure must be simple"… "Subject… verb … direct object … preposition … indirect object".

In [11], a metamodel for textual use case descriptions is presented. The metamodel allows

developers to specify the behavior of use cases in a flow-oriented way. It defines a textual representation easily understandable for common users. The notation is formal enough allowing automatic check consistency tasks between the model and the textual descriptions.
Another approach is proposed by Bittner and Spence [3] in which some level of formality into use case descriptions is allowed without restricting the use of natural language. Here, use cases are also described in a flow-oriented way defining the behavior through event sequences.
In our work we combine two of the aforementioned approaches [5][10] in order to maximize the understanding of use cases for common users and to provide, at the same time, a notation in which the automatic analysis and validation are possible. In particular, we use the templates proposed by Cockburn [5] together with the SVDPI pattern proposed by Graham [10].
In a study that we performed to software development organizations in our city, we did not obtain good results. In general, the organizations do not use standardized techniques to specify requirements. They use only the natural language to describe the functionality of systems. Thus, our work starts from these textual descriptions and build the use cases by using our semi-formal notation based on the techniques aforementioned [5][10].

3.2 Knowledge Extraction in Use Cases

With respect to knowledge extraction, several works can be found in the literature. For instance, in [6][15] a method for processing textual requirements is proposed. The method identifies main attributes of actions described into each step of a use case. To do so, it uses linguistic tools to build parse trees. Then, once actions have been identified, the method allows knowing the operations accepted and required by an entity. Thus, the definition of component interfaces or required services is possible.
In [21] authors present a method composed of a set of guides for addressing the construction of use case specifications. The method creates a use case specification containing an unambiguous natural language description. To do so, the method progressively transforms initial and partial natural language descriptions of scenarios into well-structured use case specifications. The basis of the method is a set of linguistic patterns and structures.
In [20], a formal model, named Generic UC View, is introduced to allow developers to reason about knowledge. Based on behavior protocols, pro-cases can be checked for compliance via an already existing verifier. As pro-cases' syntax is simple, resembling regular-expressions, there are simple guidelines for transforming a use case written in classical textual form (based on a template) into a pro-case. Pro-cases is a formal technique which allows specifying behavior and which is also designed for high readability.
In our work we use the method proposed in [6][15] because it provides an easy application without requiring any extra tool. Also, the proposal provides linguistic tools that are available on the Web with many on-line examples of their uses. In addition, the method was modified to solve specific requirements of GIS services. A detailed description of this method is provided in the next section.

4. GIS SERVICES IDENTIFICATION

In this section we firstly present a brief description of our taxonomy defined in [22]. The taxonomy has been created by using services extracted from the ISO 19119 standard [17] and from other classifications of geographic operations [1][25] . The taxonomy is used to classify the services required by the users as well as Non-Technical Requirements that are relevant to the selection of OTS components.
Next, our methodology is presented. Textual use cases are analyzed in order to extract the main required GIS services. In this way, once the services are identified the taxonomy can be instantiated to be used to find the correct GIS components that provide these services.

4.1 GIS Services Taxonomy

In order to create the GIS service taxonomy, we analyzed geographic services categories, defined in ISO 19119 [16], and examples of real geographic services. In addition, a Non Technical Requirements category was added to the taxonomy because of its importance on the selection of OTS components.
Part of this taxonomy can be seen in Table 1. It shows four columns: level, service, main verbs, and representative objects. The two first columns are extracted from the taxonomy of the ISO 19119 standard, and represent the main geographic services that can be used in a service specification. The two last columns denote a list of key words in which sentences and verbs are defined separately. For instance, at the human-interaction level, the *catalogue-viewer service* can be identified by the verbs locate, explore, and manage; and by metadata about geographic data or geographic services.

4.2 Methodology for Extracting Information from Use Case Specifications

Our methodology is based on the proposals defined in [6][15] with some modifications allowing the extraction of useful information to classify the use cases according to our taxonomy. As we aforementioned, the methodology applies linguistic tools to build a parse tree in which actions of predefined textual use cases are identified. Then these actions are used to discover the required GIS services.

Table 1. Fragment of modified geographic services taxonomy

Layer	Service	Service Description	
		Main Verb	Representative Object
Human Interaction	Catalogue Viewer	Locate Explore Manage	Geographic data metadata Geographic service metadata
	Service Editor	Control Understand Compose Invoke Plan	Service Chain of services
	Geographic Feature Editor	Interact Display Query Add	Feature Characteristic Orientation Perspective Transparency Texture
	...		

Particularly, we use the template for use cases defined in [5] in which the functionalities of the system are described. In addition, a controlled language [13] is used in order to restrict the use of words and sentence structures. Thus, each use case step must fulfill the following premises:

Premise 1: *A step of a textual use case describes either (a) communication between an actor and system, or (b) an internal action.*

Premise 2: *The action is described by a simple English sentence following the SVDPI pattern ("Subject…Verb… Direct objet …Preposition… Indirect objet")*

Figure 2 shows the main steps of the methodology. The four steps (A-D) of our methodology are performed by each action defined in the main scenario of a use case specification. Let us briefly illustrate each of these steps.

A) Determining the POS

Inputs of the methodology are textual use cases documented by using the template. Then, the first step, determining the POS (part-of-speech), starts analyzing each sentence of the main scenario of the use case. This step analyzes each word and specifies the type (verb, noun, etc.) and the role of each of them within the sentence in which they are defined.

B) Generating the Parse Tree

The second step creates different parse trees according to the sentences of the main scenario of the use cases. In each parse tree, nodes represent the phrases and leafs represent the words of each sentence. In order to perform the steps A and B, we use the FreeLing[1] tool suite that provides support for the Spanish language.

C) Generating Event Tokens

In the third step, event tokens are created by finding main verbs and representative objects within each sentence of the parse tree.

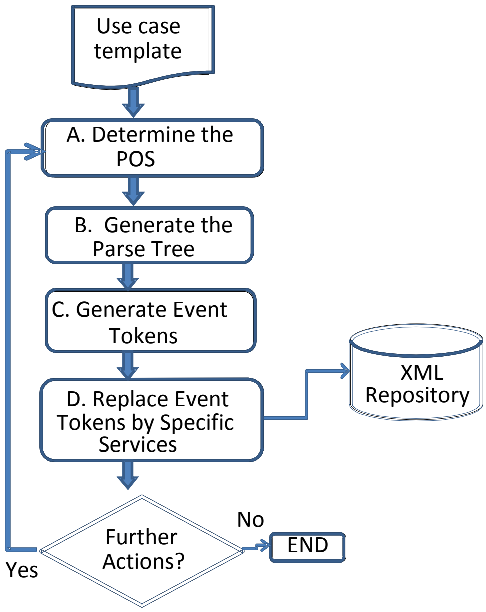

Figure 2. Steps to extract GIS services from textual use cases

D) Replacing Event Tokens by Specific Services

The last step performs the mapping between the event tokens created in step C) and the services defined in our taxonomy (Table 1). To give support to this last step, we create a semi-automatic tool that uses EuroWordNet[2] as thesaurus for the Spanish language. It is a user-guided tool that assists users in the process of choosing synonym relations to make suitable mappings. The results of the mapped services are stored in an XML repository that will be used to find mappings between the user's requirements and the information of OTS components published on the Web.

5. A CASE STUDY

In this section we present a case study in order to show how our methodology works. The specification was provided by a local organization of Comodoro Rivadavia in Argentina. It was translated to our notation of use cases by applying the template and the specific vocabulary.

1 http://garraf.epsevg.upc.es/freeling/
2 http://www.illc.uva.nl/EuroWordNet/

Table 2 shows the resultant use case in which a service to modify a coordinate of an electric line is presented.

Table 2. Textual Use Case Example

Use Case Name	Modify location coordinates of an electric line	
Service	Update information of electric lines	
Description	Assign new geographic coordinate	
Actors	System Administrator	
Scope	Line layer, representing electric lines	
Precondition	Electric line layer has been created in the system	
Main Scenario	**1**	The user selects an electric line
	2	The user changes coordinate attribute
	3	System displays electric line with location updated
Postcondition	System has updated location of electric line	

According to our methodology, we apply the four steps (Figure 2) to each action defined in the main scenario of the use case specification.

Steps A) and B) are performed together by the application of Freeling tools suite [7]. Considering the first action in the main scenario of the use case ("The user selects an electric line"), the tool creates a parse tree classifying each word of the sentence. Figure 3 shows this tree.

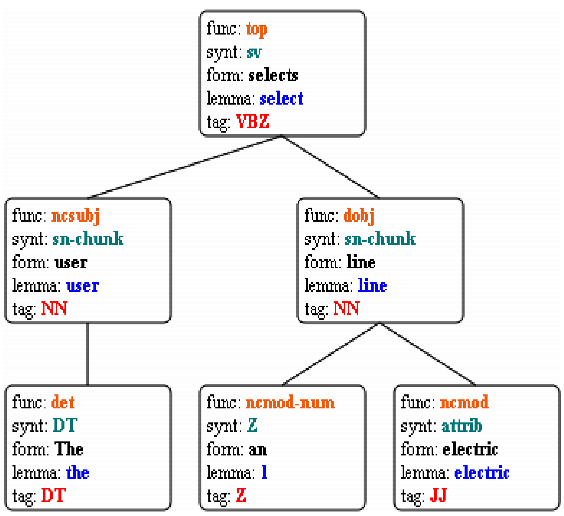

Figure 3. Parse tree for: "The user selects an electric line"

Next, in step C) the method generates the Event Token. Following the example, in the sentence "The user selects an electric line", the main verb is the principal word (root) of the verbal phrase. In this case, the verb "select" is classified as the main verb. The representative object is created from the nouns in the direct object of the sentence; and it must be subordinated to the main verb. In our example "line" is the representative object. Therefore, the event token will be "select line".

Finally, step D) performs the mapping between the event token created in step C) and the services defined in our taxonomy. In our example, the event token (select electric line) is compared to the service descriptions (columns 3 and 4 of Table 1) of the taxonomy. Once the user selects the description that matches a required service, he/she will obtain the standardized name of it and the level in which it is contained. The user, by using our tool, finds that the verb "*select*" matches the verb "*interact*" and the object "*line*" is similar to "*feature*". Thus, the tool proposes the standard service *"Geographic feature editor"* of the *Human Interaction Category*. The results of the mapped services, as shown in Figure 4, are stored in an XML repository that will be used to find mappings between the user's requirements and the information of OTS components published on the Web.

```
<service>
        <object> feature</object>
        <verb>interact</verb>
        <category>human interaction</category>
</service>
```

Figure 4. XML requirement representation

6. DISCUSSION

We started to validate our approach considering real web portals of component catalogues (i.e., ComponentSource[3], FreeGIS[4], etc.), and requirements from a case study in the domain of oil companies. Based on the results of preliminary applications, where we mapped the models generated by the two wrappers, we identified the following lessons:

The use of the standard ISO 19119 in both models allowed us a better mapping between offered and required services of GIS. This is a clear advantage that came from a better understanding among all parties. We had to carefully evaluate how much of the information required to assess OTS components was actually available from information in the catalogues. We analyzed the current gap between the required and provided information, so refinement of taxonomies was guided to reduce the gap, yielding in more realistic attributes. After all these efforts, and providing guidelines for using the tools, we realized that detecting and selecting candidates was faster – and produce higher stakeholders' satisfaction.

3 http://www.componentsource.com/
4 http://freegis.org/

The use of textual use cases allows us to apply Natural Language Processing techniques, which help extract requirements mirroring the standard. However, use cases differ widely in breadth and scope, and its appropriate selection is not straightforward. We emphasize the use of scenarios appropriate to all roles involving a system. The architect role is one widely considered but we also have roles for the system composer, the reuse architect, and others, depending on the domain. It is important when analyzing a system that all roles relevant to that system be considered since design decisions may be made to accommodate any of the roles. The process of choosing use cases for analysis forces designers to consider the future uses of, and changes to, the system. It also forces to consider other notations (such as use case diagrams) that should be properly adapted to fit our approach.

7. CONCLUSIONS AND FUTURE WORK

We are still working on making the process of searching geographic components easier. So, as UML is a widespread used tool for software modeling, we analyzed information provided by use case specifications and the ways of using this information to find those services which satisfy required functionality in these use cases.
A methodology for use case knowledge extraction has been proposed and tailored to the particular problem of geographical services search. Besides, it was also necessary to adapt the Geographic Services Taxonomy previously defined.
As future work, we will go on working on the automation of the proposed methodology for its validation. After that, this methodology will be combined with the methodology defined for publishing GIS services in order to make a semi-automatic mapping between supply and demand.

REFERENCES

[1] Albrecht, J. Universal GIS Operations for Environmental Modeling, 1996.
[2] Armour, F., Miller, G. Advanced Use Case Modeling Volume One, Software Systems. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2001.
[3] Bittner, K., Spence, I. Use Case Modeling. Addison-Wesley, 2000.
[4] Cechich A., Réquilé A., Aguirre J., Luzuriaga J. Trends on COTS Component Identification. 5th International Conference on COTS-Based Software Systems. Orlando, USA. IEEE Computer Science Press. 2006.
[5] Cockburn, A. Writing Effective Use Cases, Addison-Wesley Pub Co, ISBN: 0201702258, 1ra ed., Enero 2000.
[6] Drazan, J., Mencl, V. Improved Processing of Textual Use Cases: Deriving Behavior Specifications. In Proceedings of SOFSEM 2007. LNCS, vol. 4362. pp. 856-886. Harrachov, Czech Republic, Enero 20-26, 2007.
[7] Freeling Home Page, http://garraf.epsevg.upc.es/freeling/
[8] Gaetan, G., Cechich, A. Buccella, A. Un Esquema de Clasificación Facetado para Publicación de Catálogos de Componentes SIG. In Proceedings of CACIC 2008. Chilecito, La Rioja, Argentina, Octubre 2008. ISBN 987-24611-0-2
[9] Gaetan, G., Cechich, A. Buccella, A. Aplicación de Técnicas de Procesamiento de Lenguaje Natural y Web Semántica en la Publicación de Componentes para SIG. In Proceedings of ASSE 2009. Mar del Plata, Argentina, Agosto de 2009.
[10] Graham, I.: Object-Oriented Methods: Principeles and Practice, Addison-Wesley, 2000.
[11] Hoffmann, V., Lichter, H., NyBen, A., Walter, A. Towards the Integration of UML-and textual Use Case Modeling. Journal of Object Tecnology, Vol. 8, No. 3, 2009.
[12] Kholkar, D., Krishna, G., Shrotri, U., and Venkatesh, R. Visual Specification and Analysis of Use Cases. In SoftVis '05: Proceedings of the 2005 ACM symposium on Software visualization, pp 77-85, New York, NY, USA, ACM, 2005.
[13] Kulak, D., Guiney, E. Use Cases:Requirements in Context. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, ISBN: 0321154983, 2da ed., Agosto 2003.
[14] Lucena, V. F. Flexible Web-based Management of Components for Industrial Automation. Phd thesis, Stuttgart University. (2002)
[15] Mencl, V. Deriving Behavior Specifications from Textual Use Cases In: Proceedings of Workshop on Intelligent Technologies for Software Engineering (WITSE04, part of ASE 2004), Linz, Austria, Oesterreichische Computer Gesellschaft, Sept. 2004.
[16] Percivall, G. ISO 19119 and OGC Service Architecture. FIG XXII International Congress, Washington, D.C. USA, 2002.
[17] OGC. Topic 12: OpenGIS Service Architecture. Open GIS Consortium, 2002
[18] OMG. UML Superstructure Specification, v2.1.2. OMG Formal Document 2007-11-02, Noviembre 2007.
[19] Overhage S. UnSCom: A Standardized Framework for the Specification of Software Components. In Proceedings of the 5th Annual International Conference on Object-Oriented and Internet-Based Technologies, Concepts, and Applications for a Networked World (NODe 2004). LNCS 3263, 169-184, (2004)
[20] Plasil, F., Mencl, V. Getting "Whole Picture" Behavior in a Use Case Model. Transactions of SDPS: Journal of Integrated Design and Process Science 7(4), pag.63-79, Dic. 2003.
[21] Rolland, C, Achour, C. Guiding the construction of textual use case specifications. In Data & Knowledge Engineering, vol. 25 no. 1-2, pág. 125-160, Marzo 1998.
[22] Saldaño, V., Buccella, A., Cechich, A. Una Taxonomía de Servicios Geográficos para facilitar la identificación de componentes. In Proceedings of CACIC 2008. Chilecito, La Rioja, Argentina, Octubre 2008. ISBN 987-24611-0-2

[23] Spivey, J. The Z Notation: A Reference Manual. Prentice Hall, 1992.
[24] Szyperski, C. Component Software-Beyond Object-Oriented Programming. Addison-Wesley, 1998.
[25] Timpf, S. Geographic Task Models for Geographic Information Processing. Meeting on Fundamental Questions in Geographic Information Science, Manchester, UK. 2001
[26] Wallnau, K., Hissam, S., and Seacord, R. Building Systems from Commercial Components. Addison-Wesley, 2002.
[27] Whittle, J and Jayaraman, P. Generating Hierarchical State Machines from Use Case Charts. In RE '06: Proceedings of the 14th IEEE International Requirements Engineering Conference (RE'06), pp 16-25. Washington, DC, USA, IEEE Computer Society, 2006.