A Machine-Processable Dublin Core Application Profile for Analysis Patterns to Provide Linked Data

Lucas Francisco da Matta Vegi
Federal University of Viçosa, Brazil
lucasvegi@gmail.com

Jugurta Lisboa-Filho
Federal University of Viçosa, Brazil
jugurta@ufv.br

Joep Crompvoets
Katholieke Universiteit Leuven, Leuven, Belgium
joep.crompvoets@soc.kuleuven.be

Abstract

Analysis patterns are reusable computational artifacts aimed at the analysis stage of the software development process. Although the analysis patterns can facilitate the work of analysts and developers, the access to them is still very poor because of the way they are usually described and made available. The Analysis Patterns Reuse Infrastructure (APRI) was proposed in order to reduce these deficiencies for supporting, cataloging and encouraging the reuse of analysis patterns. This infrastructure comprises a repository of analysis patterns documented through a specific metadata profile and that can be accessed via Web services. Based on the proposal of APRI, this article presents the specific metadata profile to the documentation of analysis patterns called Dublin Core Application Profile for Analysis Patterns (DC2AP). This application profile is described by RDF files and identified via URI, thus providing Linked Data that increase the potential for reuse of the analysis patterns.

Keywords: analysis patterns; reuse; metadata standards; Dublin Core; Sematic Web; Linked Data.

1. Introduction

Analysis patterns are reusable computational artifacts aimed at the analysis stage of the software development process. Ideas applied during the requirements analysis and conceptual modeling of specific domain software, can be expressed abstractly through analysis patterns. From the application of analysis patterns, other analysts can reuse these ideas during the software project of any domain. According to Fowler (1997), analysis patterns are ideas proven to be useful in a given context that may be applicable for other practical contexts. Thus, these reusable computational artifacts can make analysis stage faster and more accurate for analysts and developers (Fernandez and Yuan, 2000) avoiding rework to develop and test solutions that already exist.

Although the analysis patterns can facilitate the work of analysts and programmers by adding value through reuse of proven useful and tested ideas, the access to them is still very poor (Blaimer et al., 2010). So far there is no template to specify the analysis patterns that is widely accepted, making each set of analysis patterns being specified according to the preferences of its authors. In addition to not having a pattern specification, the analysis patterns are normally provided in scientific books and papers that are means of access that are not machine-processable. These restrictions do not allow analysis patterns to be effectively retrieved by a search software (Blaimer et al., 2010), thus burdening the time taken to retrieve an analysis pattern and limiting the quality of the results obtained by searches. It happens because the understanding of the descriptions of these analysis patterns is restricted to the human ability of comprehension.

In order to minimize these problems of specification and enhance retrieval of analysis patterns and therefore increase the potential for reuse of them, Vegi et al. (2012) proposed the architecture of an Analysis Patterns Reuse Infrastructure (APRI). This infrastructure, which was inspired by the components of Spatial Data Infrastructures (SDI) proposed by Béjar et al. (2009), consists of a repository of analysis patterns, documented in a specific metadata profile and that can be accessed via Web services.

This article presents a Dublin Core application profile created specifically to describe analysis patterns in a manner consistent with the proposal of APRI (Vegi et al., 2012). The semantic base of the elements of this application profile is provided by the template proposed by Pantoquilho et al. (2003) and Raminhos et al. (2006) to describe the analysis patterns. The application profile presented by this work is machine-processable, enabling analysis patterns to be described and published as Linked Data (Bizer et al., 2009) through files in the Resource Description Framework (RDF) format (Klyne & Carroll, 2004).

The remainder of this paper is organized as follows: Section 2 describes related work to documentation of analysis patterns, Semantic Web and Linked Data. The machine-processable metadata profile proposed by this article is presented in Section 3, while Section 4 presents some conclusions and future work.

2. Related Work

2.1. Documentation and Organization of Analysis Patterns

Documentation of analysis patterns is an important way for contextualizing the reuse scope of a pattern and for enabling the sharing of knowledge among designers. However, this documentation is performed in a heterogeneous manner among the authors, since there is no standardized way to specify analysis patterns (Blaimer et al., 2010). There are many approaches to specify analysis patterns ranging from non-formalized textual descriptions to formalized descriptions based on templates.

Some analysis patterns specified in a non-formalized textual manner can be found in Fowler (1997) and Hay (1995). This little formal way of describing an analysis pattern affects reuse because it makes it harder for designers to quickly understand the contextual scope of patterns, and, mostly, it limits the retrieval of analysis patterns through computerized search engines. Thus important detailed information for designers may not be described or even retrieved, thus limiting the spread of these patterns and thereby their potential for reuse.

Analysis patterns have also been described through the use of templates, which are structures with predetermined topics similar to those used to describe design patterns (Gamma et al., 1994). Usually a template is composed of essential topics such as context, problem, motivation and solution (Fowler, 1997), combined with other specific topics defined by their authors.

Some analysis patterns documented through templates can be found in the work of Lisboa-Filho et al. (2002) and Fernandez and Yuan (2009). Meszaros and Doble (1997) present in their work a template composed of topics: name, problem, context, motivation, solution, participants and related patterns. Lisboa-Filho et al. (2002) applied this template as shown in Figure 1. Pantoquilho et al. (2003) and Raminhos et al. (2006) propose a detailed template developed specifically for the documentation of analysis patterns. This template combines common topics used previously by several authors, with new topics aimed at describing the analysis patterns more broadly.

Besides adequate documentation, another important factor to increase the potential for reuse of analysis patterns is the way it is organized and therefore made available. This is because before a pattern is applied to a project, the designer needs to know of its existence and then select it (Blaimer et al., 2010). Usually analysis patterns are organized together forming collections. These collections can have different formats, such as books, articles and websites, and they can still be classified as pattern languages and pattern catalogues (Blaimer et al., 2010). The pattern languages are basically collections of analysis patterns aimed at solving a specific problem. In a pattern language, the patterns are related to each other and must follow application rules—for example, the order in which they must be applied to solve the problem in question (Blaimer et al., 2010). The pattern catalogues are collections of analysis patterns not necessarily related, but organized based on criteria in common and searching ability. Fowler (1997) presents an example of an analysis pattern catalog. The organization of the patterns described in his book is made from groups of patterns that have an application domain in common and may be found by potential users through a table of contents (Blaimer et al., 2010).

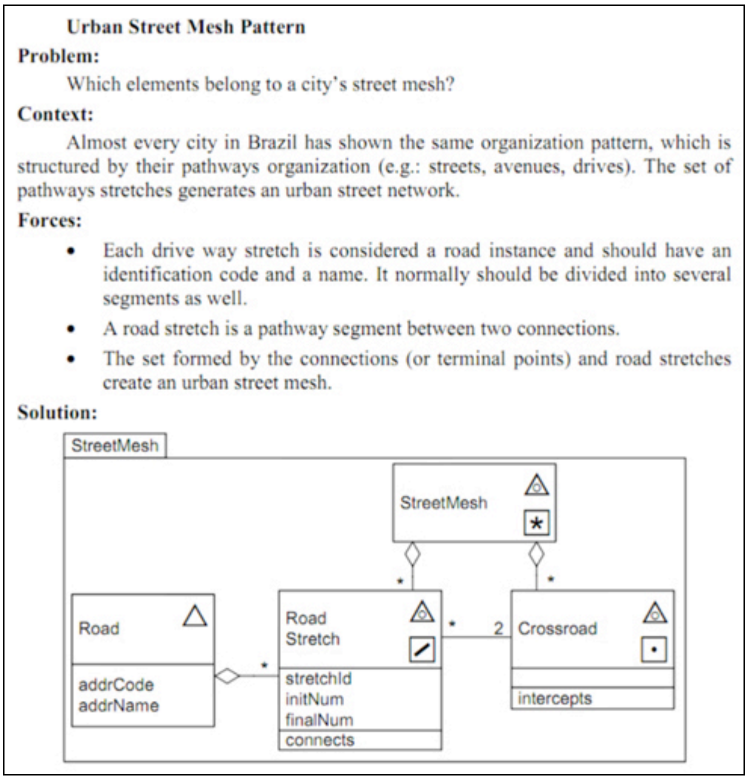

FIG. 1. Urban Street Mesh Pattern (Source: Lisboa-Filho et al., 2002)

2.2. Analysis Patterns Reuse Infrastructure

Spatial Data Infrastructure (SDI) is a relevant base collection of technologies, policies and institutional arrangements that facilitate the availability of, and access to, spatial data (Nebert, 2004). Currently, most SDIs are based on Service-Oriented Architecture (SOA), supporting creation of shared environments that are distributed and interoperable based on web Services (Davis Júnior & Alves, 2005).

In the field of Geographic Information Systems (GIS), SDIs have been used very efficiently as a tool for reusing services and geospatial data. In a way analogous to how geospatial data are documented and retrieved in an SDI, Vegi et al. (2012) proposed in their work an architecture for Analysis Patterns Reuse Infrastructure (APRI). In the proposed infrastructure, analysis patterns should be documented using metadata and can be retrieved through web services. Figure 2 presents the infrastructure architecture proposed by Vegi et al. (2012) facilitating the dissemination and increase of reuse of analysis patterns.

The APRI (Figure 2) consists of the following components:

- *Pattern Portal:* contains a set of websites focused on obtaining the analysis patterns, and tools and services that provide the discovery, cataloging and reuse of them.
- *Metadata Repository:* repositories that contain metadata in XML (eXtensible Markup Language) for the specification of analysis patterns and services contained in an APRI. The machine-processable Dublin Core Application Profile proposed later in this paper should be used to specify analysis patterns contained in these metadata repositories.

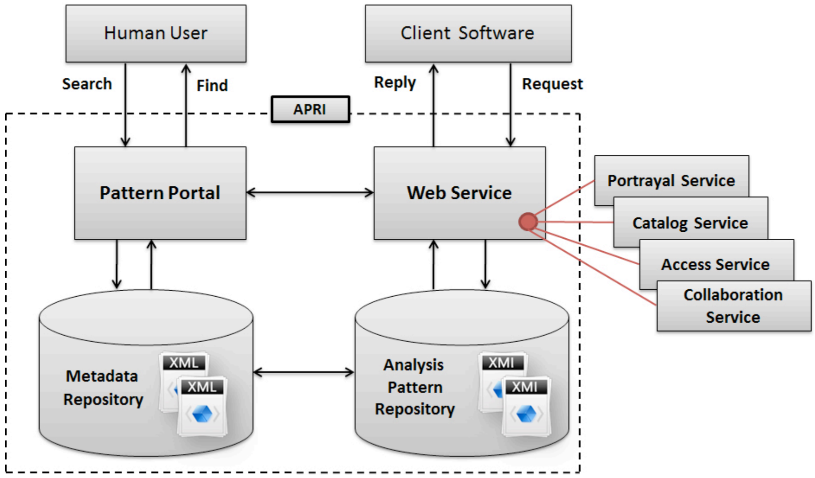

FIG. 2. Analysis Patterns Reuse Infrastructure (APRI) Architecture (Source: Vegi et al., 2012)

- *Analysis Pattern Repository:* repositories that contain the diagrams that represent the solutions proposed by analysis patterns in the XMI format (XML Metadata Interchange), allowing their use for visualization and collaboration services.
- *Portrayal Service:* services that support the visualization of diagrams that represents the solutions proposed by analysis patterns contained in an APRI.
- *Catalog Service:* services that enable the discovery and use of analysis patterns and services of an APRI, based on their metadata.
- *Access Service:* services that allow access to and download of the analysis patterns.
- *Collaboration Service:* services that allow designers to share use experiences to improve the analysis patterns.

2.3. Semantic Web and Linked Data

The Semantic Web is seen as a layer of the Web where it is possible to publish, obtain and use data that can be processed directly or indirectly by machines (Berners-Lee, 2000). The desire to extend the capabilities of conventional Web consisting of hypertext documents for a global environment, where machine-processable structured data are widely published and used, is not new. Berners-Lee et al. (1994) predicted in their work that in the course of time the Web in which the objects were predominantly documents interpretable by humans would evolve into an environment with more machine-processable semantic information.

While the Semantic Web aims to transform the conventional Web into an environment composed of machine-processable data, Linked Data provides the means to make this possible. Linked Data are machine-processable structured data published on the Web. These data are well defined and are linked to other data from different sources through semantic relationships (Bizer et al., 2009). The main difference between conventional hypertext Web and Linked Data is the types of relationships established between the data. While in conventional Web the nature of the relationship between two documents is implicit, in Linked Data that nature is explicit and understandable by computer (Bizer et al., 2009).

Linked Data uses three main technologies to support the Semantic Web. Uniform Resource Identifiers (URIs) (Berners-Lee et al., 2005) are used to identify the data. The HTTP protocol on the other hand is used as a mechanism for data retrieval and RDF is used to describe the machine-processable data (Keizer et al., 2011).

RDF is a framework of machine-processable metadata used to describe both Web documents and real-world concepts in a network such as people and companies (Sauermann et al., 2008). From the use of RDF to describe the data, it becomes possible to establish semantic relationships between them due to the data is encoded. Each relationship between RDF data consists of the triple formed by the concepts of subject, predicate and object. In this triple, subject and object are URI identifiers of the related data and the predicate specifies the semantics of the relationship, i.e., the kind of relationship between the data (Klyne & Carroll, 2004).

The main objective of Linked Data is to build the so-called Web of data, thus allowing the Web to be used as a single global database, thereby reducing barriers to the reuse, integration and implementation of data from distributed heterogeneous sources (Bizer et al., 2009).

The Web of data created from the publication of Linked Data on the Web enables the creation of new types of applications classified into three main categories: (1) Linked Data browsers, (2) Linked Data search engines, and (3) applications of a specific domain. Linked Data browsers allow users to navigate between data from different sources through semantic relationships established between them, and then to browse through a particular source of data and from that to go on to the entire network by relationships among the data. The Linked Data search engines support performance of complex searches similar to those offered by database management systems, allow data to be retrieved from Web of data through searches that consider the nature of relationships between them (Bizer et al., 2009).

Linked Data must be published on the Web accompanied by metadata to enable potential users, humans or machines, to assess their quality and reliability (Hartig, 2009). This metadata can be provided by using metadata standards, e.g., Dublin Core (DCMI, 1998).

3. A Machine-Processable Dublin Core Application Profile for Analysis Patterns

The Dublin Core Application Profile for Analysis Patterns (DC2AP) was developed based on the template to specify analysis patterns as proposed by Pantoquilho et al. (2003). The main objectives of DC2AP are to improve the retrieval and reuse of analysis patterns by means of a description that allows a more precise treatment by a computer, and thus provide detailed information about the analysis patterns that were not retrieved by search engines.

3.1. Mapping of Dublin Core Metadata Elements to Pantoquilho et al.'s Template

In contrast to the Dublin Core metadata standard, which is generic and therefore aimed to document resources of several domains, the template proposed by Pantoquilho et al. (2003) is designed specifically for the documentation of analysis patterns, so it is rich in specific details of this specific domain. Due to such level of details, this template was chosen as a basis for the creation of DC2AP.

The first step in creating the DC2AP was the realization of a mapping between the elements proposed by the Dublin Core and the elements of the template as proposed by Pantoquilho et al. (2003). With this mapping, the elements of both structures were compared and classified based on their semantic intersections, allowing a subsequent fusion between them. Table 1 shows the result of mapping done between the Dublin Core standard and Pantoquilho et al.'s template.

Although Table 1 shows only the mapping between the elements of Simple Dublin Core and the Pantoquilho et al.'s template, the elements contained in Qualified Dublin Core were also considered in this comparative process, but none of them had direct equivalents in the template.

Some elements of Table 1 present the comment "partly." These elements are not semantically identical to the other in which they have been mapped, but are just partially equivalent. Although this is not the ideal scenario, all the elements were mapped taking into account the highest possible equivalence among them.

Several mappings between elements of the Dublin Core and elements of other structures have already been performed and made available in the literature. An example of such mapping is presented in U.S. Library of Congress (2008).

TABLE 1: Mapping Dublin Core to Pantoquilho et al.'s analysis pattern template

Simple Dublin Core element	Pantoquilho et al.'s Template element
Title	1. Name 2. Also Known As
Creator	3. History (partly)
Subject	7. Context
Description	5. Problem 6. Motivation 7. Context 8. Applicability 14. Examples (partly) 18. Known Uses (partly)
Publisher	No direct equivalent
Contributor	3. History (partly)
Date	3. History (partly)
Type	No direct equivalent
Format	No direct equivalent
Indentifier	1. Name (partly)
Source	15. Related Patterns (partly)
Language	No direct equivalent
Relation	13. Anti-Patterns Trap (partly) 15. Related Patterns (partly) 16. Design Patterns (partly)
Coverage	No direct equivalent
Rights	No direct equivalent
No direct equivalent	4. Structural adjustments 9. Requirements 9.1. Functional requirements 9.2. Non-functional requirements 9.3. Dependencies and contributions 9.4. Conflict identification & guidance to resolution 9.5. Priorities 9.6. Participants 10. Modelling 10.1. Structure 10.1.1. Class diagram 10.1.2. Class description 10.2. Behaviour 10.2.1. Collaboration or sequence diagrams: 10.2.2. Activity diagrams 10.2.3. State diagrams 10.3. Solution Variants 11. Resulting context 12. Consequences 17. Design guidelines

3.2. Addition of New Metadata Elements and Creation of Application Rules

From the mapping described above, all equivalent elements have been identified and combined, thereby allowing that the Pantoquilho et al.'s template was fused to Dublin Core, giving rise to the basic structure of the DC2AP.

Most elements of the Pantoquilho et al.'s template that had direct equivalent mapping became element refinements of the others from the Dublin Core. This happened because the Dublin Core elements are generic, and therefore require specializations to compose an application profile for a specific domain. These necessary specializations were made by the elements of the chosen template (Pantoquilho et al., 2003).

During the fusion process of the structures in question, two elements from Pantoquilho et al.'s template were discarded. The element "Applicability" was discarded because its semantics are very similar to the element "Problem" and therefore was considered redundant. The element "Structural adjustments" was also discarded as unnecessary in the context of an application profile of metadata. In this context there are many well-defined application rules not necessary to document structural adjustments performed during use of the profile.

Following the fusion of structures, some elements have undergone semantic adjustments and new ones were proposed to complete the set of elements that comprise DC2AP. Table 2 presents all the elements that make up the profile proposed by this work.

The use of application profiles is generally controlled by rules. This work proposed rules on the obligatoriness, occurrence and value type of each DC2AP element:

- *Obligatoriness:* [M] Mandatory; [O] Optional; [Cd] Conditional.
- *Occurrence:* [S] Single; [Mu] Multiple.
- *Value Type:* [St]String; [D] Date; [U] URI; [N] Null; [UNS] URI; Number and String; [US] URI and String.

TABLE 2: DC2AP elements and application rules

DC2AP Element and their Application Rules			New
1. Identifier [M] [S] [UNS]			
2. Title [M] [S][St]	2.1. Alternative Title [O] [Mu] [St]		
3. Creator [M] [Mu] [St]			
4. Subject [M] [Mu] [St]			
5. Description [M] [S] [N]	5.1. Problem [M] [S] [St]		
	5.2. Motivation [M] [Mu] [St]	5.2.1. Example [M] [Mu] [St]	
		5.2.2. Known Uses ** [O] [Mu] [St]	
	5.3. Context [M] [S] [St]		
6. Publisher [O] [Mu] [St]			
7. Contributor [Cd] [Mu] [St]			
8. Date [M] [S] [N]	8.1. Created [M] [S] [D]		
	8.2. Modified [Cd] [S] [D]		
9. Type [M] [S] [US]	9.1. Notation [M] [S] [St]		Yes
10. Format [M] [Mu] [US]			
11. Source [Cd] [S] [UNS]			
12. Language [M] [S] [US]			
13. Relation [Cd] [S] [N]	13.1. Is Version of [Cd] [S] [UNS]		
	13.2. Is Replaced by * [Cd] [Mu] [UNS]		
	13.3. Replaces * [Cd] [Mu] [UNS]		
	13.4. Is Part of [O] [Mu] [UNS]		
	13.5. Has Part [O] [Mu] [UNS]		
	13.6. Is Designed with ** [O] [Mu] [UNS]		Yes
	13.7. Should Avoid ** [O] [Mu] [UNS]		Yes
	13.8. Complemented by ** [O] [Mu] [UNS]		Yes
	13.9. About [Cd] [S] [St]		
14. Coverage [O] [Mu] [St]			
15. Rights [Cd] [Mu] [US]			
16. History * [M] [Mu] [N]	16.1. Event Date [M] [S] [D]		Yes
	16.2. Author [M] [Mu] [St]		Yes
	16.3. Reason [M] [S] [St]		Yes
	16.4. Changes [Cd] [S] [St]		Yes
17. Requirements [M] [S] [N]	17.1. Functional Requirements [M] [Mu] [St]		
	17.2. Non-functional Requirements [O] [Mu] [St]		
	17.3. Dependencies and Contributions [M] [S] [St]	17.3.1. Dependency Graph [M] [S] [U]	Yes
		17.3.2. Contribution Graph [Cd] [S] [U]	Yes
	17.4. Conflict identification and Guidance to Resolution [Cd] [Mu] [St]		
	17.5. Priorities Diagram [M] [S] [U]		
	17.6. Participants [M] [Mu] [St]		
18. Modelling [M] [S] [N]	18.1. Behaviour [M] [S] [N]	18.1.1. Use Case Diagram [M] [S] [U]	Yes
		18.1.2. Collaboration/Sequence Diagrams [M] [Mu] [U]	
		18.1.3. Activity/State Diagrams [O] [Mu] [U]	Yes
	18.2. Structure [M] [S] [N]	18.2.1. Class Diagram [M] [S] [U]	
		18.2.2. Class Descriptions [M] [S] U]	
		18.2.3. Relationship Descriptions [M] [Mu] [St]	Yes
	18.3. Solution Variants ** [O] [Mu] [U]		
19. Resulting Context ** [O] [Mu] [St]			
20. Design Guidelines ** [O] Mu] [St]			
21. Consequences [M] [S] N]	21.1. Positive [M] [Mu] [St]		Yes
	21.2. Negative [M] [Mu] [St]		Yes

* Version Control element
** Experiences Collaboration element

Due to space limitations, the semantic description of each of the elements that comprise DC2AP, as well as some details of the rules for applying them are not presented in this paper. However, this information can be obtained at: `http://purl.org/dc2ap/TechnicalDescription`.

As shown in Table 2, DC2AP contains some elements for version control of documented patterns and others for the sharing of experiences of use. These features were incorporated into this profile to allow the creation of dynamic collections of analysis patterns, where new improved versions of the patterns may be proposed from the collaboration of experience of using them. Moreover, all the versions of the analysis patterns may be related to each other, thereby providing the creation of a repository of analysis patterns rich in details. These resources allow potential users to retrieve the version that best fits their needs more efficiently. All these characteristics are consistent with the proposal of the APRI (Vegi et al., 2012).

3.3. Providing Linked Data with RDF semantic properties

In order to transform the metadata profile previously presented in a machine-processable profile and therefore suit it to the proposal of Linked Data, each of the elements proposed by DC2AP were assigned URI identifiers and described by RDF semantic properties. According to Coyle and Baker (2009), it is important before creating new RDF semantic properties to look for existing properties that are semantically equivalent to the elements proposed by an application profile, because the use of known properties associated with elements of a profile increases the metadata semantic interoperability. So, searches were initially performed for RDF vocabularies that contains semantic properties equivalent to elements of DC2AP. Table 3 presents the compatible RDF vocabularies selected after the searches.

TABLE 3: RDF compatible vocabularies

Vocabulary Title	Vocabulary URI
The Dublin Core Metadata Element Set, v1.1	`http://purl.org/dc/elements/1.1/`
Dublin Core Terms	`http://purl.org/dc/terms/`
RDA Group 1 Elements	`http://rdvocab.info/Elements/`
RDA Roles	`http://rdvocab.info/roles/`

Although four RDF vocabularies were found containing semantic properties equivalent to elements of DC2AP, it was also necessary to declare new semantic properties to describe the elements that were not associated with the vocabularies found. All new RDF properties have been declared in accordance with the DCMI Namespace Policy (Powell et al., 2007), and then identified with URIs redirected by the server `http://purl.org` to RDF schema files on the server `http://dpi.ufv.br`. These new semantic properties together make up the vocabulary "DC2AP Element Set", identified by URI `http://purl.org/dc2ap/elements/`.

Besides the association and the creation of RDF semantic properties to describe the elements of DC2AP, some elements were also associated with Vocabulary Encoding Schemes and Syntax Encoding Schemes. Both types of schemes are lists containing controlled vocabularies used to standardize the registration of metadata of a machine-processable application profile (Coyle & Baker, 2009), thus preventing the occurrence of ambiguity between the terms used in the documentation of analysis patterns.

Due to space limitations, both the URIs referring to the RDF semantic properties and the lists of controlled vocabularies associated with elements of DC2AP are not presented in this article. However this information can be obtained from the technical description of this application profile available at: `http://purl.org/dc2ap/TechnicalDescription`.

By applying DC2AP to document analysis patterns through RDF properties, the metadata repository proposed by APRI architecture presented in Figure 2, actually becomes a source of

Linked Data of the specific domain of analysis patterns. Examples of analysis patterns documented by machine-processable RDF properties can be found at: `http://purl.org/dc2ap/Examples`. A metadata editor able to document machine-processable analysis patterns with DC2AP is still under construction. A beta version of this software can be found at: `http://purl.org/dc2ap/Editor`.

4. Conclusion and Future Work

DC2AP enables description of analysis patterns in detail, since it was designed with the focus on specifics of the domain of this kind of pattern. The specifics are mainly described by the template proposed by Pantoquilho et al. (2003), and used as the semantic base in the DC2AP creation. This application profile of Dublin Core was developed in line with the proposal of APRI (Vegi et al., 2012), thus intended to be part of this reuse infrastructure to solve the problems of documentation, organization, searching and access to analysis patterns.

Because it is a machine-processable application profile, DC2AP allows the creation of digital collections of analysis patterns that are searchable via search engines. In this way, analysis patterns are retrieved more quickly and efficiently, giving users an ease of access to well documented analysis patterns.

In order to fit the concept of a machine-processable application profile, DC2AP had its properties and application rules described using RDF files identified through URI. These descriptions provide for a metadata repository proposed by the APRI architecture to behave as a Linked Data source in the global database making up the Web of Linked Data. When they are described as Linked Data, analysis patterns have their potential for reuse expanded and enable greater integration of the patterns with other data from distributed heterogeneous sources (Bizer et al., 2009).

As future work, we intend to validate the acceptability of DC2AP through a quantitative research applied to the user community of analysis patterns. Besides, it is intended as well to use the DC2AP as the basis for the definition and implementation of Web services proposed in APRI to search, view, apply and contribute experiences of using analysis patterns. These services will be domain specific Linked Data applications and thus can discover new sources of data from the RDF links existing in the analysis patterns described in an APRI. This possibility of finding new sources of data will allow greater dynamism to the metadata repositories of an APRI, since the services will be able to retrieve new related data found in other sources whenever they are published on the Web. As another future work area, we intend to extend the APRI architecture for other types of reusable computational artifacts such as frameworks and design patterns. With this architecture extension it will be possible to create new Dublin Core application profiles to document the new reusable computational artifacts supported by APRI.

Acknowledgements

This work was partially financed by Brazilian funding agencies: FAPEMIG, CNPq and CAPES. The authors also thank the financial support from the Sydle enterprise.

References

Béjar, Rubens, Miguel Á. Latre, Javier Nogueras-Iso, Pedro R. Muro-Medrano, & Francisco J. Zarazaga-Soria. (2009). An architectural style for Spatial Data Infrastructures. *International Journal of Geographical Information Science, 23*(3), 271-294.

Berners-Lee, Tim, Robert Cailliau, Ari Luotonen, Henrik F. Nielsen, & Arthur Secret. (1994). The World-Wide Web. *Communications of the ACM, 37*(8), 76-82.

Berners-Lee, Tim, Roy Fielding, & Larry Masinter. (2005). Uniform Resource Identifier (URI): Generic syntax. Retrieved March 20, 2012 from http://tools.ietf.org/html/rfc3986

Berners-Lee, Tim. (2000). *Weaving the Web: The Past, Present and Future of the World Wide Web by its Inventor*. London: Texere Publishing.

Bizer, Christian, Tom Heath, & Tim Berners-Lee. (2009). Linked Data - The story so far. In T. Heath, M. Hepp, & C. Bizer (Eds.), *Special Issue on Linked Data, International Journal on Semantic Web and Information Systems (IJSWIS), 5*(3) (pp. 1-22). Elsevier.

Blaimer, Nicolas, Andreas Bortfeldt, & Giselher Pankratz. (2010). *Patterns in object-oriented analysis*. Working Paper No. 451, Faculty of Business Administration and Economics, University of Hagen (Germany).

Coyle, Karen, & Thomas Baker. (2009). Guidelines for Dublin Core Application Profiles. Retrieved March 20, 2012 from http://dublincore.org/documents/profile-guidelines/

Davis-Júnior, Clodoveu A., & Leonardo L. Alves. (2005). Local Spatial Data Infrastructures based on a Service-Oriented Architecture. *Proceedings of the 8th Brazilian Symposium on GeoInformatics* (GeoInfo), 30-45.

DCMI. (1998). Dublin Core metadata element set, version 1.0: Reference description. Retrieved March 13, 2012 from http://www.dublincore.org/documents/1998/09/dces/

Fernandez, Eduardo B., & Xiaohong Yuan. (2000). Semantic Analysis Patterns. *Proceedings of the 19th International Conference on Conceptual Modeling* (ER 2000), LNCC vol. 1920. Springer, 183-195.

Fernandez, Eduardo B., & Xiaohong Yuan. (2009). An Analysis Pattern for invoice processing. *Proceedings of the 16th Conference on Pattern Languages of Programs* (PLoP), 01-10.

Fowler, Martin. (1997). *Analysis Patterns: reusable object models*. Addison-Wesley Publishing.

Gamma, Erich, Richard Helm, Ralph Johnson, & John Vlissides. (1994). *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Publishing.

Hartig, Olaf. (2009). Provenance information in the Web of Data. *Proceedings of the 2nd Workshop on Linked Data on the Web* (LDOW2009).

Hay, David C. *Data Model Patterns: Convention of thoughts*. (1995). New York: Dorset House Publishing.

Keizer, Johannes, Ashan Morshed, Caterina Caracciolo, & Gudrun Johannsen. (2011). Thesaurus alignment for Linked Data publishing. *Proceedings of the International Conference on Dublin Core and Metadata Applications 2011* (DC-2011), 37-46.

Klyne, Graham, & Jaremy J. Carroll. (2004). Resource Description Framework (RDF): Concepts and abstract syntax. Retrieved March 21, 2012 from http://www.w3.org/TR/rdf-concepts/

Lisboa-Filho, Jugurta, Cirano Iochpe, & Karla A. V. Borges. (2002). Analysis Patterns for GIS data schema reuse on urban management applications. *CLEI Electronic Journal, 5*(2), 01-15.

Meszaros, Gerard, & Jim Doble. (1997). A pattern language for pattern writing. In R. C. Martin, D. Riehle, & F. Buschmann (Eds.), *Pattern languages of program design 3* (pp. 529-574). Boston: Addison-Wesley.

Nebert, Douglas D. (2004). *Developing Spatial Data Infrastructures: The SDI Cookbook*. GSDI-Technical Working Group. Retrieved from http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf

Pantoquilho, Marta, Ricardo Raminhos, & João Araújo. (2003). Analysis Patterns specifications: Filling the gaps. *Proceedings of the 2nd Viking Conference Pattern Languages of Programs* (PLoP), 169-180.

Powell, Andy, Harry Wagner, Stuart Weibel, Thomas Baker, Tod Matola, Eric Miller, & Pete Johnston. (2007). Namespace policy for the Dublin Core Metadata Initiative. Retrieved March 21, 2012 from http://dublincore.org/documents/dcmi-namespace/

Raminhos, Ricardo, Marta Pantoquilho, João Araújo, & Ana Moreira. (2006). A systematic Analysis Patterns specification. *Proceedings of the 8th International Conference on Enterprise Information Systems* (ICEIS), 453-456.

Sauermann, Leo, Richard Cyganiak, Danny Ayers, & Max Völkel. (2008). Cool URIs for the Semantic Web. W3C Interest Group Note 03 December 2008. World Wide Web Consortium. Retrieved March 13, 2012 from http://www.w3.org/TR/cooluris/

U.S. Library Of Congress. (2008). Dublin Core to MARC Crosswalk. Retrieved March 20, 2012 from http://www.loc.gov/marc/dccross.html

Vegi, Lucas F. M., Douglas A. Peixoto, Liziane S. Soares, Jugurta Lisboa-Filho, & Alcione P. Oliveira. (2012). An infrastructure oriented for cataloging services and reuse of Analysis Patterns. *Proceedings of BPM 2011 Workshops* (rBPM 2011), LNBIP vol. 100, Part 4. Springer, 338-343.