APRImora: An Architecture Based on Semantic Web to Retrieve Information in Pattern Repositories

Angélica Aparecida de Almeida Ribeiro¹, Jugurta Lisboa-Filho^{1*}, Lucas Francisco da Matta Vegi¹, Alcione de Paiva Oliveira¹, Regina Maria Maciel Braga², Emílio José de S. Fonseca¹ ¹ Departamento de Informática – Universidade Federal de Viçosa. 36570-000 – Viçosa – MG – Brazil. ² Departamento de Ciência da Computação - Universidade Federal de Juiz de Fora. 36036-900 - Juiz de Fora-MG, Brazil.

* Corresponding author. Email: jugurta@ufv.br Manuscript submitted October 22, 2015; accepted December 25, 2015. doi: 10.17706/jsw.11.3.272-286

Abstract: Software patterns are computing artifacts used to document knowledge that may be reused during software development process. There are several types of patterns, such as analysis, design, and architectural, among others. Design patterns are the most well known by designers, but many of them are described in books and scientific papers, a recurring way of documenting patterns that limits their reuse potential. Aiming to not only minimize this limitation but also provide ways of recovering contextualized knowledge in these patterns, the present paper presents the architecture APRImora, an extension based on Semantic Web of the Analysis Patterns Reuse Infrastructure (APRI). In this architecture, the patterns are documented by metadata defined as application profiles of the Dublin Core standard and stored in the RDF, allowing them to be discovered by search engines. The APRImora architecture helps designers discover and reuse software patterns based on semantic relations, which favors their dissemination and reuse.

Key words: Analysis pattern, design patterns, pattern repositories, retrieving information, semantic web.

1. Introduction

Reuse mechanisms such as design patterns [1] and analysis patterns [2] are crucial computing artifacts in the software development process. However, the great challenge a designer still faces today is discovering a pattern that solves a given problem. Search engines not always help because, although the search is facilitated by its constant improvement, a simple query may return a large number of results, many not relevant for the context. For example, when searching on Google for the design pattern Singleton typing only the word Singleton, with no additional information, approximately 6.6 million results are returned, a great number of which irrelevant and inappropriate to solve the problem.

According to Cunha *et al.* [3], web data are organized to be read and understood by humans and not by software agents. This is due to the way the Web organizes the information available on the Internet by using hypertext provided by the HTML language. Its organization allows for a more user-friendly interaction with the global network, however, the HTML language is used only to list resources to the user with no meaning presented or association on the meanings.

In order to solve this issue, the Semantic Web was created, which aims to transform the Web of documents into the Web of data, in which all data on the Web must be associated to a given semantics and are connected among themselves. According to Cunha [3], in the Web setting, this would represent

attributing meaning to the data, connecting them with other datasets or knowledge domains, and creating a relation of meaning among the content published on the Internet so that they can be analyzed not only by humans, but also by software agents. That results in more effective searches with results closer to what the user wants.

In order for computing artifacts to be easily retrieved and reused, Vegi *et al.* [4] proposed the APRI, i.e., Analysis Patterns Reuse Infrastructure. Its goal is to allow the analysis patterns to be documented, disseminated, and enhanced through an approach based on metadata and Web services. The analysis patterns added to the APRI are documented by an application profile of the Dublin Core metadata standard and stored in the RDF format so that they are accessible to search engines [5].

The present paper describes a proposal for extending the APRI architecture for an architecture based on the Semantic Web, turning it into a Patterns Reuse Semantic Infrastructure called APRImora. The remaining of the paper is structured as follows. Section 2 presents a review of the technologies of the Semantic Web, besides other important technologies related to the research. Section 3 lists some correlated studies. The architecture proposed for APRImora is presented in Section 4. Section 5 describes the DC2DP Metadata Editor, which documents design patterns in the infrastructure proposed. The final considerations and some future studies are described in Section 6.

2. Theoretical Framework

Subsection 2.1 comments on the Semantic Web and its importance for the research performed. The analysis patterns are stored in the APRI using the RDF format, hence a language must be used that is able to query RDF documents. Some aspects of the SparQL query language are described in Subsection 2.2. Subsection 2.3 discusses the Linked Data concept, since the patterns stored in APRImora are documented following this concept. The use of Web services allows the patterns to be reused both by humans and automatically by CASE tools, thus Web services are described in Subsection 2.4. Subsection 2.5 discusses the Dublin Core metadata standard since its simplicity and versatility helps construct profiles to document specific domains such as analysis and design patterns.

2.1. Semantic Web

The Web has changed the way people communicate and do business. This has transformed the developed world towards a knowledge-based economy and, more importantly, a knowledge-based society [6].

According to Berners-Lee *et al.* [7], the Web developed more quickly as a medium for documents to the people than to the data and information that can be automatically processed. Consequently, most times a search is conducted, thousands of results with little or no relevance are obtained. This occurs because the meaning of the Web content cannot be processed by machines, besides the ability of software to interpret sentences and extract useful information for the users being limited [6].

From this context appeared the Semantic Web, which makes up for this issue by providing means of organizing data and allowing them to be interpreted by computers. The Semantic Web resources allow search results to be improved and facilitates the automation of relevant tasks. Berners-Lee *et al.* [7] add that the Semantic Web is not a separate Web, but an extension of the current one where information has a well-defined meaning, thus allowing computers and people to collaborate.

For Berners-Lee *et al.* [7], the challenge of the Semantic Web is to provide a language that expresses both the data and the rules to think such data, and that allows the rules of any knowledge system to be exported to the Web. Fig. 1 illustrates the layers that make up the Semantic Web. Details on each layer can be found in [8].

Journal of Software



Fig. 1. Architecture of the semantic web [8].

2.2. SparQL

RDF is a flexible and extensible format used to represent information on Web resources such as personal information and metadata on digital artifacts such as music and images, among others. Moreover, it provides means to integrate different sources of information. A standardized query language to search RDF data offers developers and end users a way of writing and consuming the search results in this broad range of information.

SPARQL Protocol and RDF Query Language (SparQL) is a language and a data-access protocol developed and recommended by the W3C for queries on RDF files [9].

Fig. 2 illustrates a SparQL query extracted from [10] to help understand this language and its syntax. Fig. 2(a) shows a triple corresponding to the data. Fig. 2(b) illustrates the query, in which the clause SELECT identifies the variables that will appear in the search result and the clause WHERE provides the basic pattern to be combined with the data. The basic pattern in this example consists of a single variable (?title) in the object position. Fig. 2(c) shows the result of the query.

```
<http://example.org/book/book1>
<http://purl.org/dc/elements/1.1/title> "SPARQL Tutorial" .
(a)
SELECT ?title
WHERE
{
<http://example.org/book/book1><http://purl.org/dc/elements/1.1/title
> ?title .
}
(b)
"SPARQL Tutorial"
(c)
```

Fig. 2. (a) RDF triple, (b) SparQL query, (c) Query result [10].

2.3. Linked Data

The goal of the Semantic Web is to provide means to organize data and enable its interpretation by computers, thus making work more helpful, besides developing systems that can support reliable interactions on the network [11]. Moreover, the Semantic Web is seen as a layer of the Web in which one can publish, obtain, and use data that may be directly or indirectly processed by machines [11].

However, its importance goes beyond only making data available on the Web since it connects such data so that a person or machine is able to explore them. This interconnection of data allows for more information related to a given piece of data searched to be found [12]. It provides means for the conventional Web to be transformed into a Web with machine-processable data. This connection is called Linked Data.

Linked Data refers to the set of best practices to publish and connect data on the Web. These practices led to the creation of a global data space that contains billions of pieces of information, the so-called Web of Data [13]. According to Berners-Lee [12], as with the hypertext Web, the Web of Data is built from Web documents. However, the difference between them is that on the hypertext Web the related links are anchored in documents written in HTML. On the Web of Data, arbitrary things are connected by the description in RDF using Uniform Resource Identifiers (URIs), which identify any type of object or concept.

The URI and the RDF format are technologies that help the Linked Data to support the Semantic Web. The URI provides a simple and extensible way of uniquely identifying through a string of characters an abstract or physical resource [14]. RDF is a framework to represent Web information and its structure is made up of a set of triples called RDF graph, formed by the concepts of subject, predicate, and object. Subject and object are URI identifiers that represent the respective related data and the predicate specifies the semantics of this relationship, i.e., the type of relationship among the data [15].

2.4. Web Service

Web service is a solution used in systems integration and communication between different applications, with the advantage of standardized communication between services, which allows the platform and programming language to be independent, such as a system developed with Java running on a Linux server being able to seamlessly access a service implemented on .NET running on a Microsoft server [16].

Semantic Web Service (SWS), as the name suggests, is an intersection of the Web services and Semantic Web [17]. According to Martin *et al.* [17], the main focus of SWSs is the use of richer and more declarative descriptions of elements in the distributed computing dynamics. These descriptions enable complete automation, more flexible service, the use and construction of more powerful tools, and better methodologies to work with services.

The richer representation of the structure enables a broader specification in many aspects of the services, thus SWSs can provide a solid basis for a large number of activities during the Web service's life cycle [17].

2.5. Dublin Core Metadata Standard

The set of elements of the Dublin Core metadata standard was called "Dublin" because it was created in 1995 in a workshop in the city of Dublin, OH, USA. "Core" because its elements are basic and generic, which allows it to be used to describe a wide range of resources [18]. Given its versatility, the Dublin Core (DC) metadata standard is able to document anything from works of art to books in a library.

The Dublin Core standard is made up of a set of 15 elements [19], namely: title (name of the resource), creator (person or organization responsible for the content), subject (subject covered by the document), description (description of the content), publisher (entity responsible for making the content available), contributor (entity responsible for contributing to the content), date (date associated to an event occurring in the content's life cycle), type (the nature or genre of the content), format (format in which the resource is presented), identifier (a unique identifier for the content), source (source of the content), language (the language the content was written in), relation (relationship of the resource with other resources), coverage (spatial and temporal characteristic of the resource), and rights (information on the rights of the resource).

Although it is a simple metadata standard that doesn't offer enough resources to describe data of complex domains [20], Dublin Core's versatility in documenting artifacts of several types is achieved by extensions known as application profiles. Since 2000, the Dublin Core community has proposed profiles in which the main idea is to adequate the elements to generate new metadata sets to meet the specific requirements of applications from different domains [21].

In the APRI context, two Dublin Core metadata profiles were defined: DC2AP [22] to document analysis patterns and DC2DP [23] to document design patterns.

According to Vegi *et al.* [22], DC2AP was developed from the mapping of elements contained in the Dublin Core metadata standard with the elements of the template proposed in the study by Pantoquilho *et al.* [24] to document analysis patterns. DC2AP is a machine-processable application profile, which allows the analysis patterns to be described and published as Linked Data through RDF files. DC2AP aims to enhance the retrieval and reuse of analysis patterns through a description that enables a more precise treatment by a computer, offering detailed information on the analysis patterns that previously could not be retrieved by search engines.

The Dublin Core application profile to document design patterns (DC2DP), proposed by Ribeiro *et al.* [23], is an application profile of the Dublin Core metadata standard created specifically to describe the design patterns in a way that is compatible with the APRI proposal. The DC2DP elements were defined from the combination of the metadata sets, i.e., the Dublin Core metadata standard elements, the elements of the template by Gamma *et al.* [1], and the DC2AP elements. Dublin Core is the basis of DC2DP, so all elements belonging to this metadata standard are part of the DC2DP profile. The template by Gamma *et al.* [1] contributes with the specific elements to document design patterns, while the version control elements were reused from the DC2AP profile.

Similarly to DC2AP, DC2DP is also machine-processable, allowing design patterns to be described and published as Linked Data through RDF files, therefore the patterns can be retrieved and reused more precisely by a computer. A detailed technical description of these two application profiles, containing the semantic description of each of their elements as well as details of their application rules, can be found at http://www.dpi.ufv.br/projetos/apri.

3. Related Work

Monteiro [25] proposes an architecture based on Semantic Web for digital educational repositories in healthcare that make available learning objects with educational aspects described in metadata forms. Learning objects (LO) are structured resources used to make self-learning content available and are appropriate to develop Distance Learning content. Making LO available on the Web contributes to lower costs with their production. In order to allow LOs to be available, Learning Management System (LMS) collections are created so that the LOs are accessed from the course environments themselves. Digital Educational Repositories (DER) can also be developed, where an increasingly larger number of objects is shared. Hence, the DER must have an appropriate architecture and carry enough descriptive information for the selection of LOs, particularly regarding educational aspects.



Fig. 3. APRI architecture.

Thus, Monteiro [25] proposes adding technologies to the DER architecture capable of refining the search result in LO by using the Semantic Web as a key element to contribute to the search refinement.

Another study directly related with the proposal presented in the current paper is the Analysis Patterns Reuse Infrastructure (APRI) proposed by Vegi *et al.* [4]. APRI's architecture is inspired by the Spatial Data Infrastructures (SDIs), which, in turn, are defined as collections of technologies, policies, and institutional arrangements that facilitate the availability of and access to spatial data [26]. Many SDIs employ the concept of service-oriented architecture (SOA), thus allowing shared, distributed, and interoperable environments to be developed based on Web services [27].

Similarly to the SDI, APRI describe the analysis patterns with metadata and store it in a repository. Its goal is to provide mechanisms for the analysis patterns to be made available and found by designers, which helps in these patterns being publicized and, consequently, reused. More details about APRI can be found in [4].

4. APRImora: An Architecture Based on Semantic Web for Pattern Reuse

Initially, the APRI was developed focusing only on the reuse of analysis patterns [22]. Later, Ribeiro *et al.* [23] proposed adding design patterns to the APRI repositories, thus broadening its application scope and allowing the patterns to be reused not only during the software development analysis, but also during the design stage.



Fig. 4. APRImora architecture.

The patterns stored in the APRI repository can be retrieved from a Web browser and also through services that allow the patterns to be discovered, catalogued, and reused from the Pattern Portal (Fig. 3). However, since search engines are used, the system may return a number of results that are irrelevant to the research context and inappropriate to meet the needs of the search.

The patterns documented in the APRI are described as Linked Data, which helps in their retrieval. Nevertheless, the metadata still have a limited formal semantics. The most common method to search for a dataset consists in searching for the elements using keywords that match the metadata.

The fact that metadata-based searches return a large amount of results outside the research context when using search engines requires the elements to be semantically defined. That gives the meaning of the elements and is a key point to help recover information from the repositories precisely. Thus, the present study extends the APRI to an architecture based on the Semantic Web able to help the user better recover patterns contained in its repositories, which originates APRImora, a Pattern Reuse Semantic Infrastructure (Fig. 4). An initial version of APRImora was presented in [28].

APRImora architecture's core consists of the same components of the APRI, but with semantics added to the elements and to the way they are retrieved and inserted into the structure. The following subsections describe the components of this architecture.

4.1. Pattern Portal

The Pattern Portal is a portal composed of a set of websites focused on obtaining patterns. It provides tools and services for the discovery, cataloguing, and reuse of these patterns. Also available in the Pattern Portal is the metadata profile editor for analysis patterns (DC2AP Metadata Editor) [29] and the profile editor for design pattern metadata (DC2DP Metadata Editor).

Each element part of the metadata set of these two editing tools has rules regarding its enforcement, occurrence, and value type, which must be validated for the documented pattern to be consistent. After documenting the analysis or design pattern, the enforcement rules referring to these metadata are validated by a Web service called Validation Service. This service verifies, for example, if all fields whose occurrence rule is Mandatory [M] have been filled. If not, the user will receive an error message.

The occurrence rules, regarding the plurality of the elements, are validated by the interface itself using buttons that add fields to elements that must be multiple while omitting simple fields.

Some elements are filled and validated by controlled vocabularies, i.e., sets of standardized terms that aid the data input and output in an information system, thus promoting greater precision and effectiveness in the communication between users and the information system [30]. Examples of controlled-vocabulary elements include type, format, language (DC2AP), and type and language (DC2DP).

When the Validation Service returns an affirmative response, the RDF Service can be used to generate an RDF document of the new pattern, which is automatically stored in the Metadata Repository in a user-defined folder.

Besides the metadata editing tools, the Apache Jena framework, or simply Jena, is part of the Pattern Portal. Jena is a free, open-source Java framework to build Semantic Web and Linked Data applications. This framework is made up of different APIs interacting among themselves to process RDF data [31].

In the APRImora architecture, this framework is used to retrieve information from the data and metadata repositories, along with the SparQL language, using the ARQ module, a query mechanism used by Jena that supports the SparQL language [31].

4.2. Semantic Web Layers

The URI, XML and Namespace, RDF and RDFS, and OWL technologies, belonging to the Semantic Web layers, are used directly in the documents contained in APRImora's Metadata Repository. Other Semantic Web layers (Fig. 1), such as Logic Framework, Proof, Trust, Signature, and Encryption, were not added to APRImora's structure since there are still no recommendations by the W3C to use these layers [25].

The URI layer is used to assign a unique name to the elements using the Unicode standard, which is the universal standard adopted for this addressing. The URI guarantees the uniqueness of the elements, thus preventing ambiguity. In APRImora, the URI address is used to uniquely name the documented patterns and relate them to other patterns. The field Identifier is responsible for receiving the URI value, indicating the uniqueness of the patterns documented in APRImora through the DC2AP or DC2DP, metadata, since the field Relation and its specializations are responsible for receiving URI values that associate several patterns documented in the reuse infrastructure described in this study.

In APRImora, the field Identifier receives a user-informed URI address generated using Persistent

Uniform Resource Locators (PURLs)¹. PURLs are Web addresses that act as permanent identifiers, i.e., allow the resources to change address (server) without affecting the systems that depend upon them. Thus, the Web addresses may migrate from one domain to another without losing the reference of the allocated resource. In order to generate a URI, the user must have an administrator account on the site purl.org, fill out the fields referring to the PURL they wish to create, and await the verification of whether the URI they intend to register is available or if it has already been taken by another user.

On the XML layer, the Namespaces allow the content syntax to be made available on the Web to be defined and its structure to be specified [25]. As mentioned before, APRImora uses controlled vocabulary and syntax in some elements of the DC2AP and DC2AP metadata profiles. Using the XML layer in APRImora allows the metadata to be defined, thus characterizing the syntax of the elements that comprehend the DC2AP and DC2DP profiles.

The XML language allows the syntax of the content available on the Web to be defined, i.e., the content structure, the presentation form, and the content itself. However, only the syntax might not be enough. For example, based only on the term "author" of a design pattern, it cannot be inferred whether the "author" is a person or a team, nor other patterns belonging to the same "author" can be listed. In order to solve this limitation, the use of the RDF data model is proposed, presented in the third layer of the Semantic Web architecture.

These types of data are not dealt with in an RDF document, but rather their properties or information units. Hence, RDF goes beyond XML in terms of syntax and structure of the documents available on the Web, although this advance is not enough since it still does not allow the terms to be conceptualized and disambiguated [25]. In APRImora, the RDF layer is used to formalize the metadata semantics.

The ontology layer is used to describe the domain of interest. It is implemented using the OWL language, which was designed to be used by computing applications that need to process content from the information instead of only presenting it to humans [25]. This language expands the ability of inferences aided by XML, RDF, and RDFS, thus providing an additional vocabulary along with formal semantics. In APRImora, the OWL layer is used to help filling out the metadata.

SparQL is a query language used to retrieve information from an RDF document. According to Monteiro [25], SparQL provides protocols to query and handle RDF graphs, thus allowing several inferences. This language is used perpendicularly to the RDF and OWL layers. In APRImora, the information is retrieved from repositories using the SparQL language in the ARQ module belonging to the Apache Jena framework.

4.3. Linked Open Data

The patterns documented and stored on the Metadata Repository are described using the Linked Data concept, generating a Linked Data cloud among the patterns in this repository, as shown in Fig. 5. The light-gray nodes indicate design patterns and the dark-gray nodes illustrate analysis patterns. This Linked Data cloud represents the connections made through the URI among the design patterns themselves and between the design and analysis patterns.

The connections presented in Fig. 5 indicate a pattern is related with others. According to Gamma *et al.* [1], some patterns can be used as a set, e.g., Composite is constantly used along with Iterator or Visitor. Some patterns can be used as alternatives, as is the case of Prototype used in place of Abstract Factory. Moreover, some patterns can result in similar designs, although their purposes are different, such as the structure diagrams of Composite and Decorator being similar.

The main advantage of relations among patterns is allowing the designer to have multiple ways of thinking about the patterns, thus deepening their perception on what each pattern does and in which

¹http://purl.org/docs/index.html

situation it can be employed. These multiple ways allow the designer to choose the best way to use the pattern and those that complement it, enhancing designing as the patterns are reused.



Fig. 5. Linked Data cloud of part of the APRImora metadata repository.

Linked Open Data, according to Zaidan [32], is a worldwide data publication effort, making them open and available to be used anywhere. In order for these data to be published and connected, Linked Data must be used. Thus, Linked Open Data was added to the APRImora architecture through a connection with the Metadata Repository to indicate that the Linked Data cloud generated by the patterns documented and stored in this repository are now part of the Linked Open Data, which makes the patterns open and available to be used.

4.4. Pattern Repository

The Pattern Repository is a repository containing descriptions of diagrams expressed in XMI (XML Metadata Interchange), JPG, and PNG, which represent the solutions proposed by the patterns, thus allowing them to be used by portrayal and collaboration services. With the APRImora proposal, the Pattern Repository is made up of two repositories, Design and Analysis, which relate to each other through Linked Data. This way, a design pattern can be referenced by an analysis pattern and vice-versa.

The relation between the analysis and design patterns has the advantage of sharing the experience of using these patterns, i.e., the analyst can list which design patterns can be used to implement a given analysis pattern. The same occurs regarding design patterns, in which the designer can indicate which analysis patterns that design pattern can implement.

4.5. Web Service and Semantic Web Service

Patterns and services can be searched for in APRImora either by a Human User through search engines or by Client Software through Web Service or Semantic Web Service. The search carried out by a Human User follows the steps shown in the sequence diagram in Fig. 6.

As seen in Fig. 4, the connection between the Web Service/Semantic Web Service and the Pattern Portal is two-way since the services can use the tools to carry out searches and the Pattern Portal uses these services along with its tools. For example, the metadata editors use Web Services to validate (Validation Service) the metadata fields, visualize diagrams (Portrayal Service), and download the RDF files (Access Service) that

contain the pattern descriptions. The Semantic Web Service, in turn, is used by the Pattern Portal to generate an RDF document (RDF Service) and the hyperbolic tree (Hyperbolic Tree).



Fig. 6. Sequence diagram of a search carried out by a human user.



Fig. 7. Hyperbolic tree with design patterns.

With the search result returned, the user can use a Semantic Web Service to generate a hyperbolic tree, which is an easily understood graph on the pattern relationship. In the context of APRImora, the central node of this tree will always be the pattern returned according to the search criteria, while the other nodes connected to the central node are patterns related with it with their respective nodes. This is possible thanks to the use of Linked Data.

Fig. 7 illustrates a hyperbolic tree corresponding to the connections of possible design patterns existing in the pattern repository, whose root node is the pattern Prototype. The advantage of using a hyperbolic tree is the fact that it allows the user to visually navigate among the connections between the patterns.

APRI had only the component Web Service, which in the APRImora architecture is divided into two components, Web Service and Semantic Web Service. Web Service makes the following services available:

- *Portrayal Service* services that support the visualization of the diagrams of solutions proposed by the patterns;
- *Catalog Service* service that allows the patterns and services to be discovered and used based on their describing metadata;
- Access Service services that allow accessing and downloading the patterns;
- *Collaboration Service* services that allow designers and developers to share their use experiences to improve the patterns;
- *MDA Service* services that allow source code to be generated from diagrams;
- Validation Service service that allows metadata to be validated.

The Semantic Web Service component makes the following services available:

- *RDF Service* service that allows RDF documents to be generated;
- *Hyperbolic Tree* service that generates a visualization of the connections among the patterns and enables navigation among them.

The Web Service and Semantic Web Service components have a two-way connection with Linked Open Data, which indicates the services can carry out external searches as long as the result obtained is in a format compatible with those supported by the search carried out by APRImora. Both in APRImora's internal architecture and in the environment external to the infrastructure at hand, the language used to retrieve information is SparQL, while the search is done only in RDF files.

The connection between Web Service/Semantic Web Service and Linked Open Data leads to the possibility of recovering and discovering other data sources related to the patterns whenever these are published. The opposite is also possible since external agents can search the APRImora repositories using services.

5. DC2DP Metadata Editor

The DC2DP Metadata Editor, mentioned in section 4.1, is a Web system that allows the user to create, edit, and save metadata according to the rules described in the DC2DP application profile. With this system, the metadata can be edited and exported to RDF files. Fig. 8 shows the homepage of the DC2DP Metadata Editor.

In the DC2DP Metadata Editor, one can navigate among the DC2DP profile elements using a directory tree (Fig. 8a). In order for the user to edit the metadata in accordance with the DC2DP rules, each element has an acronym right below its name (Fig. 8b) indicating the enforcement rules, occurrence, and value type allowed. Table 1 presents these rules' acronyms.

Table 1. DC2DP Rules Acronyms			
Enforcement	Occurrence	Value Type	
[M] Mandatory [O] Optional [Cd] Conditional	[S] Simple [Mu Multiple	[St] String [D] Data [U] URI [N] Null [UNS] URI, number or string [US] URI or string	

Besides the acronyms, the user can check the meaning of each element using the Help feature (Fig. 8(c)). This option sends the user to the DC2DP elements' technical description document, exactly in the section corresponding to the element desired, which is done based on its URI defined by the PURL.

For elements with multiple occurrence, there are four types of buttons (Fig. 8(d)) for the user to insert additional fields to describe further information related with that element (+), exclude the field added (x),

and change the order of these fields (up and down arrows).

Once the user has edited the pattern elements, the following metadata editor features presented in (Fig. 8(e)) can be used:

- Open opens previously documented patterns saved in the RDF format;
- *Clear* deletes the data of the fields on the current page;
- *Clear all* deletes all data of all pages;
- Validity checks if all mandatory fields have been filled in;
- *RDF* saves an RDF document containing the elements of the documented pattern.

DC2 Metad	DP Departamento de Universidade I Informatica de Viço	Federal Sa
Home File Services He		
+ Expand All DC2DP I Identifier Classification Creator Subject Description Delte Publisher Contributor Type Source Language Deltation Coverage Relation Coverage Rights Deltetory De	1. Identifier An unambiguous reference to the resource within a given context. Identifier: MISILWSI (b) 2. Title A name given to the resource. Title: MISIBI 2.1. Alternative Title Any form of the title used as a substitute or alternative to the formal title of the resource. One per line. Alternative Titles: Image: 1: Image: Image: Image: Image: Image: Image:	
		\Rightarrow
	Open Clear All (e) Validate RDF	

Fig. 8. DC2DP metadata editor.

6. Conclusions and Future Work

One of the big issues faced by developers regarding the analysis and design steps is discovering reuse patterns that meet their needs. The search engines used to retrieve the patterns do not always help since a simple search returns a large number of results, many of which irrelevant to that software-design context.

The APRI was proposed aiming to help developers search for patterns. The analysis and design patterns contained in the APRI are documented using the DC2AP and DC2DP application profiles, respectively, and later stored in their repositories. However, storing the patterns in repositories is not enough since there are hundreds of patterns and a manual search can be exhausting while search engines can return a large number of results. Thus, the main goal of this paper was to propose the APRI extension for an architecture able to help the user better retrieve patterns contained in its repositories.

To that end, the APRI was extended to an architecture based on the Semantic Web, making it able to help the user retrieve patterns contained in its repositories, which originates APRImora, a Pattern Reuse Semantic Infrastructure. The Semantic Web layers were used to extend the APRI since they provide means for data to be organized in a way that they can be easily interpreted by computers. Moreover, the way the Semantic Web organizes the data allows the search results to be improved. Hence, by adding Semantic Web layers to the APRI, it is expected that the information on the computing artifacts stored in the repositories are more precisely retrieved, thus allowing the user to reuse the patterns that best meet the needs, besides avoiding wasted time with exhaustive research that do not return relevant results.

As future works, it is intended to create a functional prototype of the APRImora structure so that experiments can be made to prove its efficiency. In addition, adding software agents to the APRImora structure is proposed to help in the automatic recovery of knowledge.

Acknowledgment

This work was financed by Brazilian funding agencies FAPEMIG, CNPq and CAPES.

References

- [1] Gamma, E., *et al.* (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Reading: Addison Wesley Publishing Company.
- [2] Fowler, M. (1997). Analysis Patterns: Reusable Object Models. Addison-Wesley Publishing.
- [3] Cunha, D. R. B., Lóscio, B. F., & Souza, D. (2011). Linked data: Da web de documentos para a web de dados' Livro Texto dos Minicursos, SBC, Escola Regional de Informática Ceará, Maranhão, Piauí (ERCEMAPI), 79-99.
- [4] Vegi, L. F. M., Peixoto, D. A., Soares, L. S., Lisboa-Filho, J., & Oliveira, A. P. (2011). 'An infraestructure oriented for cataloging services and reuse of analysis patterns. *Proceedings of the 9th Int. Conf. on Business Process Management* (pp. 338-343).
- [5] Vegi, L. F. M., Lisboa-Filho, J., Crompvoets, J., Soares, L. S., & Braga, J. L. (2013). A dublin core application profile for documenting analysis patterns in a reuse infrastructure. *International Journal of Metadata, Semantics and Ontologies*, *8*(4), 267-281.
- [6] Antoniou, G., & Harmelen, F. V. (2008). A Semantic Web Primer. Cambridge, MIT Press, Massachusetts.
- [7] Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. Scientific American, 284(5), 28-37.
- [8] Berners-Lee, T., Fielding, R., & Masinter, L. (2005). Retrieved March 30, 2015, from Uniform resource identifier (URI): Generic syntax. http://tools.ietf.org/html/rfc3986
- [9] Hawke, S. (2013). SPARQL query results XML format. Retrieved March 24, 2015, from http://www.w3.org/TR/2013/REC-rdf-sparql-XMLres-20130321
- [10] Harris, S., & Seaborne, A. (2013) SparQL 1.1 query language. March 30, 2015, from http://www.w3.org/TR/2013/REC-sparql11-query-20130321
- [11] W3C. March 22, 2015, from http://www.w3.org/standards/semanticweb/.
- [12] Berners-Lee, T. (2006) Linked data. Retrieved March 30, 2015, from http://www.w3.org/DesignIssues/LinkedData.html
- [13] Bizer, C., Heath, T., & Berners-Lee, T. (2009). Linked data The story so far. *International Journal on Semantic Web and Information Systems*, *5*(*3*), 1-22.
- [14] Berners-Lee, T. (2005). Semantic web concepts. Retrieved March 30, 2015, from http://www.w3.org/2005/Talks/0517-boit-tbl
- [15] Klyne, G., Carroll, J. J., & McBride, B. (2014). RDF 1.1 concepts and abstract syntax. Retrieved March 31, 2015, from http://www.w3.org/TR/rdf11-concepts/
- [16] Pamplona, V. F. (2010). Web Services: construindo, disponibilizando e acessando web Services via J2SEeJ2ME.RetrievedMarch10,2015,from

http://javafree.uol.com.br/artigo/871485/Web-Services-Construindo-disponibilizando-e-ace ssando-Web-Services-via-J2SE-e-J2ME

- [17] Martin, D., & Domingue, J. (2007). Semantic web service. *IEEE Intelligent Systems*, 22(5), 12-16.
- [18] National Information Standards Organization NISO. (2007) The Dublin Core Metadata Element Set: An American National Standard, NISO Press, Bethesda, MD, USA.
- [19] DCMI Dublin core metadata initiative. (1999). Retrieved March 30, 2015, from http://www.dublincore.org/documents/1999/07/02/dces/
- [20] National Information Standards Organization NISO. (2001) The Dublin Core Metadata Element Set: An American National Standard, NISO Press, Bethesda, MD, USA
- [21] DCMI metadata basic. Retrieved March 10, 2015, from http://dublincore.org/metadata-basics/
- [22] Vegi, L. F. M., Lisboa-Filho, J., & Crompvoets, J. (2012). A machine-processable dublin core application profile for analysis patterns to provide linked data. *Proceedings of the Conf. on Dublin Core and Metadata* (pp. 23-32).
- [23] Ribeiro, A. A. A., Lisboa-Filho, J., Vegi, L. F. M., & Oliveira, A. P. (2014). DC2DP: A dublin core application profile to design patterns. *Proceedings of the 16th International Conference on Enterprise Information Systems* (pp. 209-215).
- [24] Pantoquilho, M., Raminhos, R., & Araújo, J. (2003). Analysis patterns specifications: Filling the gaps.
- [25] Monteiro, F. S. de. (2013). Web semântica e repositórios digitais educacionais na área de saúde: Uma modelagem com foco no objetivo de aprendizagem para refinar resultados de busca. University of Brasília, Brasília, Brazil.
- [26] Nebert, D. D. (2004). Developing spatial data infrastructures: The SDI cookbook. Retrieved February 26, 2015, from http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf
- [27] Davis, J. C. A., & Alves, L. L. (2005). Local spatial data infrastructures based on a service-oriented architecture. *Proceedings of the Symposium on Geoinformatics (Geoinfo)* (pp. 30-45).
- [28] Ribeiro, A. A. A., Lisboa-Filho, J., Vegi, L. F. M., Oliveira, A. P., Villela, R. M. M. B., & Fonseca, E. J. S. (2015). APRImora: A semantic architecture for patterns reuse. *Proceedings of the 27th International Conference on Software Engineering and Knowledge Engineering* (pp. 465-470).
- [29] Peixoto, D. A., Vegi, L. F. M., & Lisboa-Filho, J. (2013). DC2AP metadata editor: A metadata editor for an analysis patterns reuse infrastructure. *Proceedings of the 25th International Conference on Advanced Information Systems Engineering* (pp. 138-145).
- [30] Kobashi, N. Y. (2008). Vocabulário controlado: Estrutura e Utilização. Retrieved March 30, 2015, from http://www2.enap.gov.br/rede_escolas/arquivos/vocabulario_controlado.pdf
- [31] Apache. J. Retrieved March 10, 2015, from https://jena.apache.org/getting_started/index.html
- [32] Zaidan, F. H. (2013). LOD Linked Open Data Dados Abertos Vinculados. Retrieved April 15, 2015, from http://itweb.com.br/blogs/lod-linked-open-data-dados-abertos-vinculados/



Angélica Aparecida de Almeida Ribeiro got her MSc (master of science) degree in computer science from Federal University of Viçosa (UFV), Brazil, in 2014, a B.S. degree in computer science from the Faculdade Governador Ozanam Coelho, Ubá, Brazil, in 2011. She is hourly professor in the disciplines of Operating Systems and Introduction to Computers in the Pronatec course. She has experience in computer science WITH

emphasis in information systems. Her research interests are strongly focused on the database, geographic database and design patterns.



Jugurta Lisboa-Filho got his DS (doctor scientiae) degree in computing science from Federal University of Rio Grande do Sul, Brazil, in 2000 and the MS (master scientiae) degree also in computing science from Federal University of Rio de Janeiro, Brazil, in 1992. He is an associate professor at the Department of Informatics, Federal University of Viçosa, Brazil, since May 1988. His research topic interests include conceptual data modelling, reuse mechanisms, geographic information systems and spatial data infrastructure. He is

the chair of the master program in computing sciences, leads research projects and has authored and co-authored many journal papers, conference papers and book chapters.



Lucas Francisco da Matta Vegi got his MSc (master of science) degree in computer science from Federal University of Viçosa, Brazil, in 2012. He is an assistant professor at the Department of Informatics of the Federal University of Viçosa, Brazil. He has experience in computer science with emphasis in information systems. His research interests are strongly focused on the software engineering, mechanisms for reuse in software development

process and mobile computing.



Alcione de Paiva Oliveira received the B.S. degree in oceanography in 1985 from University of Rio de Janeiro State, a M.S. degree in computer science in 1990 from Military Institute of Engineering, and a Ph.D. degree in computer science in 1996 from Pontifical Catholic University of Rio de Janeiro. He is currently a full professor in the Department of Informatics, Federal University of Viçosa, Minas Gerais, Brazil. His research interests include

artificial intelligence, ontology, computational linguistics and multiagent systems.



Regina Maria Maciel Braga Villela is graduated in computer science from Federal University of Juiz de Fora in 1991. She received her master degree in computer and systems engineering from Federal University of Rio de Janeiro in 1995. She is Ph.D. in computer science and systems engineering from Federal University of Rio de Janeiro (2000). Professor Braga is currently an associate professor at the Federal University of Juiz de Fora,

working on the undergraduate program in computer science and is a permanent member of the master program of computer science. Braga has experience in computer science, with emphasis on software engineering and databases, specifically in the following topics: component based software development, ontologies, data integration and scientific workflows.



Emílio José de Souza Fonseca got his B.S. degree in computer science from Federal University of Viçosa, Brazil, in 2014. He participated in projects involving reuse mechanisms. He has experience in computer science with emphasis in information systems, working mainly with databases and Web systems development. Currently, he works in CEAD (Coordenadoria de Educação Aberta e à Distância - UFV), using Flash in the

development of instructional practical classes to support the teaching on students in primary, secondary and higher.