APRImora: A Semantic Architecture for Patterns Reuse

Angélica Aparecida de Almeida Ribeiro[1], Jugurta Lisboa-Filho[1], Lucas Francisco da Matta Vegi[1],
Alcione de Paiva Oliveira[1], Regina Maria Maciel Braga Villela[2] and Emílio José de S. Fonseca[1]

[1] Departamento de Informática
Universidade Federal de Viçosa
Viçosa-MG, Brazil, 36570-000

[2] Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora
Juiz de Fora-MG, Brazil, 36036-900

angelica.ribeiro@fagoc.br, jugurta@ufv.br, lucas.vegi@ufv.br, alcione@dpi.ufv.br, regina@acessa.com, emiliojsf@gmail.com

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 format, 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.**

Keywords- semantic Web; analysis pattern; design patterns; retrieving information; pattern repositories.

I. Introduction

Reuse mechanisms such as design patterns [8] and analysis patterns [7] 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. [5], 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. 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 [5]. 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. [20] 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 [21].

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 Semantic Web, besides other technologies related to the research. Section 3 lists some correlated studies. The architecture proposed for APRImora is presented in section 4. The final considerations and some future studies are described in section 5.

II. Theoretical Framework

A. Semantic Web

According to Berners-Lee et al. [1], 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.

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. [1] add that the Semantic Web is not a separate Web, but an extension of the current one.

(DOI reference number: 10.18293/SEKE2015-186)

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 [1]. Figure 1 illustrates the layers that make up the Semantic Web. Details on each layer can be found in [2].

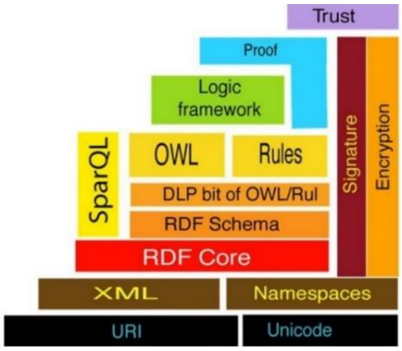

Figure 1. Architecture of the Semantic Web [2]

B. SPARQL Protocol and RDF Query Language

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 by means of a set of triples, formed by the concepts of subject, predicate, and object [10].

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].

C. 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 [22]. 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 [22].

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 [3]. 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 [4]. As with the hypertext Web, the Web of Data is built from Web documents [3]. 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 provides a simple and extensible way of uniquely identifying through a string of characters an abstract or physical resource [2].

D. 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 [15].

Semantic Web Service (SWS) is an intersection of the Web services and Semantic Web. According to Martin et al. [12], 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.

E. Dublin Core Metadata Standard

The set of elements of the Dublin Core metadata standard is composed by 15 basic and generic elements, which allows it to be used to describe a wide range of resources [6].

Although it is a simple metadata standard that doesn’t offer enough resources to describe data of complex domains, 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 to adequate the elements to generate new metadata sets to meet the specific requirements from different domains.

In the APRI context, two Dublin Core metadata profiles were defined: DC2AP [18] to document analysis patterns and DC2DP [17] to document design patterns.

The DC2DP elements were defined from the combination of the Dublin Core metadata standard elements, the elements of the template by Gamma et al. [8], 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. 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 can be found at `http://www.dpi.ufv.br/projetos/apri`.

III. Related Work

Monteiro [13] 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.

Thus, Monteiro [13] 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) [20]. 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 [14]. Many SDIs employ the concept of service-oriented architecture (SOA), thus allowing shared, distributed, and interoperable environments to be developed based on Web services.

Similarly to the SDI, APRI uses metadata to describe the analysis patterns to be stored in its 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.

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

Initially, the APRI was developed focusing only on the reuse of analysis patterns [19]. Later, Ribeiro et al. [17] 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.

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 (Figure 2). 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 (Figure 3).

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.

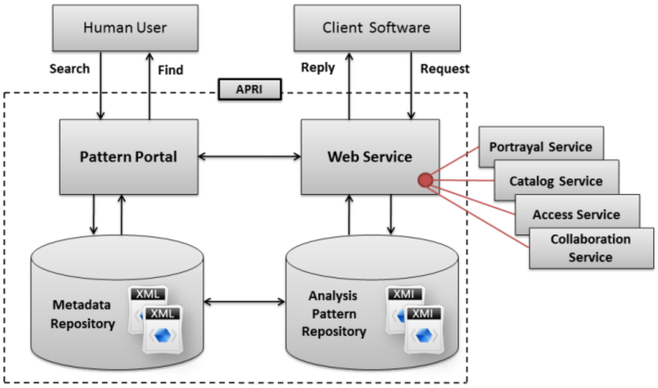

Figure 2. APRI architecture

A. 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) [16] 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 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 [11]. 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[1], 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.

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

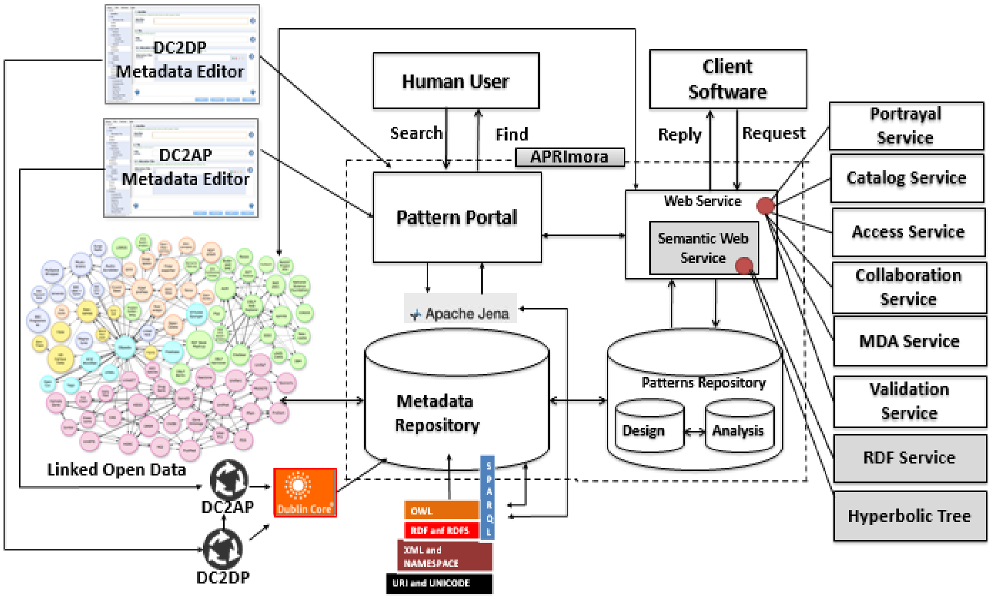

Figure 3. APRImora architecture

B. *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 (Figure 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 [13].

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 is used to uniquely name the patterns and relate them to other patterns. The field Identifier is responsible for receiving the URI value, indicating the uniqueness of the patterns 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)[2]. 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.

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 [13]. APRImora uses controlled vocabulary and syntax in some elements of the DC2AP and DC2AP. Using the XML layer in APRImora allows the metadata to be defined with the syntax 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 [13]. 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 [13]. 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. SparQL provides protocols to query and handle RDF graphs, thus allowing several inferences [13]. This language is used perpendicularly to the RDF and OWL layers. In APRImora, the information is retrieved from repositories using the SparQL in the ARQ module belonging to the Apache Jena framework.

C. *Linked Open Data*

The patterns stored on the Metadata Repository are described using the Linked Data, generating a cloud among the patterns in this repository (Figure 4). The light-gray nodes indicate design patterns and the dark-gray nodes illustrate analysis patterns. This

[1] https://jena.apache.org/getting_started/index.html
[2] http://purl.oclc.org/docs/index.html

Linked Data cloud represents the connections made through the URI among the design and analysis patterns.

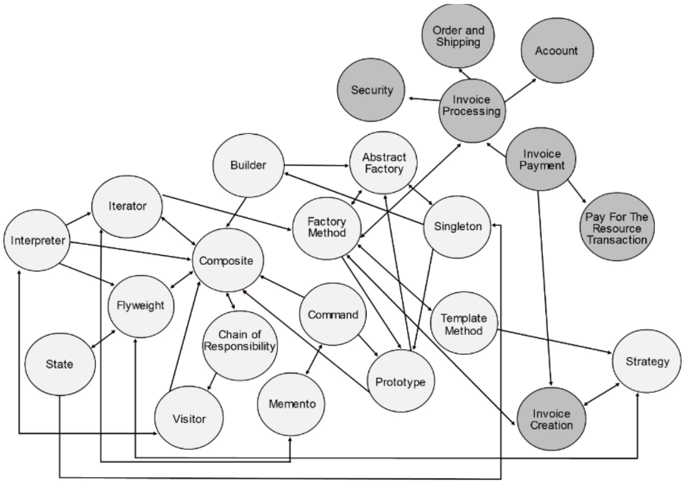

Figure 4. Linked Data cloud of part of the APRImora metadata repository.

The connections presented in Figure 4 indicate a pattern is related with others. Some patterns can be used as a set, e.g., Composite is constantly used along with Iterator or Visitor [8]. 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 situation it can be employed. These multiple ways allow to choose the best way to use the pattern and those that complement it, enhancing designing with patterns reuse.

Linked Open Data is a worldwide data publication effort, making them open and available to be used anywhere [23]. In order for these data to be published and connected, Linked Data must be used. Thus, it 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.

D. 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. In APRImora, the Pattern Repository is made up of two repositories, Design and Analysis, which relate to each other through Linked Data. So, 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.

E. 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.

As seen in Figure 3, 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, is used by the Pattern Portal to generate an RDF document (RDF Service) and the Hyperbolic Tree.

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 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. This is possible thanks to the use of Linked Data.

Figure 5 illustrates a hyperbolic tree corresponding to the connections of possible design patterns existing in the pattern repository, whose root node is the pattern Prototype. Using a hyperbolic tree is possible to visually navigate among the connections between the patterns.

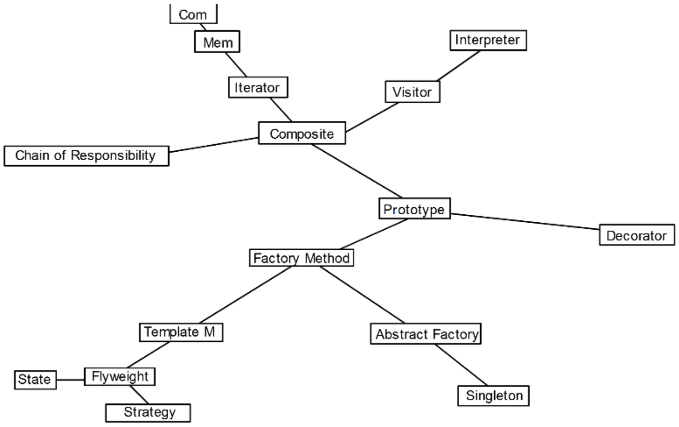

Figure 5. Hyperbolic tree with design patterns

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 – support the visualization of the diagrams of solutions proposed by the patterns;
- Catalog Service – allows the patterns and services to be discovered and used based on their describing metadata;
- Access Service – allow accessing and downloading the patterns;
- Collaboration Service – allow designers and developers to share their use experiences to improve the patterns;

- MDA Service – allows source code to be generated from diagrams;
- Validation Service – allows metadata to be validated.

The Semantic Web Service component makes the following services available:

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

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 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 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.

V. 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 to help developers search for patterns. The analysis and design patterns contained in the APRI are documented using the DC2AP and DC2DP application profiles, 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.

To help the user better retrieve patterns contained in its repositories, the APRI was extended in this paper to an architecture based on the Semantic Web, 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.

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

Acknowledgment

This work was financed by FAPEMIG, CNPq and CAPES.

References

[1] T. Berners-Lee, J. Hendler, and O. Lassila, “The semantic web,” Scientific American, vol. 284, n. 5. pp. 28-37, 2001.

[2] T. Berners-Lee, “Semantic Web Concepts”. 2005. [Online]. Available: http://www.w3.org/2005/Talks/0517-boit-tbl

[3] T. Berners-Lee, “Linked Data”. 2006. [Online]. Available: http://www.w3.org/DesignIssues/LinkedData.html

[4] C. Bizer, T. Heath, and T. Berners-Lee, “Linked Data – the story so far,” Int. Journal on Semantic Web and Information Systems, vol. 5 n. 3. pp. 01-22, 2009.

[5] D. R. B. Cunha, B. F. Lóscio, and D. Souza, “Linked Data: da Web de Documentos para a Web de Dados,” in Livro Texto dos Minicursos ERCEMAPI, A. M. Santana et al., SBC: Teresina, BR, 2011, pp. 79-99.

[6] DCMI - Dublin Core Metadata Initiative, “Dublin Core Metadata Element Set, Version 1.1: Reference Description”. 1999. [Online]. Available: http://www.dublincore.org/documents/1999/07/

[7] M. Fowler, Analysis Patterns: reusable object models. Addison-Wesley Publishing, 1997.

[8] E. Gamma, R. et al., Design patterns: elements of reusable object-oriented software. Addison-Wesley Publishing, 1994.

[9] S. Hawke, “SPARQL Query Results XML Format (Second Edition)”. 2013. [Online]. Available: http://www.w3.org/TR/2013/REC-rdf-sparql-XMLres-20130321/

[10] G. Klyne, J.J. Carroll, and B. McBride, “RDF 1.1 Concepts and Abstract Syntax”. 2014. [Online]. Available: http://www.w3.org/TR/rdf11-concepts/

[11] N. Y. Kobashi, “Vocabulário controlado: Estrutura e Utilização”. 2008. [Online]. Available: http://www2.enap.gov.br/rede_escolas/arquivos/vocabulario_controlado.pdf

[12] D. Martin, and J. Domingue, “Semantic Web Service, Part 1,” IEEE Intelligent Systems, vol. 22, n. 5. pp. 12-16, 2007.

[13] F. S. Monteiro, “Web semântica e repositórios digitais educacionais na área de saúde: uma modelagem com foco no objetivo de aprendizagem para refinar resultados de busca,” unpublished.

[14] D. D. Nebert, “Developing spatial data infrastructures: the SDI cookbook”. 2004. [Online]. Available: http://www.gsdi.org/docs2004/Cookbook/cookbookV2.0.pdf.

[15] V. F. Pamplona, “Web Services: Construindo, disponibilizando e acessando Web Services via J2SE e J2ME”. 2010. [Online]. Available: http://javafree.uol.com.br/artigo/871485/Web-Services-Construindo-disponibilizando-e-acessando-Web-Services-via-J2SE-e-J2ME.html

[16] D. A. Peixoto, L. F. M. Vegi, and J. Lisboa-Filho, “DC2AP Metadata Editor: A metadata editor for an Analysis Patterns Reuse Infrastructure,” Proc. of the CAiSE’13 Forum. pp. 138-145, 2013.

[17] A. A. A. Ribeiro, J. Lisboa-Filho, L. F. M. Vegi, and A. P. Oliveira, “DC2DP: a Dublin Core Application Profile to Design Patterns,” Proc. of the 16th ICEIS. pp. 209-215, 2014.

[18] L. F. M. Vegi, J. Lisboa Filho, G. L. S. Costa, A. P. Oliveira, and J. L. Braga, “DC2AP: A Dublin Core Application Profile to Analysis Patterns,” Proc. of the 24th Int. Conf. on Software Engineering and Knowledge Engineering (SEKE). pp. 511-516, 2012.

[19] L. F. M. Vegi, J. Lisboa-Filho, and J. Crompvoets, “A machine-processable Dublin Core application profile for analysis patterns to provide linked data,” Proc. of the Int. Conf. on Dublin Core and Metadata. pp.23-32, 2012.

[20] L. F. M. Vegi, D. A. Peixoto, L. S. Soares, J. Lisboa-Filho, and A. P. Oliveira, “An infrastructure oriented for cataloging services and reuse of Analysis Patterns,” Proc. of BPM 2011 Workshops, pp. 338 – 343, 2012.

[21] L. F. M. Vegi, J. Lisboa-Filho, J. Crompvoets, L. S. Soares, and J. L. Braga, “A Dublin Core application profile for documenting analysis patterns in a reuse infrastructure,” IJMSO, vol.8, n. 4. pp.267-281, 2013.

[22] W3C, “Semantic Web”. [Online]. Available: http://www.w3.org/standards/semanticweb/

[23] F. H. Zaidan, “LOD – Linked Open Data – Dados Abertos Vinculados”. 2013. [Online]. Available: http://itweb.com.br/blogs/lod-linked-open-data-dados-abertos-vinculados/