SemSearch 2008, CEUR Workshop Proceedings, ISSN 1613-0073, online at CEUR-WS.org/Vol-334/

Search, Natural Language Generation and Record Display Configuration: Research Directions Stemming From a Digital Library Application Development Experience (Discussion Paper)

Andrew Russell Green and José Antonio Villarreal Martínez

Instituto Mora (National Council for Science and Technology, Mexico)
and Instituto de Investigaciones Estéticas (National Autonomous University of Mexico)
ahg@servidor.unam.mx,quetzal1910@gmail.com

Abstract. Digital libraries and archives stand to benefit greatly from the Semantic Web (SW), which may provide a basis for novel end-user functions targeted at research and teaching. The project "Image Preservation, Information Systems, Access and Research" seeks to develop an adaptable digital library application based on a back-end of semantically modeled data. By "adaptable" we mean able to adapt to diverse library and archive scenarios, especially involving the integration of different types of material (photographic prints, negatives, drawings, periodicals, books, etc.) in a single system. A problem we have encountered is: the design of algorithms for processing information as it moves from the model to the user interface, and, following user input, from the interface back into the model. In this paper we discuss two specific issues that are encompassed by this general problem: full-text search mechanisms and record display configuration.

Key words: Semantic Web search, record display configuration, natural language generation, digital libraries

1 Introduction

Digital libraries and archives stand to benefit greatly from the Semantic Web (SW). Semantically modeled catalogues should provide a basis for new functions to help users sift through large and diverse repositories, discover patterns, explore associations among objects, find relevant information, and create and share descriptions of objects in a structured, flexible manner. This is the promise the SW holds for knowledge repositories, and one can hardly underestimate its potential impact in History and other Social Sciences: archives are primary sources—essential deposits of partially processed information, used for research in these disciplines—and despite the high degree of interrelation among data in different

archives, catalogues are often isolated and employ divergent record formats that are hard to align using standard information technology.[1]

This issue is one of the reasons the project Image Preservation, Information Systems, Access and Research (IPISAR) set out to build a SW-based digital library application. The project investigates the dissemination, study and management of heritage resources, and attempts to provide solutions to common problems in these areas.

The application being built, called "Pescador", will store catalogue data in a persistent triple store (whose function will be similar to that of a relational database in traditional systems). The requirements for the application include the ability to integrate data in various catalogue formats and adapt to the cataloguing needs of diverse archives. In this paper, the terms "catalogue format" and "record format" refer to the selection, organization and meaning of fields used to describe objects in an archive or library catalogue, as well as other conventions related to catalogue creation. Since Pescador will use the SW to model catalogues, each record format will correspond to a distinct kind of graph structure, often requiring specialized vocabulary and rules, and related to specialized application logic.

The application will have three main types of user: (1) regular users (or "patrons") who will consult the material provided by the digital library, (2) cataloguers, who will provide and manage the library's materials and metadata, and (3) catalogue designers/modelers/programmers, who will select or create the catalogue record formats and corresponding ontologies, and adapt the system to the needs of a given scenario. Pescador will provide a Web interface for the first two kinds of users; on this level, numerous functions targeted at research, teaching and cataloguing are planned [5]. When these users view data from the catalogue, they will see a user-friendly organization of information extracted from the SW graph; similarly, when cataloguers modify elements in the catalogue, they will employ easy-to-use forms, and the SW graph will be changed according to their input. The third type of user, the catalogue designer/modeler/programmer, will use a programming interface.

A problem we have encountered is: the design of algorithms for processing information as it moves from the model to the user interface, and, following user input, from the interface back into the model. In this paper we discuss two specific issues that are encompassed by this general problem: full-text search mechanisms and record display configuration. We conclude that record display, natural language generation and other text generation logic, text fragment caching mechanisms, and full-text search algorithms must be studied and designed together.

To date, two incomplete versions Pescador have been created. Both are currently used for Web sites that offer simple consultation functions for on-line archives (available at [8] and [3]). Our proposals stem from the experience of developing these versions of the application. Though the project IPISAR may

[1] The situation of historical archives varies greatly from one archive to another. Other recurring difficulties include access restrictions and insufficient funding; the first of these is also a major focus of the project described in this article. See [6] and [1].

yet generate new archival Web sites using the second version, it is clear that to implement all proposed features, a major rewrite is unavoidable. It should be noted that work on the rewrite has yet to begin. The general nature of the proposals outlined here is a reflection of this.

All versions of Pescador are provided under the terms of the free GNU GPL license.

2 Display Templates

There exist several general systems for record display specification, which we will call "display template systems", and many SW applications use internal template mechanisms. We agree with the definition of the problem given by the authors of Fresnel (an important proposal in this area), who state that "presenting Semantic Web content in a human-readable way consists in addressing two issues: specifying what information contained in an RDF graph should be presented and how this information should be presented." [2] However, this definition is deceptively simple, as both parts of the problem—the selection of information from the model and its transformation into a presentable format—can be quite complex.

Clearly there is a need for templates in SW applications: models often do not contain all the information required to create user-friendly descriptions, and even when they do, it is not always be desirable to show users all available information. The most basic kind of SW template involves a selection and ordering of properties; when the template is "applied" to a resource, label-value pairs are created from the properties' labels (often set using `rdfs:label`) and values for that resource. On this foundation, numerous advanced features may be built, such as:

- Facilities for creating sections, subsections and similar structures within records. This is often required for lengthy description; see, for example, full records in [3] and [8].
- Ways of including additional elements in records, such as images and text that is not part of a label-value pair.
- Facilities for defining short, human-readable labels for resources—normally used for values in label-value pairs, to refer to resources that are the objects of the properties displayed.
- Ways of setting special, context-appropriate property labels. (Consider, for example, a property with the `rdfs:label` "Location Photographed". In a photograph's catalogue record, one might wish to call the property "Location", since in this context, the full label would be needlessly long.)
- Means of embedding the result of one template in the result of another one.
- Means of retrieving information from diverse parts of the model—not just over the direct properties of the resource being described. This may be accomplished using path definitions.
- A hierarchy of templates and inheritance of templates' characteristics over the hierarchy.

- Media-agnostic template definitions, or a separation of record content specifications from media-specific formatting details.
- Facilities for embedding arbitrary logic—in other words, executable code—in templates, in a manner similar to languages for creating dynamic Web pages (JSP, ASP, PHP, RHTML, etc.). This allows templates to run loops, generate text and modify their output on the basis of conditions described in the executable code.
- Programmatic template creation and modification. For example, at runtime, a search component may create temporary templates that display only fields containing hits.
- Vocabulary and conventions for modeling the templates themselves.

Fresnel, Pescador 0.1 and Pescador 0.2 all implement different subsets of these possible features. A challenge for the next version of Pescador is to determine which features are required, and how to integrate them with our system while maintaining support for encapsulation and separation of concerns. In additional, we must take into account a lesson learned in work on Pescador 0.2, namely: that the scope of a templating system is wider than record display itself. This is because numerous elements of a user interface must be coordinated with record display. To illustrate this, let us consider a catalogue in which photographs are described with the fields "photographer", "title", "date", "location" and "topics". A user interface that provides access to such a catalogue would refer to these fields in several places, not just when displaying records. For example, a menu might offer the option of listing items ordered by date or title. Another might offer the possibility of grouping items by photographer, location or topic. An advanced search interface could include options for searching only within one or more of these fields. On the screen displaying the catalogue records themselves, diverse functions may be available in fields' context menus. Last but not least, the interface for adding, deleting and modifying items in the catalogue will mention fields in various ways. In all these parts of the interface, references to fields must be consistent, clear and appropriate for their respective contexts. To achieve this, specialized display specification mechanisms are required, and it makes sense to integrate these mechanisms with the template system.

3 Natural Language Generation and Searching

In this section we review two related issues: natural language generation (NLG) and full-text search. Like display templates, these problems fall under the broad category of algorithms for processing information as it moves back and forth between the model and the user interface.

NLG is "the subfield of artificial intelligence and computational linguistics that focuses on computer systems that can produce understandable texts in English or other human languages" [9]. Typically SW applications have not used complex, human language-aware subsystems to create text output, opting instead for simpler mechanisms that extract strings from the model and place

them in slots established by an interface generation subsystem (which may use a display template mechanism, as described above). Though in many cases this is sufficient, in developing Pescador we have come across several scenarios that call for a more elaborate language generation mechanism, able to create, from subgraphs, understandable fragments of natural language, taking into account language features such as pluralization, gender and the chaining of adjective phrases.

We demonstrate the problem with the hypothetical results of a full-text search in a mixed archive (Fig. 1). In this mock-up, items are grouped according to their relationship to nodes that produced full-text hits. Of course, these relationships would exist as paths in the graph. Only an NLG system would be able to produce concise, correct and easy-to-read descriptions of associations such as those shown in the mock-up. (For many languages the generation of descriptions like these is more complicated than it is for English—an example is Spanish, in which adjectives must agree both in number and gender with the nouns they describe.) In other processes that might employ NLG, its potential benefits are similar though perhaps less notorious.

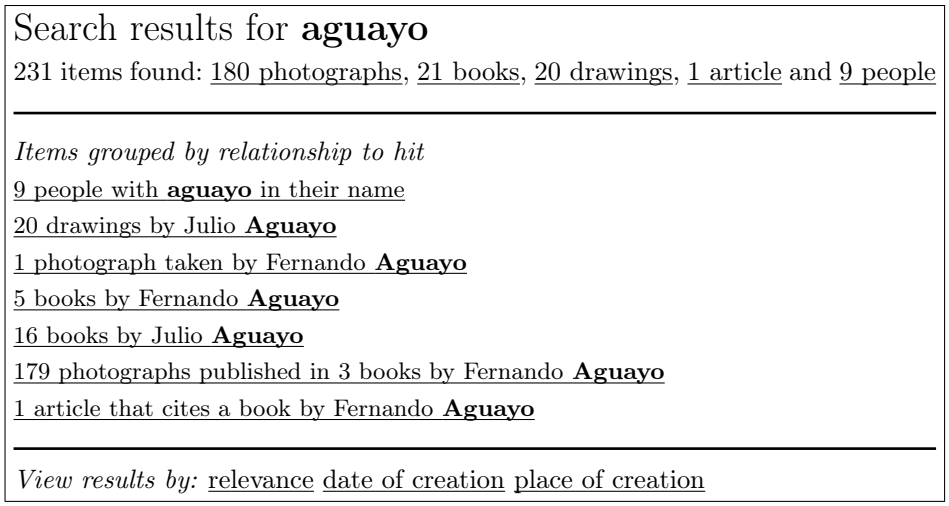

Fig. 1. Mock-Up of Search Results
Underlined elements are hyperlinks.

Thus we identify NLG processes—including, but not limited to, the translation of paths into natural language descriptions—as an issue to be studied for the next version of Pescador. In general, we view NLG as a low-level process in user interface generation, since it creates small text fragments—as opposed to a display template system, which operates at a higher level, organizing much larger segments of the interface. Note that at this low level of text fragment generation, the NLG system, once implemented, will not be alone; many text

fragments will still be easier to create using more traditional sorts of text concatenation logic (for example, a string with a person's family names and given names). Note also that we can distinguish two types of text fragment generation processes: (1) those that create transient text fragments, not cacheable for later reuse (for example, the relationship descriptions in Fig. 1); and (2) those that generate more stable fragments, which might be retained in a cache and inserted repeatedly into the user interface.

We mention the distinction between transient and cacheable text fragments because, to explain the issues we are facing in full-text searching, we must first review the functioning of these stable, cacheable fragments. In Pescador 0.2, cached text fragments are mainly low-level building blocks of catalogue records. That version of the system caches them not only to speed record generation, but also to allow full-text search within them. This is important because of the way users expect full-text search to work. To illustrate briefly: suppose that a model contains resources that refer to people, and that those resources may have three properties: `hasFamilyNames` and `hasGivenNames`, which point to literals, and `hasTitle`, which points to resources that represent titles. The resources for titles, in turn, have two properties: `hasAbbreviation` and `hasFullName`. Human-readable labels for people are constructed with literals; for example, the label "Smith, Dr. George" would aggregate literals that represent the abbreviation for "doctor" and Dr. Smith's family and given names. In a full-text search, to correctly locate resources associated with "Smith", "George" or "Dr", the search component would need only look at strings contained in the model. But what if a user searches for the *exact phrase*, "Smith Dr George"—that is to say, those words together, in precisely that order? Nowhere in the model do they appear in that manner, but the end user does not know that, and if s/he has seen such text fragments in the catalogue, s/he will expect such a search to produce results. A possible solution is for the system to cache this text fragment, make it available to the search component, and associate it with the resource that refers to Dr. Smith; then searches may find that phrase and return a hit on the correct resource.

Despite the apparent feasibility of such a mechanism, there are unsolved issues related to how cached, generated text fragments may be treated in searches. In the preceding example, clearly it is fine for searches for "Smith", "George" or "Smith, Dr. George" to located the generated fragment and thus return Dr. Smith as a search result. But searches for "Dr" should not do the same—rather than finding all the generated text fragments that include that string, they should provide more meaningful results, for example "**Dr**, abbreviation for the title 'doctor', borne by 20 people in the knowledge base". The precise algorithm needed here remains to be flushed out.

4 Conclusion

In this paper we have considered two issues related to the algorithms for processing information as it flows between the model and the user interface: record display templates and full-text search algorithms. We conclude that template mech-

anisms, NLG and other text generation logic, text fragment caching mechanisms, and full-text search algorithms must be studied and designed together, in order to construct friendly, easy-to-understand and meaningful catalogue records, interfaces and search results. This is in part because most users will not have technical knowledge of the SW, or of how data is modelled and transformed to construct catalogue records, and they will expect searches to be performed on records as displayed. As a result, the application's search component must consider these same data transformation algorithms when it moves from hits in the SW graph to user-consumible seach results. This realization provides starting points for further work towards the creation of the application we envisage.

References

1. Aguayo, F., Roca, L.: Estudio introductorio. In: Aguayo, F., Roca, L. (eds.): Imágenes e investigación social. Instituto Mora, México (2005) 9-28 http://durito.nongnu.org/docs/Aguayo_Roca_2.html
2. Bizer, C., Lee, R., Pietriga, E.: Fresnel Display Vocabulary for RDF: User's Manual. World Wide Web Consortium (2005) http://www.w3.org/2005/04/fresnel-info/manual-20050726/
3. Fototeca Digital: Fotógrafos y Editores Franceses en México. Siglo XIX. Instituto Mora and Instituto de Investigaciones Estéticas, National Autónomous University of Mexico (2007) http://afmt.esteticas.unam.mx
4. Green, A.: Logic and a Little Language for Heritage Resource on the Semantic Web. Poster accompanying a system demonstration, presented at the 4th European Semantic Web Conference (June, 2007) http://durito.nongnu.org/docs/innsbruck2.pdf
5. Green, A. R.: Metadatos transformados: Archivos digitales, la Web Semántica y el nuevo paradigma de la catalogación. In: Amador C., P., Robledano A., J., Ruiz F., R. (eds): Quintas Jornadas: Imagen, Cultura y Tecnología. Universidad Carlos III de Madrid: Madrid (2007) 11-22 http://durito.nongnu.org/docs/metadatos_transformados_green.pdf
6. Green, A. R.: Rescate de la memoria. Ciencia y Desarrollo (Sept. 2006). Consejo Nacional de Ciencia y Tecnología, Mexico
7. Kochut, K. and Janik, M., SPARQLeR: Extended Sparql for Semantic Association Discovery (2007) http://www.eswc2007.org/pdf/eswc07-kochut.pdf
8. Marcas de Fuego de la Biblioteca "José María Lafragua" de la BUAP. Autonomous University of Puebla (2006) http://www.marcasdefuego.buap.mx/
9. Reiter, E., Dale, R.: Building Natural Language Generation Systems. Cambridge University Press: Cambridge, UK (2000)