Document Souls: Joining Personalities to Documents to produce pro-active documents engaged in contextualized, independent search

Gregory Grefenstette[1] and James G. Shanahan[2],

[1] CEA/LIST/DTSI/SCRI/LIC2M
Commissariat à l'Energie Atomique, Centre de Fontenay-aux-Roses, B.P. 6
92265 Fontenay-aux-Roses Cedex, France
Gregory.Grefenstette@cea.fr
http://www.cea.fr
[2] Clairvoyance Corporation, 5001 Baum Boulevard, Suite 700
Pittsburgh, PA 15213-1854 USA
jimi@clairvoyancecorp.com
http://www.clairvoyancecorp.com

Abstract. The idea behind the semantic web is that documents will contain additional markup that make explicit the information content of unstructured media. We present here the Document Souls system which allows documents to become animate, actively searching the wider world for more information about their own contents, attaching relevant information to itself as additional markup. A Document Soul is a set of information requests that are attached to a document as annotation. A demon within the system polls these requests and activates search agents that asynchronously respond to unsatisfied requests. To control search and relevance, collections of information requests are packaged as personalities which filter out unwanted information. In this paper, we present the structure of the Document Souls architecture and the function of personalities for performing contextualized search.

1 Introduction

Growing frustration with the string based indexing techniques now available for searching the Web has lead to a demand for a Semantic Web [1] in which documents would contain, in addition to their raw human-readable content, extra annotation that makes certain information in the document explicit in a standard format [2]. Most such annotation systems have supposed that this annotation in added by the authors of the document, maybe semi-automatically [3], or by reinterpreting structure explicit in the page [4], or by filtering the document content through an existing structure such as an ontology [5]. All these annotation approaches view the document as a static entity that is transformed by the process of an outside agent into another static entity.

An alternative view is to consider a document as an active participant in its own annotation. If we consider that the relation between a document's contents and the

information space represented by the WWW and the Grid is ever changing, then it is necessary to have a mechanism for continually creating new annotations and modifying old annotations associated with a document. We decided to center this updating processing within the original document and thus designed the Document Soul[3] system that we present here[4].

2 Scenarios

Before describing the Document Souls architecture, we begin with some scenarios that describe its general utility.

- It's late in the evening and you are preparing to leave the office. A last e-mail arrives from your boss calls with an attached document to be worked on. You now have three options: hang up your coat and start to work on the document, send the document to your home account and work on it overnight, or pretend you didn't read the message until the next day. A fourth option is to add a Document Soul to the document. The document then works on itself overnight identifying people, products, finding their homepages[5] and descriptions, preparing your work for the next day.
- A part of the intellectual property of a company resides in the patents that it controls. These documents currently lie dormant until someone familiar with them comes across some instance of possible infringement, and decides to call up the patent and compare it to the litigious case. Adding Document Souls to these patents can transform them into active agents scouring corporate websites for new products that relate to the patent. When the augmented patent finds something interesting, it alerts the legal team with the summary of the possible infringement, the incriminating website, and a copy of itself in the lawyers' mailboxes.
- You find an interesting description of site you think you might like to visit. You attach a Document Soul to the page, and supplementary information that would allow you to plan a trip (train times, maps, nearby sites, photos) are searched for by the augmented document. The next day, you reopen the document and find this extra annotation that eases your decision making.

It is evident that recent advances in web-based information extraction technology [6][7] have made any of these scenarios feasible. What is not evident is that a general information and search tool can perform such a wide variety of distinct tasks without

[3] The imagery associated with the would 'soul' involves the quickening of inert matter.

[4] The work presented here was performed while both authors were on the staff of the Xerox Research Centre Europe (Meylan, France; www.xrce.xerox.com). "Document Souls" is a registered trademark of Xerox. The intellectual property of the system described here is covered by US patents 6868411, 6778979, 6732090, and other patents pending

[5] Accessing perhaps an information aggregator such as ZoomInfo which has been scanning the web, collating information about people and companies in the US.

creating an unmanageable amount of irrelevant noise. This observation led us to define Personalities for the Document Souls system. In the next sections we will describe the architecture of this Document Souls system for contextualized, independent search.

3. Document Souls Architecture

The architecture of a Document Souls system is the following :

- A collection of Document Soul endowed documents
- A demon that polls Document Soul endowed documents at regular intervals to treat any unsatisfied requests
- A personality which is a packaged collection of domain-specific search requests
- Software agents which can satisfy a request and generate new requests

We now explain these components with some examples.

3.1 Endowing and polling a document

A document becomes endowed with a Document Souls once it is annotated with an unsatisfied search request. An example of a search request is FindPeopleHere. This request is appended to the original document as additional XML markup, such as <ds:request=FindPeopleHere scope=wholedoc/>, in which 'ds' refers to the Document Soul namespace. Once such a request appears in document, the document is endowed. When the demon accesses this document, it removes the request from the document will activate a software agent associated with "FindPeopleHere", passing along the contents of the documents to the agent. The agent will identify people [8] in the document. Identified names will be repackaged as XML markup and added to the original document to be made available to other requests, for example as <ds:FoundPeople people="John Smith, Tom Jones">. If the document contains the additional request <ds:request=FindHomePages scope=FoundPeople>, this subsequent request will be sent, the next time the demon polls the document, to a different software agent that finds an returns homepages[6] for each of the named entities.

In addition to such simple requests for additional annotations, some requests can generate both additional annotation and additional requests. For example, the request <ds:request="FindStockPrice" code=XRX time= "17:30EST"> will be sent to an agent that will which will return both the stock quote at the end of the market as well as an additional request similar to the original that will be activated by the demon

[6] For example, such a service exists for computer scientists at http://hpsearch.uni-trier.de/

watching over the endowed document the next day. Responding to one request can generate a number of additional requests.

3.2 Software Search Agents

A software agent is any independently callable software module that accepts information passed to it by the demon, processes this information, and possibly returns annotation and/or new requests that the Document Soul demon integrates into the original document. The agent may be passed the entire contents of the documents, or only subparts. A typical action of an agent would be to interact with an online data source, such as Medline to search for articles concerning a certain concept. The agent is responsible for preparing the query, negotiating connection protocols (which might involve electronic payments), recovering the results of the information request, and formatting the results for inclusion as additional notation. Creating such agents currently involves wrapping [9] search engines but they will become easier to construct once Semantic Web Services [10] become more prevalent. An agent may also work locally, preparing work for some other agent. For example, one agent may extract all medical terms from a document, and isolate them in a separate annotation; another agent may take the results of this extraction and formulate three-by-three combinations of these medical concepts that are added into the document as requests for searches on a medical source such as MedLine (e.g. <ds:request=MedLineSearch content="cancer AND radiofrequency ablation AND tumor"); and a series of other agents may run each query independently marshalling their results to return to the Document Souls demon.

3.2 Personalities

In order to focus the requests that are applied to a document, we collect Document Soul requests into a personality, a packaged set of information requests (and their associated software agents). Here are some examples of personalities that we can define, with the list of information requests that they contain:

- Tech Watch personality:
    - identify industrial concepts and products in the document; identify names of companies and individuals; build an organizational chart for each company; find the company; find competitors for each named company; find tutorials on the concepts; find white papers; access stock histories of companies; find out who they are hiring; find press releases and business reports; find conferences in which the concepts appear in the call-for-paper or program; search for patents owned by the people and companies above; find addresses of home offices; branch offices; get map to companies; build patent database around concepts in the document; collect URLS mentioning companies or products; identify other products offered by the company; get weather reports for the home office;

- Patent Attorney Personality
    - find other patents by same inventors; find other patents with same International Patent Code; find all patents which reference this one; identify concepts from Description; identify concepts from Claims; find other patents with same concepts; find home pages of all the inventors; find home pages of Assignee; find any references to published papers by the inventors; try to identify any products associated with assignee, inventors, and concepts; find other papers written by authors together, separately;
- Scientific Personality
    - identify concepts; identify central subject domain; produce summaries (quantitative, undirected, directed); find online versions of itself; isolate its bibliography; find home pages of all the authors; find home pages of all cited authors; find online versions of cited papers; find tutorials about concepts; find conferences in which the concepts in the document are talked about; research patents on topic; find/create BibTex version of citation; find other papers written by authors together, separately; find papers with related concepts by other authors
- Fiction Reader Personality
    - identify character names; link character names to appearances in book; find place names; find time period; link place names to maps on the WWW; link place names to photos; find street names; map street names

Each one of the individual tasks listed within the above personalities can be automated with current natural language processing and search technology. Some tasks can be useful in more than one personality. For example, the task "find other papers written by authors together, separately" which would take a list of authors and search through repositories such as CiteSeer or the Computer Science Bibliography at http://dblp.uni-trier.de, and return articles by members of the list, is listed as being part of the "Scientific Personality" and as being part of the "Patent Attorney Personality."

Applying different personalities to the same text will produce different sets of annotations. An example of this could be endowing a call-for-papers with a scientific personality during the paper preparation phase, and then taking the same original document and endowing it with a "travel agent" personality that would search for train connections and hotel possibilities once the paper was accepted.

The results of endowing a document with one of these personalities can result in a tremendous amount of annotation, even though the information retrieved has been restricted by the personality's limitations on what sources are searched and how these sources are filtered by the task-centric search agent. Since the annotation is appended in XML format, the information can be hidden from the user of the document until requested, just as hyperlinks hide a vest network of related information [11]. A style

sheet for a given personality can present the information, for example, in a hierarchical and hopefully intuitive fashion. Figure 1 taken from the Xerox Document Souls fact sheet[7] presents an interface that shows both the selection of a personality to be applied to a web-based document (in the upper left-hand frame) as well as the results of a personality-oriented annotation (here a BiologyWatch personality). The results of the annotations are presented both as pop-windows connected to highlighted words, and as a separate frame (lower left) that provides a hierarchical structure into the retrieved information.

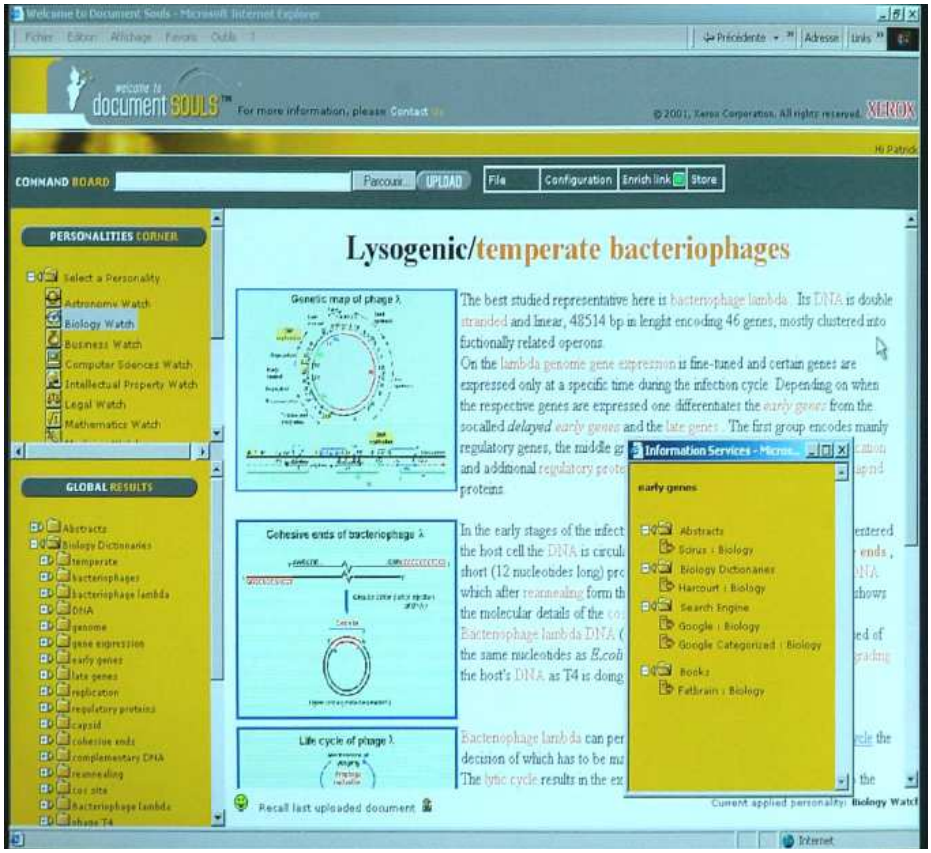

Figure 1. This screen shot shows an example of a Document Souls interface for a Biology Watch Personality (from the Document Souls Information sheet published at http://www.xrce.xerox.com/showroom/pdf/docsouls.pdf)

Composing a new personality involves choosing a new combination of information requests with their associated task-centric search agents. The details involved in build-

[7] http://www.xrce.xerox.com/programs/kc/docsouls.html

ing a new personality are described in the U.S. patent 6,732,090 that can be downloaded at www.uspto.gov. A task-centric search agent can be a simple wrapper [12] if the data available on the web site is already centered on a specific domain, or it can involve both a search over a general data repository followed by task-specific filter [13]. In the current implementation, each search agent is handcrafted. A personality is then any collection of such search agents.

4. Related Systems and Conclusion

The idea of having anticipatory information systems is not new. Watson [14] , for example, from the InfoLab at the University of Northwestern, was a program which operated while a user created a document. Watson retrieved information from third-party service providers as the user worked, information from which the user could select for further investigation. Autonomy proposed a system called ActiveKnowledge which could analyze documents as they were being prepared on the user's computer desktop and then provide links to relevant information. A number of systems exist for adding automatic annotation to text from Microsoft's ActiveTags to ClearForests Tags system which identify certain classes of entities and connect them automatically to additional online information. The Vividocs system [15] included a document analysis system that could stay active and periodically bring in new information from the Web or Grid. The principal novelty of Document Souls resides in the idea of personalities that create the context that focuses and restricts the search activities generated by a document.

Document Souls has been developed as an advanced prototype by the Xerox Research Centre Europe. Contact docsouls-techsupport@xrce.xerox.com for more information about the current state of Document Souls.

References

1. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American, vol. 284:5 (2001) 34–43
2. Simons, G.F., Lewis, W.D., Farrar, S.O., Langendoen, D.T., Fitzsimons, B., Gonzalez, H.: The semantics of markup: Mapping legacy markup schemas to a common semantics. Proceedings of the 4th workshop on NLP and XML (NLPXML-2004). Barcelona, Spain. (2004) 25 -- 32.
3. Adelberg, B.: NoDoSE---a tool for semiautomatically extracting structured and semistructured data from text documents. SIGMOD Record, 27 (1998) 283--294
4. Arasu, A., Garcia-Molina, H.: Extracting structured data from web pages. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data (2003)

5. Hammack,C, Scott,S.: LASSO: A Learning Architecture for Semantic Web Ontologies. In Proceedings of The 2004 International Conference on Machine Learning and Applications (ICMLA '04) Louisville, Kentucky, December (2004) 10-17
6. Ciravegna, F., Chapman, S., Dingli,A., Wilks, Y.: Learning to Harvest Information for the Semantic Web. Proceedings of the 1st European Semantic Web Symposium, Heraklion, Greece, May 10-12, (2004)
7. Petinot, Y., Teregowda, P.B., Han, Hui, Giles, C. L., Lawrence, S., Rangaswamy, A., Pal, N.: eBizSearch: a Niche Search Engine for e-Business. In Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2003), Toronto, July-August (2003) 413-414
8. Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In Proceedings of the Sixth Workshop on Very Large Corpora, Montreal, Canada (1998) 152—160
9. Bergholz, A., Boris Chidlovskii, B.: Learning query languages of Web interfaces. SAC (2004) 1114-1121
10. McIlraith, S., Son, T., Zeng: Semantic Web Services. IEEE Intelligent Systems. Special Issue on the Semantic Web 16. March/April (2001) 46-53
11. Brush, D. Bargeron, A. Gupta, JJ Cadiz. Robust annotation positioning in digital documents. Proceedings of CHI 2001, (2001) 285-292
12. Bergholz, A., Chidlovskii, B. Learning query languages of Web interfaces. Proceedings of SAC 2004 (2004) 1114-1121
13. Boley, D., Gini, M., Gross, R., Han, E.-H. S., Hastings, K., Karypis, G., Kumar, V., Mobasher, B., , and Moor, J. Document categorization and query generation on the world wide web using webace. AI Review, 13(5-6) (1999) 365--391
14. Budzik, J., Hammond, K., Birnbaum, L.: Information access in context. Knowledge based systems. 14 (2001) 37-53
15. Evans, D.A., Grefenstette, G., Qu, Y., Shanahan, J.G., Sheftel, V.: Agentized, Contextualized Filters for Information Management. AMKM (2003) 229-244