The CONCUR Framework for Community Maintenance of Curated Resources

Patrick Schmitz
IST, Data Services
University of California, Berkeley
Berkeley, CA, USA
pschmitz@berkeley.edu

ABSTRACT
The increasing use of computational linguistics for semantic search and discovery tools requires much work on development and maintenance of associated ontologies. Related applications depend upon curated resources like dictionaries, gazetteers, etc. In order to scale these application models and leverage the respective communities of interest, a new set of tools is needed that facilitate community development and extension of these resources while retaining the curatorial model to ensure a reliable, high quality resource. We describe the requirements and principles for such a system, and present the CONCUR framework that addresses these needs. CONCUR defines a reputation model and a set of reusable infrastructure services to maintain the resource. The reputation model combines correctness as well as utility of participants' contributions, tracked over time and by sub-domain within the resource. We describe the architectural issues of the model, potential applications, and continuing research on the model.

Categories and Subject Descriptors
H.3.5 [**Online Information Services**]: *Data Sharing*, Web-based services. K.4.3 [**Organizational Impacts**]: *Computer-supported collaborative work*.

General Terms
Design, Standardization, Verification

Keywords
Ontology, Structured Information, Community, Curation, SOA

1. INTRODUCTION
A common problem facing a broad range of enterprises is that of rapidly expanding document corpora and the associated need for mechanisms to manage and to support finding tools for these corpora. A related push in academic and cultural heritage institutions seeks to realize the value of information repositories with increased access and sharing, using information services built upon these resources. A new generation of tools leverages domain knowledge models and other structured resources to improve access and increase productivity of knowledge workers using these resources. At the same time, community production models are changing the way people collaborate to create and share knowledge within business, for academic scholarship and even for personal interest. To date, however, there are no good tools that facilitate the community maintenance of the structured information resources that underlie many of the new information management and access services.

Many commercial enterprises are deploying state-of-the-art information management technologies that support search and discovery within their rapidly expanding document corpora. These approaches are often described as *semantic search* because they go beyond simple keyword matches and model *semantics* (meaning) in the collection using a combination of *ontologies* (knowledge models) and linguistic techniques. Given the size of these corpora, it would be impractical for humans to classify every document, especially since the appropriate categories may change depending on the audience. Semantic search tools address this by automatically associating concepts to each document object.[1] Enterprises have demonstrated the utility of these tools in making "knowledge workers" more productive, and this is driving rapid growth in the associated enterprise search market. Examples include commercial tools from Autonomy, FAST, Convera, and others, as well as open source applications like the Delphi toolkit for museums [15].

Another class of sharing tools, *recommender systems*, commonly leverages domain knowledge modeled in ontologies to improve performance. E.g., the CHIP system [13] provides museum recommender services based upon a rich ontology of information about the collections and art history domain.

While various tools differentiate on certain techniques employed, they all ultimately depend upon ontologies and much of the performance of these systems is tied to the quality of these ontologies. Quality in this sense includes domain coverage, modeling richness (e.g., inference rules), and linguistic richness (synonyms and other support for NLP processing). Some commercial systems include ontology workbenches, but the process of developing and maintaining the ontologies is painstaking, and is constrained by an editorial model that does not accommodate community collaboration.

In addition to ontologies, other structured information resources are also curated to provide authoritative reference services to an information community. For example, many domains within the humanities develop and maintain dictionaries, gazetteers, and name authorities to support scholarship (e.g., the Cuneiform Digital Library Initiative [3]). These reference resources are becoming more

DocEng'08, September 16–19, 2008, São Paulo, Brazil.

[1] A similar process is characterized in some models as *information extraction*, but the distinction is not important here.

important to information management and associated document engineering, and yet are maintained in an ad hoc manner. They are often "owned" and maintained by individuals or small groups who can at best accept suggestions from a community via email or other unstructured messages.

To address these needs, we present the CONCUR architecture for community maintenance of curated resources. This infrastructure will support the maintenance of structured information resources, leveraging a reputation model based upon the correctness and utility of a participant's contributions over time. The next sections lay out the principles and requirements for the system, some related work and a description of the CONCUR architecture.

2. PRINCIPLES AND REQUIREMENTS

The ultimate authority and responsibility for a given knowledge resource rests with the curators, and so the infrastructure must serve their needs and fit in well with their workflow. It must be generally easy to use, but more specifically should allow curators to quickly find and judge the best suggestions from the community, without wading through long lists of naïve suggestions. This means that the performance of contributors must be tracked and used to rank new suggestions (although it should be possible to set an a priori reputation for known contributors). The system must model authority and expertise as a function of area or subject domain, both for the curators as well as for contributors. This ensures that a curator is not asked to consider suggestions outside their domain, and also recognizes that expertise is localized: a high reputation in Chinese ceramics should not confer high reputation to the domain of African textiles.

For contributors in a community, the system must again be easy to use, allowing a user to clearly describe a suggestion in a structured manner, and in the context of the existing resource (e.g., to add a synonym to a concept, to add a new narrower concept under an existing broader concept, or to propose a new inference rule as a function of existing concepts). It should also support timely response to suggestions and a means of seeing a decision trail for suggestions. Some existing projects like the AAT [6] take ad hoc input from users but these are neither particularly responsive nor transparent in their decisions. In order to foster the kind of dynamics that drive successful open-source projects and online communities, the infrastructure should support incentives for contribution and reputation. These may include explicit ranking of contributors, highlighting productive contributors or groups (e.g., those from a given institution), etc. For academic environments in which citations are a key metric, support for provenance of contributions may help justify the time and energy devoted to contributing to a shared resource. An implicit principle in this discussion is the importance of identity in the system – autonomous contribution misses the point. Some systems may allow pseudonymous contributions while others will require or encourage the association to real-world identities (especially to recognize a priori expertise in a domain).

The framework must be flexible enough to support different applications, including the linguistic enrichment of ontologies, the extension of knowledge modeling in ontologies, the addition or revision of items in dictionaries, gazetteers, name authorities, etc. The resource itself must be modeled in the abstract, and it must be possible to define application-specific measures of utility for contributions (e.g., number of documents or objects indexed using a new synonym, amount of end-user activity around a given addition, etc.).

For certain applications like semantic indexing (or information extraction) the tools should support what-if scenarios to determine the impact of a given suggestion before allowing or rejecting it.

3. RELATED WORK

Many systems use automatic mechanisms to enrich certain types of resources, and considerable attention has been devoted to extension and enrichment of ontology. For example, the model in Bhat, et al. [2] uses latent semantic analysis to identify potential synonyms, etc., to enrich a thesaurus. While such approaches might assist contributors to identify suggestions, curated resources only retain their authority if all suggestions are reviewed.

Social tagging systems implicitly support some of the functionality we describe in an unstructured manner. The activity of tag memes (e.g., for conferences, projects, etc.) and ad hoc mechanisms such as the *lat:* and *long:* prefixes on tags that predated structured geo-tagging demonstrate how a community can work together to define models of organization even within a folksonomic model. In [14] we showed that there are implicit ontologies in the data generated by social tagging activity. Wu et al. [18] use another approach and demonstrate similar results. To date, however, these systems do not provide tools to structure the model and to formalize decision-making.

Freebase [4] is a commercial system allowing a community to contribute instance data to an ontology. However, there is no formal mechanism for contributing to the ontology structure, nor does Freebase have a formal structure to support curatorial review.

There are examples of open-source ontology development, demonstrating that communities can produce structured resources like this. The OpenGALEN project [12] is among the oldest open source ontology projects. It follows a collaborative process without any formalization of provenance, authority, etc. Sunstein [16] describes how a deliberative process like this tends to produce worse results than a polling or review model, which argues against the pure open-source model for this kind of application. The Gene Ontology (GO) project [5] leverages a model of curator-groups similar to our own, but has no model for tracking provenance or reputation. The Microarray Gene Expression Data (MGED) ontology project comes the closest to some of the ideas we describe. Their Pronto tool [17] provides a set of web service APIs and client tools to support submission of new terms, or comments on existing terms. However, they do not provide formal support for curation or for provenance or reputation tracking, and they do not abstract the application to support other actions and different resource types.

The CONTEXT/SR platform [1] includes a model to review the results of automated information extraction, but does not include a reputation component. Ignat and Norrie [9] describe a system that supports collaborative editing of XML documents, defining a set of formal operations that change the XML structure. This approach provides a model for making changes to a curated (XML) resource, although their model does not consider curation/review mechanisms.

Early work on TRELLIS [7] describes a trust framework for annotating sources of information. The system allows a user to describe the reliability and credibility of sources of information, distinguishing between general past performance, and the specific context of a given instance. The system then aggregates ratings for sources from a community of users. There is some accommodation for usage of individual items of information, but there seems to be

no generalized metric for utility of the given source of information. In addition, the model has only monolithic measures for each source, and has no means to qualify reliability or utility as a function of information domain.

A similar model is extended to social networks in [8], where users can describe levels of trust in others in their network. This model adds description of a subject area for the trust, but does not discuss how this is used. The SUNNY model [10] refines this model with the use of probabilistic trust associations and a Bayesian Network model for computing trust and confidence in social networks. However, SUNNY does not model focus of trust by domain, and none of these models incorporate a notion of how useful the source is (when found to be correct).

The EigenTrust model defines trust based upon performance in a peer-to-peer network, computed over time and as a function of the network. It also recognizes that many systems will assign a priori trust to certain parties, recognizing some pre-existing reputation or authority. The EigenTrust model does not model utility of contributions, and is quite different in application (it is intended specifically for peer-to-peer networks).

The CKC Challenge [11] echoes some of the requirements we describe (e.g., provenance and reputation tracking), but does not mention utility. As they are issuing a challenge to develop tools, they do not propose any specific solutions.

4. PROPOSED MODEL

The CONCUR model defines a framework and tools to facilitate definition and submission of additions to the curated resource, tools to support review of these suggested additions, and a reputation model that tracks the performance of contributors over time. The framework follows Service-Oriented Architecture (SOA) principles in the definition of shared services, and abstracts the curated resource behind a service contract definition to enable broad re-use in different contexts. The next sections describe the roles and principles of the model.

4.1 Roles in the framework

There are two roles in the model: *contributors* of content and *reviewer/curators* who vet the suggestions. Contributors drive the model, providing the raw material of new suggestions that extend and enrich the resource. The reviewer/curators consider these suggestions and decide whether they are valid, sound proposals, or if they are problematic, or simply incorrect.

As the model scales for some applications, there may be intermediate reviewers that moderate the public contributions and act as a filter for the resource curators. These participants act as a reviewer of content from the perspective of the original contributors, however each suggestion that an intermediate reviewer allows (i.e., that is judged to be correct) is in turn passed on to the next higher curatorial authority. As a result, the intermediate reviewers appear to the resource curators to be contributors, and so these participants act in a dual role of both reviewer and contributor.

4.2 Reputation model

Many other systems employ reputation models in an attempt to surface more reliable or more popular content produced within a large community. In the CONCUR model, reputation is important for similar reasons. First, we want to highlight productive contributors within the community to motivate their continued work. Second, and perhaps more importantly, we need the system to accommodate the limited time that many resource curators can devote to this work. To make their time as productive as possible, the system must sort all the pending suggestions so that the most likely suggestions are presented first. If a curator has time to review five suggestions a day, these should be the five best suggestions in the system. By tracking the reputation of the contributors (including reviewer-contributors that vet suggestions before the curator sees them), such a ranking is possible.

However, what existing reputation systems lack is a weighting factor for the utility of the contributions made in the past. For example, suppose Professor A. suggests new synonyms for concepts in an ontology that are consistently correct but are in such an obscure area that the impact is very minor, while Student B. suggests new inference rules that are only correct 70% of the time, but when correct enable discovery of many additional objects on popular search terms. In a reputation model based purely on correctness, Prof. A. would have new suggestions ranked above those of Student B., but since the CONCUR reputation model weights correctness *and* utility, the suggestions of Student B. rise up in the queue.

An important principle is the flexible and loosely Wittgensteinian notions of correctness and utility of annotations. In this model, the contributions are *correct* to the extent that the resource curator agrees with them, and are *useful* to the extent that some application metric or community activity reflects use.[2]

Correctness is established by reviewers and modeled as a function of graph locality in an associated domain ontology. We use an ontology to model the fact that an expert in Chinese Ceramics should perhaps be able to carry some of her recognized expertise to the area of Korean Statuary, but a great deal less reputation should be assumed if she is working on concepts around African Textiles. The correctness measure decays with conceptual distance from directly measured concepts (initially using graph distance as a proxy, but allowing in principle for express modeling of conceptual distance in the ontology).

For a CONCUR application that maintains an ontology, the domain model is implicit in the resource itself. For something like a gazetteer that has a tree or directed graph structure, a mapping to subject areas is also straightforward. For something like a dictionary that does not have an implicit graph structure, it may be necessary to specify a domain ontology and to declare the subject areas into which suggested contributions fit, as part of the submission process.

4.3 Measuring utility

Utility is somewhat more difficult to measure than correctness, and will be a function of the specific application. As such, rather than describing required metrics, the CONCUR model instead specifies a contract (API) for a service that encapsulates the specifics of the resource and application. In this way, the model can accommodate a range of utility metrics for different applications. We are developing several prototype applications to explore specific metrics for common resource types, both to explore the model and as well to provide a working system for some common use-cases.

[2] In one sense, the ultimate users whose activity can determine utility fill a third role. This role is generally abstracted behind the utility metrics and so we do not describe it in the model.

In an application to maintain a categorization and inference ontology such as that used by the Delphi system [15] and semantic search tools, some candidate utility metrics include:

Text-mining impact: How the change affects recall and/or precision, where we use counts of text-mining matches (the number of objects the change impacts) as a simple proxy.

User-interest: If a new concept or synonym is proposed, we can track usage of these by end-users. This is non-trivial to capture in practice, and it remains to be seen how best to capture and weigh different specific activities to measure user-interest. Furthermore, we must consider the inherent directionality of the graph when we analyze activity: changes to a narrower concept should be considered to be of interest to the extent that a closely broader concept gets lots of activity, however the converse should be less so: just because a narrower concept gets lots of activity may not mean the broader concept has a comparable level of interest.

In a name authority with corpus citations, the analogous metrics could be the number of citations found for a given person/name, and the level of interest from the user community (e.g., using searches or clicks on a name or name-variant).

4.4 Abstracting the curated resource

Just as we abstract the model for gathering utility metrics for a given application, the framework also defines a basic service contract for the operations that actually update the managed resource. Initially, the model describes basic CRUD (create, read, update, delete) functions for entities in the resource. To maintain the abstraction and still enable flexible definition of the resource, this will be a fairly coarse and loosely-coupled service contract. The rest of the system will reference abstract suggestions/changes.

4.5 Curatorial tools

As reviewer/curators evaluate suggested changes, they should ideally have tools that let them explore *what-if* scenarios. E.g., for an ontology used to index a given corpus, a tool could perform an incremental index and show the differences produced by the proposed changes. These will tend to be very application specific, but will make it easier for curators to assess suggested changes.

5. CONCUR DEVELOPMENT

We are currently building the infrastructure services for CONCUR, refining the service models and building out the main framework. One prototype application focuses on the faceted ontology that supports the Delphi deployment for the Phoebe A. Hearst Museum of Anthropology. A second prototype will likely focus on a dictionary or gazetteer. We see great demand for these tools and expect to deploy them much more widely as the services mature.

6. ACKNOWLEDGMENTS

Our thanks to Matthew Schutte for his contributions to the model and to an earlier unpublished exploration of these ideas.

7. REFERENCES

[1] Astudillo, H., et al., 2008. Contexta/SR: A multi-institutional semantic integration platform. In *J. Trant and D. Bearman (eds.). Museums and the Web 2008*.

[2] Bhat, V., Oates, T., Shanbhag, V., Nicholas, C., 2004. Finding aliases on the web using latent semantic analysis, *Data & Knowledge Engineering 49 (2004) 129-143*.

[3] Cuneiform Digital Library Initiative, http://cdli.ucla.edu/.

[4] Freebase, created by Metaweb Technologies. http://www.freebase.com/.

[5] Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium (2000) Nature Genet. 25: 25-29.

[6] J. Paul Getty Trust, Art & Architecture Thesaurus (AAT). Available online (as of 6/10/2008) at: http://www.getty.edu/research/conducting_research/vocabularies/aat/about.html.

[7] Gil, Y. and Ratnakar, V., 2002. Trusting information sources one citizen at a time. In *Proc. of the First International Semantic Web Conference, Sardinia, Italy*.

[8] Golbeck, J., Hendler, J. and Parsia B. 2003. Trust networks on the semantic web. In *Proc. of Cooperative Intelligent Agents, 2003*.

[9] Ignat, C-L., Norrie, M., (2006). Supporting Customised Collaboration over Shared Document Repositories. CAiSE 2006: 190-204

[10] Kuter, U., and Golbeck, J., 2007. Sunny: A new algorithm for trust inference in social networks, using probabilistic confidence models. In *Proc. of AAAI, 2007*

[11] Noy, N.F. Chugh, A. Alani, H., 2008. The CKC Challenge: Exploring Tools for Collaborative Knowledge Construction. IEEE Intelligent Systems, Jan-Feb 2008.

[12] Rector, A., et al., 2003. OpenGALEN: Open Source Medical Terminology and Tools. AMIA Annu. Symp. Proc. 2003.

[13] Rutledge, L., Aroyo, L., Stash,, N., 2006. Determining user interests about museum collections. WWW 2006: 855-856

[14] Schmitz, P., 2006. Inducing Ontology from Flickr Tags. In *Proc. of Collaborative Web Tagging Workshop,* WWW '06.

[15] Schmitz, P., and Black, M., 2008. The Delphi Toolkit: Enabling Semantic Search for Museum Collections. In *J. Trant and D. Bearman (eds.). Museums and the Web 2008*.

[16] Sunstein, C., 2006, Infotopia, How many minds produce knowledge, Oxford University Press, New York, NY

[17] Whetzel, T., 2006. Pronto. Presentation given at MGED Ontology Workshop. MGED9 – Seattle, 2006.

[18] Wu, H., Zubair, M., and Maly, K. 2006. Harvesting social knowledge from folksonomies. In *Proc. of the 17th Conference on Hypertext and Hypermedia (2006)*.