CiteSeer^x, the Next Generation CiteSeer

Isaac Councill, C. Lee Giles
Information Sciences and Technology
Pennsylvania State University, University Park, PA, 16802, USA
http://citeseer.ist.psu.edu

CiteSeer was a radical change in research and scholarly information access and is still a popular search engine and digital library. However, its original architecture was flawed from the beginning since it was designed with highly integrated software modules with unscalable procedures and a lack of expandability. Next Generation CiteSeer, CiteSeer^x, was created to remedy these problems with a redesign that increases utility, reliability and services, and expands the breadth and depth of CiteSeer's collection. As part of CiteSeer^x we propose a personalized service, MyCiteSeer, to facilitate search through the use of individual search histories combined with exploiting patterns of citations and searches within the community of users. To support collaborative CiteSeer usage and thereby to promote the formation and activity of research communities, collaborative spaces will also be available. The intent is to increase the reliability and sustainability of CiteSeer as a community resource.

In looking for ways to make CiteSeer a useful tool in a collaboration system, we propose 1) making the digital library architecture open to allow researchers in other institutions to integrate their novel algorithms with a stable testbed of documents, indexing, and retrieval technologies and 2) incorporating advanced Web services for authors and those using the system for literature reviews based on past histories. We are currently leveraging the ACM's subject hierarchy for topics in computer science as an organizing structure for the documents in the collection. This, in turn, has allowed us to study trends in publication, and thus trends in research over time based on the publication trends by category over time.

Experience with the legacy CiteSeer system has yielded four distinct lessons that have influenced the design of the Next Generation CiteSeer infrastructure: 1) the system must be capable of evolving to stay current with the state of the art in relevant technologies; 2) a rapid prototyping capability is required for designing new functionality into the system with minimal effort in order to help researchers focus on novel technologies rather than refactoring existing code; 3) it is necessary to provide a robust service framework to help researchers focus on novel technologies rather than engineering production-capable server code; and 4) a monolithic architecture may compound problems inherent in the previous three lessons in addition to making the system difficult to adapt. These issues are treated directly in the Next Generation CiteSeer system through the adoption of a component architecture that takes care of executing code in a high-performance, distributed service framework while treating specific services as plug-ins.

The core execution system developed for Next Generation CiteSeer is designed to integrate disparate program components in a highly scalable, fault-tolerant fashion. The system is analogous in many respects to web service technology in that there exists a central repository of available tasks (services) much like a UDDI repository and tasks may reside on disparate servers available over a network. These tasks may be combined into arbitrary workflows via configurable workflow definitions, much like the emerging Business Process Execution Language (BPEL) standard for web services. The system is different from these standards in that 1) emphasis is on high-speed communication, so heavy-weight protocols such as SOAP are replaced by very efficient protocols implemented within an optimized service environment, and 2) whereas binary data is treated as an afterthought in web service protocols, it is natural to transmit such data in the Next Generation CiteSeer engine, broadening the range of scenarios where task integration is practical.

The framework includes two node types: task providers and workflow handlers. Task providers are simply containers for discrete blocks of program functionality. Each task is given a

unique identifier and is self-describing in terms of what data it requires to operate, its optional parameters, and what data is outputted by the task. These tasks are implemented separately and registered with a task server that manages the execution of each task within a high-performance service framework. Task providers auto-register the descriptions of each task they manage to all available workflow handlers via XML messages, enabling the workflow handlers to maintain a registry of available tasks along with information required to call the tasks dynamically. Workflows are defined by XML configuration files, which are supplied to workflow managers. Arbitrarily complex workflows may be specified including sequential, parallel, and conditional execution of any set of tasks. Workflow managers validate these definitions against available task descriptions and individual workflows are made executable as long as validation is successful and the tasks remain available. Workflows are executed by passing request data to a workflow manager: the manager will execute the appropriate workflow using the request and pass results back to the requesting process.

The data serialization strategy employed by the system is modular so that custom protocols may be employed to transmit program data among nodes. Not only does this allow developers to tailor the communication framework to individual application needs, but custom protocols also open the door for one of the core benefits of traditional web services: language-neutral interoperability.

Next Generation CiteSeer will provide a major user interface enhancement over the legacy CiteSeer system in the form of a personal content portal called MyCiteSeer. This web application will allow each user to maintain an account for storing personal details and provide facilities for viewing and interacting with the repository in new ways. Specific functionality will include allowing MyCiteSeer users to correct and/or supplement repository metadata (a carry-over feature from the legacy system), tag documents, create personal document collections (e.g. favorites or project-specific collections), submit new content to the system with the ability to track the progress and results of submissions, interact with other MyCiteSeer users through a social network application, and craft notification rules for monitoring updates to repository content.

One of the hallmark features of MyCiteSeer is the ability for users to create personalized notification rules for maintaining awareness of new content and changes to existing content. A technique inspired by the Rete algorithm from artificial intelligence literature allows hundreds of thousands of notification rules to be executed in parallel in response to system updates, such that users are notified immediately upon the discovery of new information that may be of interest. Users may craft rules based on Boolean logic to search for arbitrary content in any document metadata field supported by CiteSeer as well as monitor new citations or new documents that are similar to documents of interest. This feature will support researchers who wish to stay abreast of repository content of interest and track their own citations, among many other possible uses. In addition, the notification mechanism will allow institutional tracking by monitoring publications from authors of specific affiliations (for educational or corporate research centers) or by tracking papers that acknowledge specific funding sources (for funding agencies). Such institutional tracking will be useful for automatically discovering which authors belong to a group of interest, what and where they are publishing, and who is citing them.

In contrast to the legacy CiteSeer, the Next Generation CiteSeer relies heavily on machine learning such as support vector machines and clustering for information extraction, entity disambiguation, etc.

Topic: machine learning applications, digital library, search engines
Preference: Oral or poster