IDEA GROUP PUBLISHING **IT3202**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, *International Journal of Web Services Research, Vol. 3, Issue 2*
edited by Liang-Jie Zhang © 2006, Idea Group Inc.

XWRAPComposer:

A Multi-Page Data Extraction Service

Ling Liu, Georgia Institute of Technology, USA
Jianjun Zhang, Georgia Institute of Technology, USA
Wei Han, IBM Research, Almaden Research Center, USA
Calton Pu, Georgia Institute of Technology, USA
James Caverlee, Georgia Institute of Technology, USA
Sungkeun Park, Georgia Institute of Technology, USA
Terence Critchlow, Lawrence Livermore National Laboratory, USA
David Buttler, Lawrence Livermore National Laboratory, USA
Matthew Coleman, Lawrence Livermore National Laboratory, USA

ABSTRACT

We present a service-oriented architecture and a set of techniques for developing wrapper code generators, including the methodology of designing an effective wrapper program construction facility and a concrete implementation, called XWRAPComposer. Our wrapper generation framework has two unique design goals. First, we explicitly separate tasks of building wrappers that are specific to a Web service from the tasks that are repetitive for any service, thus the code can be generated as a wrapper library component and reused automatically by the wrapper generator system. Second, we use inductive learning algorithms that derive information flow and data extraction patterns by reasoning about sample pages or sample specifications. More importantly, we design a declarative rule-based script language for multi-page information extraction, encouraging a clean separation of the information extraction semantics from the information flow control and execution logic of wrapper programs. We implement these design principles with the development of the XWRAPComposer toolkit, which can semi-automatically generate WSDL-enabled wrapper programs. We illustrate the problems and challenges of multi-page data extraction in the context of bioinformatics applications and evaluate the design and development of XWRAPComposer through our experiences of integrating various BLAST services.

Keywords: *code generator; data extraction; service oriented architecture; Web services*

INTRODUCTION

With the wide deployment of Web service technology, the Internet and the World Wide Web (Web) have become the most popular means for disseminating both business and scientific data from a variety of disciplines. For example, vast and growing amount of life sciences data reside in specialized Bioinformatics data sources, and many of them are accessible online with specialized query processing capabilities. Concretely, the Molecular Biology Database Collection currently holds over 500 data

sources (DBCAT, 1999), not even including many tools that analyze the information contained therein. Bioinformatics data sources over the Internet have a wide range of query processing capabilities. Typically, many Web-based sources allow only limited types of selection queries. To compound the problem, data from one source often must be combined with data from other sources to provide scientists with the information they need.

Motivating Scenario

In the Bioinformatics and Bioengineering domain, many biologists currently use a variety of tools, such as DNA microarrays, to discover how DNA and the proteins they encode may allow an organism to respond to various stress conditions such as exposure to environmental mutagens (Quandt, Frech, Karas, Wingender, & Werner, 1995; Altschul et al., 1997; DBCAT, 1999). One way to accomplish this task is for genomics researchers to identify genes that react in the desired way, and then develop models to capture the common elements. This model will be used to identify previously unidentified genes that may also respond in similar fashion based on the common elements. Figure 1 illustrates a workflow that a genomics researcher has created to gather the data required for this analysis. This type of workflow significantly differs from traditional workflows, as it is iteratively generated to discover the correct process with a small set of data as the initial input. At each step the researcher selects and extracts the part of the output data that is useful for his genomic analysis in the next step, and determines which services should be used in the next step in his data collection process. Once the workflow is constructed, the genomic researcher will use the workflow as the data collection pattern to collect large quantities of data and perform large scale genomic analysis. Concretely, Figure 1 shows a pattern of a promoter model where the data collection is performed in eight steps using possibly eight or more Bioinformatics data sources through service oriented computing interfaces.

In Step (1), microarrays containing the genes of interest are produced and exposed to different levels of a specific mutagen in the wet-lab, usually in a time dependent manner.

In Step (2) gene expression changes are measured and clustered using some computational tools (e.g., *Clusfavor* (Peterson, 2002)), such that genes that changed significantly in a micro-array analysis experiment are identified and clustered. The representative genes from Clusfavor analysis will be used as the input for the next data collection step. Typically the researcher must choose from a wide variety of tools available for this task either manually based on his past experience or using a Web service selection facility. Each tool offers specific advantages in terms of their ability to analyze the microarray data, and each requires a different method of execution.

In Step (3), the full sequence from each of the representative genes chosen in the second step is retrieved from gene-banks.

In Step (4), each gene sequence retrieved in Step (3) will be submitted to a gene matching service, such as NCBI Blast Web service, that will return homologs (other genes with similar sequences). The returned sequences will be further examined to find promoter sequences. Again, there are several services that provide gene similarity matching, many of which specialize in a particular species, such as ACEdb (Stein & Thierry-Mieg, 1999).

Once related sequences are discovered, approximately 1000-5000 bases of the DNA sequence around the alignment are extracted to capture the promoter regulatory elements — the region of a gene where RNA polymerase can bind and begin transcription to create the proteins that regulate cell function. In Step (5), these promoter sequences are identified and analyzed using specific tools, such as Mat-Inspector (Peterson, 2002), TRANSFAC, TRRD, or COMPEL (Quandt et al., 1995) to find the common transcription binding factors. To extract specific data, such as portions of a DNA sequence, returned by the sources, the data needs to be converted into a well-known format, such as XML, and post-processed in or-