SCAF - Semantic Contents Acquisition Framework

Danuta Ploch, Thomas Strecker and Ilya Tsyganov
DAI-Labor, TU Berlin
Ernst-Reuter-Platz 7, 10587 Berlin, Germany

danuta.ploch@dai-labor.de, thomas.strecker@dai-labor.de, ilya.tsyganov@dai-labor.de

ABSTRACT

We describe the Semantic Contents Acquisition Framework (SCAF), an approach to extract content of different formats from heterogeneous sources. The SCAF extends standard XSLT with elements for the access to and storage of content. Together with a management and storage infrastructure, the extensions allow a general and personalized harvesting of dedicated sources.

Categories and Subject Descriptors

H.2.5 [**Database Management**]: Heterogeneous Databases; H.3.7 [**Information Storage and Retrieval**]: Digital Libraries

General Terms

Algorithms, Management, Design

Keywords

XSLT, content extraction, semi-structured content, semantic web

1. INTRODUCTION

The amount of accessible digital content increases continuously. The available content, however, is semi-structured or even completely unstructured. Since the content is distributed over various sources which again provide their own formats, content often has to be assembled and transformed into the required structures first, before it can be used for further processing. Especially in the context of Semantic Web a translation of (existing) content into a semantic representation such as RDF is necessary in order to publish and share it in a standardized way. Only then anyone can understand and reason over it. The growing demand for content extraction from different sources and its consolidation led to the development of the SCAF - a framework for supporting and automating the content acquisition process.

The remainder of this paper is organized as follows: We begin in section 2 by describing our approach and by explaining the infrastructure of the SCAF. In section 3 we present then the realization details of each component. The last section covers the outlook where we present our future goals and the research directions associated with the SCAF.

Table 1: Technical Details of the SCAF

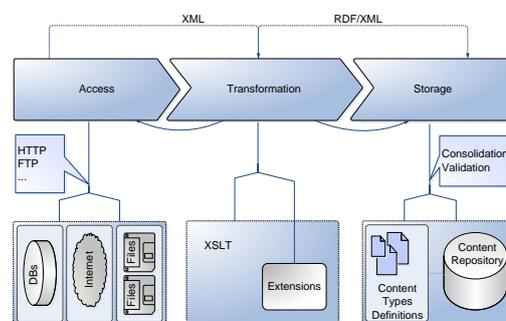| Supported Access	HTTP, FTP, File, OAI, Web Services
Technology	XSLT with extensions
Wrapper-Output	Any
Validation	XML schema definition, OWL ontology
Implemented Storages	File System, DB, Semantic Repository

Figure 1: The SCA-Framework Overview

2. APPROACH

The approach of the SCAF is the integration of standard content transformation techniques and a sophisticated content storage and management infrastructure to provide a highly flexible and robust mechanism for content acquisition. Table 2 gives a technical overview over the functions of the SCAF. In order to organize the content acquisition process – i.e. accessing content of dedicated sources, transforming it into a required representation and storing it in a content repository – a set of components, services and tools was created. Among them are services for the validation and consolidation of contents as well as for the configuration and supervision of the acquisition runs. A configuration defines the acquisition process and consists of a meta-data-part and a part describing the content transformation. Figure 1 shows the general transformation process which has been implemented.

Another component, namely the management component, organizes the acquisition runs, receives the extracted content, stores it in a content repository and notifies subscribers of incoming content.

A comprehensive use of plug-in mechanisms allows the ex-

tension of almost any component of the system for scenarios which we may have not considered yet. The next chapter describes the main components of the SCAF in detail.

3. INFRASTRUCTURE

3.1 Content Access

The first step in the content acquisition process is accessing content from a given location. Unlike Piggy Bank [2] - that only allows the content extraction from websites - the SCAF offers for this purpose a plug-in mechanism and provides specific handlers to access each kind of source. The access may be parameterized in order to facilitate filling and submitting forms, logging in or querying web-databases. The result of a successful execution is an input stream which can be used as "native" content (as byte array) or after parsing it to XML for performing content transformations. Currently supported types of requests are HTTP, FTP, OAI-PMH, web services and file system but the list can be easily extended.

3.2 Content Transformation

After having gathered content from a source the XML content has to be transformed into the desired representation. For solving this task the SCAF uses XSLT [1] to define templates and flow definitions, and XPath to access or refer to parts of the XML document. The specific task of transforming content into semantic representations imposes the need for functionality beyond the standard XSLT. In contrast to Lixto [1] – which uses XSLT and the system-internal language Elog – we use the extension mechanism of XSLT and introduced elements which allow performing the following functions.

The *open* extension encapsulates a request and is therefore capable of loading a wide range of contents from various sources into the current XML tree. The *content* extension also contains a request. In contrast to the open extension it allows to save the request result as native content to the content repository. Hence, the extension can be used to easily implement mirroring functionalities. The *about* extension computes the result of the contained transformation and returns the structured content to the management component. The format of the structured content can be anything just as the typical result of a XSLT transformation. Some other extension elements have been introduced to ease the process of stylesheet definition. Currently they include the *date*, *current-uri*, *hash* and *uuid* extension.

3.3 Content Storage

Like most every component of the SCAF, the content storage component also can be easily plugged into the SCAF depending on the requirements of a system. At the moment a flexible storage in three different content storage types is implemented. It is possible to persist extracted content into a database, the file system or a semantic store. In addition to the pure storage functionality this component can also be used for the validation of incoming content. Again, different types of content validators can be plugged in for different content formats. Already realized are the validation of XML content against an XML schema definition and the validation and consistency checking of RDF descriptions against

---

[1]http://www.w3.org/Style/XSL/

an OWL ontology. For keeping the content storage clean and achieve a high grade of content quality basic algorithms for content de-duplication and -merging were developed.

3.4 Content Acquisition Management

As already mentioned, the management component (the manager) controls the content acquisition process. One of its main tasks is the coordination of the extractors. The manager assures a continuous content extraction according to the schedules of the extractors and allows in this way to monitor sources and to fetch updates even from frequently changing sources. If errors occur the manager receives notifications and it has the ability to perform exception handling procedures. Another important function is to notify subscribers of incoming content. Since some extracted contents may be restricted to private use, the SCAF protects them by checking the requesting user's permissions for each query of the content storage. Furthermore the manager is responsible for the organization of all artifacts needed for setting up and running the SCAF. It therefore has to manage configurations, parameter-list, users, subscriptions, permissions and acquisition setups. Again the access to restricted artifacts can be controlled by the management instances, if desired. This ensures that sensitive information such as credentials or personal content can only be used by authorized instances within the infrastructure. For defining a content acquisition process an editor has to write a configuration. To facilitate the configuration process, the SCAF provides a configuration tool (Eclipse plug-in), which supports the editor to set all necessary meta-data, such as the extraction schedule or the initial request, and to write the actual transformation part of the configuration in extended XSLT.

4. FUTURE RESEARCH

The SCAF with its extensible infrastructure works well as a platform for including and testing of research results. Currently editors have to write transformation configurations manually with the aid of the configuration tool. And although the tool offers a lot of support, the grade of automatic configuration generation could be augmented and is the subject of future research. An approach is the visual analysis of the accessed content and the semi-automatic mapping of the content to structured content types. As previously reported the SCAF provides basic content-integration and de-duplication algorithms for the gathered content. The goal of future research in this area is to improve the algorithms to perform a sophisticated record linkage, make implicit knowledge explicit and increase therefore the content quality significantly.

5. REFERENCES

[1] R. Baumgartner, O. Frölich, and G. Gottlob. The Lixto systems applications in business intelligence and semantic web. In *Proceedings of the 4th European Semantic Web Conference, ESWC 2007*, pages 16–26, 2007.

[2] D. Huynh, S. Mazzocchi, and D. Karger. Piggy bank: Experience the Semantic Web inside your web browser. *Web Semantics: Science, Services and Agents on the World Wide Web*, pages 16–27, 2007.