Acoi: A System For Indexing Multimedia Objects

Menzo Windhouwer, Albrecht Schmidt, Martin Kersten
CWI, Amsterdam, The Netherlands

Introduction

The explosion of the number of Web pages also leads to countless accessible multimedia objects. Their abundance makes the Internet an interesting application for multimedia retrieval systems. Many search engines are going about to supply some retrieval functionality for independent retrieval of these objects. However, most of these multimedia search engines aim at a fixed set of multimedia index attributes. The **Acoi** system [1] provides an extensible framework for retrieving multimedia objects of any type on basis of their content, based on both low-level features and high-level concepts, and context.

In the following sections, which describe different aspects of the system, this example grammar is used as an illustrative example:

```
%atom    str      url, content_type, title, section, word, alt;
%detector         web_header(url);
%detector         page_type
                      select  true
                      from    web_object
                      where   content_type = "text/html";
%detector         web_page(url);
%start            web_object;
web_object       : url web_header web_body?;
web_header       : content_type;
web_body         : page_type web_page;
web_page         : title? anchor*;
anchor           : web_object section? alt? word*;
```

Feature Detectors

Feature detectors are used to build a semantically rich index entry for the original multimedia object. They do this on two different levels:

1. *Blackbox detectors* are implemented in a programming language to access the raw multimedia data and to derive the desired features from it. *Example:* the `web_header` detector sends a HTTP HEAD request to the specified HTTP server and extracts the content type from the respone.
2. *Whitebox detectors* consist of queries over the already collected feature values. *Example:* the `page_type` detector uses the content type to determine if an object is a page.

In the general case blackbox detectors will derive low-level feature data, *e.g.*, the color distribution of an image. But they can also be used for more complex tasks, like finding a face in an image. The function of whitebox detectors is to relate low-level features to concepts, *e.g.*, an image is a portrait because its color distribution classifies it as a photo and it contains exactly one face.

Feature Grammars

The foundation of the whole Acoi system is formed by the concept of feature grammars. A feature grammar is basically a context-free grammar extended with active non-terminals, *i.e.*, the different types of detectors. The grammar plays the following roles in the system:

- It is used as a *parser specification*. As detectors are depending on each others results, the grammar specifies the order (*i.e.*, top-down and left-right) in which they should be executed. *Example:* the `page_type` detector depends on the `web_header` detector.
- It describes the possible *relationships* between objects and names their roles in each others contexts. *Example:* other `web_objects` are related to a `web_page` in the context of `anchors`.
- Its rules can directly be translated to a *database schema*, where the left-hand side of a rule represents the table name and the right-hand side the attributes of this table. Non-terminal right-hand side symbols will become foreign keys to an entry in their own table. *Example:* the `web_object` table will contain an `url` attribute and two foreign keys: one to a `web_header` and, optionally, one to a `web_body`.

Extensibility

Multimedia retrieval is not yet a solved problem (and may never be), so the index should be easily extensible with new feature detectors. Feature grammars are quite easy to extend: just add new rules. The parser can then do an incremental parse: it takes a persistent stored parse tree and calls the new detectors to extend the branches. *Example:* the example grammar could be extended with these rules to support content-based retrieval for images:

```
%atom    int      width, height, depth, color, frequency;
%detector         image_type
                       select   true
                       from     web_object
                       where    content_type = "image/gif";
%detector         web_image(url);
web_body         : image_type web_image;
web_image        : width height depth histogram;
histogram        : color* frequency;
```

The incremental parse would try to prove the validity of this new `web_body` alternative and on success add new indexing information for images to the database.

System Architecture

The following image shows the current Acoi system architecture.
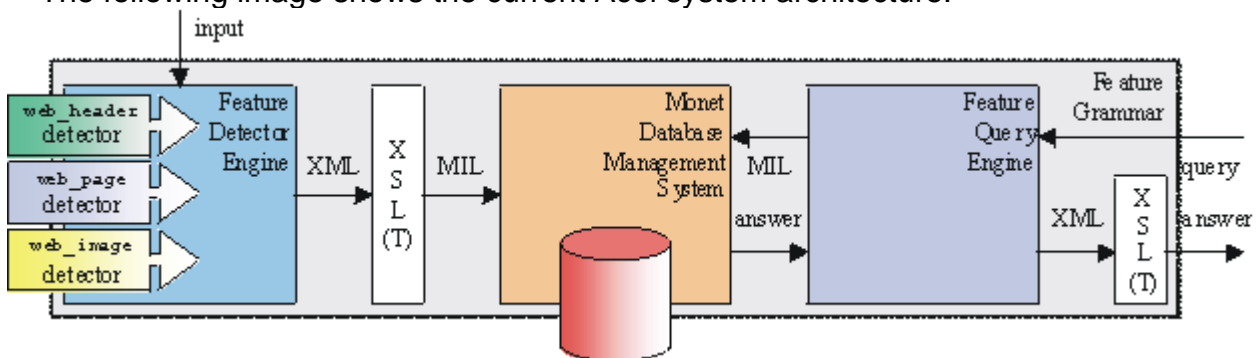
The Feature Detector Engine is the parser generated from the feature grammar. The parse trees it produces are stored in **Monet**, an extensible main memory database system. Queries, in a SQL-like syntax, are processed by the Feature Query Engine. **XML** is used as exchange format between the different tools, and **XSL(T)** is used to transform the XML document when another (proprietary) format is needed.

References

1. M.Windhouwer, A.Schmidt, M.Kersten. **Acoi: A System for Indexing Multimedia Objects**. In *International Workshop on Information Integration and Web-based Applications & Services,* Yogyakarta, Indonesia, November 1999.