Chapter 4
GuessXQ:
A Query-by-Example Approach for XML Querying

Daniela Morais Fonte
University of Minho, Portugal

Daniela da Cruz
University of Minho, Portugal

Pedro Rangel Henriques
University of Minho, Portugal

Alda Lopes Gancarski
Institut Telecom, France

ABSTRACT

XML is a widely used general-purpose annotation formalism for creating custom markup languages. XML annotations give structure to plain documents to interpret their content. To extract information from XML documents XPath and XQuery languages can be used. However, the learning of these dialects requires a considerable effort. In this context, the traditional Query-By-Example methodology (for Relational Databases) can be an important contribution to leverage this learning process, freeing the user from knowing the specific query language details or even the document structure. This chapter describes how to apply the Query-By-Example concept in a Web-application for information retrieval from XML documents, the GuessXQ system. This engine is capable of deducing, from an example, the respective XQuery statement. The example consists of marking the desired components directly on a sample document, picked-up from a collection. After inferring the corresponding query, GuessXQ applies it to the collection to obtain the desired result.

DOI: 10.4018/978-1-4666-2669-0.ch004

INTRODUCTION

In this chapter, we address the problem of accessing information in structured documents annotated in eXtensible Markup Language (XML). Those documents, being structured, are accessed using specific query languages where the interesting structural components are specified, as well as restrictions over them if needed.

The bigger the worldwide collection of XML documents gets, the more relevant is the existence of an efficient search engine. These engines should be aware of the explicit structure of the documents. This problem has raised a research area called Structured Document Retrieval (Lu, 1990). In this area, the specification of a query that yields valid results strongly depends on the user-friendliness of the search engine interface.

The standard query language for XML is XQuery (Boag, et al., 2005). XQuery queries are powerful but complex to write (the user must have a deep knowledge of the query language as well as the document structure). To help the user in the task of specifying his queries, some specialized editors have been developed (XMLSpy [Kim, 2002], EditiX[1], Oxygen[2]), but still requiring a good knowledge level of the query language.

"Example is always more efficacious than precept." This statement, by Samuel Johnson (1999), led Human-Computer Interaction researchers to suggest a new interaction paradigm called Query-by-Example (QBE). Born in the context of database querying (Ramakrishnan & Gehrke, 2007), typical QBE systems are based on the "fill in the blanks" approach. Zloof (1975) defined QBE as "a query language for use by non-programmers querying a relational database." QBE is based on the concept that the user formulates his query by filling in the appropriate skeleton tables the fields and/or restrictions on fields (relational selection concept) he intends to search.

Due to the complex nature of XML documents querying, the QBE concept was adapted to XML retrieval by showing the XML Schema Definition (XSD)[3] instead of the relational table skeleton (see for instance Tulchinsky, et al., 2008; Bohere, et al., 2003; Zhang, et al., 2002; Li, et al., 2007; Braga & Campi, 2005; Newman & Ozsoyoglu, 2004). XSD consists in a XML Schema Language usually used to express a set of rules to which an XML document must conform in order to be considered *valid* or *well formed* (according to that schema). Other very well known notation to define document families' structure is the Document Type Definition (DTD), which uses a formal syntax to declare rigorously the elements and references that may be used in the documents of the family and which content type the elements can have. Despite its simplicity and wide use, DTD is being replaced by XSD for two main reasons: XSD is written in XML and it is much more powerful and rigorous, allowing a more complete definition.

The system we present here called GuessXQ, also displays the XML Schema tree representation to the user. Moreover, the user has the chance to go through a sample (a XML document extracted from the repository) and mark over it the components he wants to retrieve from the overall collection. Element selections and restrictions are done directly on the sample document, giving the user a clear indication of the information he is searching for (Ferreira, et al., 2007; Cruz, et al., 2009; Fonte, et al., 2010).

This chapter describes how we implement the QBE concept in a Web-application for information retrieval from a collection of structured documents, the GuessXQ system. In essence, we propose an engine capable of deduce, from a specific example, the respective XQuery statement. Thus, instead of specifying the desired components of the documents and eventual restrictions in a query, the user exemplifies those components marking them directly on a sample document, picked-up from the collection. After inferring the generic statement, GuessXQ system applies it to all documents in the collection to perform the desired retrieval.

This chapter is organized as follows. The introduction briefly presents XML querying and