With the emergence of the World Wide Web, business' databases are increasingly being queried directly by customers. The customers may not be aware of the exact structure of the underlying data, and might have never learned a query language that enables them to issue structured queries. Some of the employees who query the databases may also not be aware of the structure of the data, but they are likely to be aware of some labels of elements containing the data. There is a need for a dual search engine that accommodates both business employees and customers. We propose in this chapter an XML search engine called SEEC, which accepts Keyword-Based queries (which can be used for answering customers' queries) and Loosely Structured queries (which can be used for answering employees' queries). We proposed previously a stand-alone Loosely Structured search engine called OOXSearch (Taha & Elmasri, 2007). SEEC integrates OOXSearch with a Keyword-Based search engine and uses novel search techniques. It is built on top of an XQuery search engine (Katz, 2005). SEEC was evaluated experimentally and compared with three recently proposed systems: XSEarch (Cohen & Mamou & Sagiv, 2003), Schema Free XQuery (Li & Yu & Jagadish, 2004), and XKSearch (Xu & Papakonstantinou, 2005). The results showed marked improvement.

INTRODUCTION

XML has had significant boost with the emergence of the World Wide Web, online businesses, and the concept of ubiquitous computing. The majority of current web services are based on XML. In a corporate environment, XML has been used to import/export data as well as in internal documentation. The popularity of XML is due, in part, to the following:
• XML defines the type of information contained in a document, which helps in restricting the search and makes it easier to return the most relevant results. Consider for example a university professor, who is participating in many activities. When using HTML to search for courses that the professor plans to teach, the results are likely to contain information outside the context of courses. When using XML, on the other hand, the search could be restricted to information contained in the appropriate element type (e.g. `<courses>`).

• XML implementations make electronic data interchange more accessible for e-businesses, as they are easily processed by automated programs.

• XML can be customized to suit the need of any institution or business.

• XML is URL-addressable resource that can programmatically return information to clients.

With the increasing interest in XML databases, there has been extensive research in XML querying. Some work model the XML data as a rooted tree (Cohen & Mamou & Sagiv, 2003; Li & Yu & Jagadish, 2004; Xu & Papakonstantinou, 2005), while others model it as a graph (Balmin & Hristidis & Papakonstantinou, 2003; Balmin & Hristidis & Papakonstantinou, 2004; Botev & Shao & Guo, 2003; Cohen & Kanza, 2005). However, most of these work targets either naïve users (such as business customers) by proposing Keyword-Based search engines or sophisticated users by proposing fully structured query search engines. We believe there is a need for an XML search engine that answers each user based on his/her degree of knowledge of the underlying data and its structure. Business customers are most likely not to be aware of the underlying data and its structure. On the other hand, business employees are likely to be aware of some data elements' labels (elements containing data), but they are unlikely to be aware of the structure of the data. We propose in this chapter an XML dual search engine called SEEC (Search Engine for Employees and Customers), which meets the needs of both customers and employees. It accepts XML Keyword-Based queries (e.g. for answering customers' queries), and XML Loosely Structured queries (e.g. for answering employees' queries). Consider that a user wants to know the data D, which is contained in an element labeled E. If the user knows ONLY the keywords k_1, k_2, ..., k_n, which are relevant to D, the user can submit a Keyword-Based query in the form: Q ("k_1", "k_2", ..., "k_n"). If, however, the user knows the label E and the labels (which belong to the elements containing the keywords k_1, k_2, ..., k_n respectively), but is unaware of the structure of the data, this user can submit a loosely structured query in the form: Q ("k_1", ..., = "k_n", E?). We proposed previously a stand-alone Loosely Structured search engine called OOXSearch (Taha & Elmasri, 2007). SEEC integrates OOXSearch with a Keyword-Based search engine and uses novel search techniques. It is built on top of an XQuery search engine (Katz, 2005).

Few researchers (Cohen & Mamou & Sagiv, 2003; Li & Yu & Jagadish, 2004) propose XML search engines that answer both Keyword-Based and Loosely Structured queries. Computing the Lowest Common Ancestor (LCA) of elements containing keywords is the common denominator among these engines. Despite their success, however, they suffer recall and precision limitations. As we will show in the coming sections, the reason for these limitations stems from the fact that these engines employ mechanisms for building relationships between data elements based solely on their labels and proximity to one another while overlooking the contexts of these elements. The context of a data element is determined by its parent, because a data element is generally a characteristic (or a property) of its parent. If for example a data element is labeled “title”, we cannot determine whether it refers to a book title or a job title without referring to its parent. Consider as another example that an XML document containing two elements labeled “name”, one of which refers