Abstract: Access to Web data has become an integral part of many applications and services. In the past, such data has usually been accessed through human-tailored HTML interfaces. Nowadays, data is more and more served in (syntactically and semantically) richer data formats such as XML and RDF that can be conveniently accessed and processed with Web query languages. However, ordinary Web query languages such as XQuery, XSLT, and SPARQL focus on a single Web format. Xcerpt goes beyond these languages and provides versatile access to data in different Web formats within the same query. This article highlights Xcerpt’s essential principles and features.

1. Introduction

Web querying has been recognized in recent years as a convenient means to access data on the Web, in particular with the increase of Web data in (syntactically and semantically) richer data formats than HTML. Numerous XML or RDF query languages have been proposed from both industry and academia, culminating in recent standardization activities at the W3C on the XML query languages XSLT and XQuery and the RDF query language SPARQL.

These conventional query languages, however, focus only on one of the different data formats available on the Web. Integration of data from different sources and in different formats becomes a daunting task that requires knowledge of several query languages and overcoming the impedance mismatch between the query paradigms in the different languages. For instance, bibliography management applications already access (in varying combinations) book data from Amazon, Barnes & Noble, and other vendors, citation data from CiteSeer, PubMed, ACM’s digital library, etc., topic and researcher classifications in RDF format by crawling or from syndication sites, and keywords, abstracts, or table of contents from DocBook representations of articles.

We argue that for such applications Web query languages need to be versatile [BFB05], i.e., to be able to access data in different Web representation formats.
Xcerpt [SB04, Sch04] addresses this issue by garnering the entire language towards versatility in format, representation, and schema of the data following principles laid down in [BS02, BFB04, BFB05]. It is a semi-structured query language, but very much unique among such languages (for an overview see [BBFS05]):

1) In its use of a graph data model, it stands more closely to semi-structured query languages like Lorel than to recent mainstream XML query languages.

2) In its aim to address all specificities of XML, it resembles more mainstream XML query languages such as XSLT or XQuery.

3) In using (slightly enriched) patterns (or templates or examples) of the sought-for data for querying, it resembles more the “query-by-example” paradigm than mainstream XML query languages using navigational access. Patterns and data are matched with a novel unification algorithm, called simulation unification [BFSS05]

4) In offering a consistent extension of XML, it is able to incorporate access to data represented in richer data representation formats. Instances of such features are element content, where the order is irrelevant, and non-hierarchical relations.

5) In providing (syntactical and semantical) extensions for querying, among others, RDF [FBB05, Bol05], Xcerpt becomes a versatile query language.

6) In its strict separation of querying and construction, it makes query authoring and query evaluation easier.

7) In its rule-based nature, it makes basic forms of reasoning possible as part of query programs and enables transparent mediation and integration of data by logical views. The rule layer of Xcerpt follows principles described in [BM05].

Xcerpt is currently being further developed and refined at the University of Munich and as part of the activities of the Working Group on “Reasoning-aware Querying” in the EU Network of Excellence REWERSE (“Reasoning on the Web with Rules and Semantics”), cf. http://rewerse.net/. For more information, including a prototype implementation, on Xcerpt refer to http://xcerpt.org/ and http://rewerse.net/I4/.

The remainder of this article briefly introduces the basic constructs of the Xcerpt query language and offers a glimpse of the language syntax and constructs along a concrete example from bibliography management. The article concludes with an overview of current research issues surrounding Xcerpt and its sister languages.

2. Xcerpt Basics

2.1 Data as Terms

Xcerpt uses terms to represent semi-structured data. Data terms represent XML documents, RDF graphs, and other semi-structured data items. Notice that subterms (corresponding to, e.g., child elements) may either be “ordered” (as in an XHTML document or in RDF sequence containers), i.e., the order of occurrence is relevant, or “unordered”, i.e., the order of occurrence is irrelevant and may be ignored (as in the case of RDF statements). In the term syntax, an ordered term specification is denoted by square brackets [ ], an unordered term specification by curly braces { }. Terms may contain the reference constructs ^id (“referring” occurrence of the identifier id) and id @ t (“defining” occurrence of the identifier id). Using reference constructs, terms can form (possibly cyclic, but rooted) graph structures. Term attributes are
denoted in round parentheses ( ). Terms are similar to ground functional programming expressions and logical atoms. A non-XML syntax has been chosen for Xcerpt to improve readability, but there is a one-to-one correspondence between an XML document and a data term.

2.2 Queries as Terms
Following the “query-by-example” paradigm, queries are merely examples or patterns of the queried data and thus also terms, annotated with additional language constructs. Xcerpt separates querying and construction strictly. Query terms are (possibly incomplete) patterns matched against Web resources represented by data terms. In many ways, they are like forms or examples for the queried data, but also may be incomplete in breadth, i.e., contain ‘partial’ as well as ‘total’ term specifications: A term \( t \) using a partial term specification for its subterms matches with all such terms that (1) contain matching subterms for all subterms of \( t \) and that (2) might contain further subterms without corresponding subterms in \( t \). Partial term specification is denoted by double (square or curly) brackets. Query terms may further be augmented by variables for selecting data items, possibly with “variable restrictions” using the \( \rightarrow \) construct, which restricts the admissible bindings to those subterms that are matched by the restriction pattern. They may contain query constructs like position matching, subterm negation using without, optional subterms, regular expressions for namespaces, labels, and text, and conditional or unconditional path traversal using desc. Finally, they may contain further constraints on the variables in a so-called condition box, beginning with the keyword where. Construct terms serve to reassemble variables (the bindings of which are gained from the evaluation of query terms) so as to construct new data terms. Again, they are similar to the latter, but augmented by variables (acting as place holders for data selected in a query) and the grouping construct all (which serves to collect all instances that result from different variable bindings). Occurrences of all may be accompanied by an optional sorting specification.

2.3 Rules and Programs
Query and construct terms are related in rules which themselves are part of Xcerpt programs. Rules have the form:

```
CONSTRUCT construct-term
FROM and { query-term or { query-term ... } ... } END
```

Rules can be seen as “views” specifying how to obtain documents shaped in the form of the construct term by evaluating the query against Web resources (e.g. an XML document or a database). Xcerpt rules may be chained like active or deductive database rules to form complex query programs, i.e., rules may query the results of other rules.

More details on Xcerpt’s language constructs, syntax, and semantics can be found in [Sch04, SBF05, FBS06].

3. Data Access in Bibliography Management: A Use Case for Xcerpt
In bibliography management, applications more and more access Web sources to complete bibliographic information, to find citations, author affiliations, abstracts, related articles, etc.

Xcerpt is exceptionally well suited for such data access, as it allows access to different kinds of Web data formats in the same query program, e.g., to DBLP-like article information published according to some XML schema, to articles marked up according to the DocBook format, and to RDF ontologies over topics, institutions, conferences, and/or authors mentioned in the article.
information. The following Xcerpt data term gives an excerpt of an already integrated view on articles with DBLP-like information and DocBook article content.

```
article_66_cicero_wax @ article{
  authors[ ... ],
  title[ "Space- and Time-Optimal Data Storage on Wax Tablets" ],
  within[ scrolls[ "1-94" ], ^journal_adm ],
  content[
    body[ contributions @ h1[ "Contributions" ],
    h1[ "A History of Data Storage: From Stone to Parchment" ],
    p[ "Despite ", cite[ ^article_66_scaurus_qumran ], "..." ];
    ol[
      li[ em[ strong[ "Homeric" ], " Age:" ], "..." ],
      li[ em[ "Age of the ", strong[ "Kings" ], ":" ], "..." ]
    ], ...
    tachygraphy @ h1[ "Challenges for Tachygraphy on Wax" ],
    p[ "Though conditions for writing on wax tablets are adverse ",
      "to tachygraphy, systems as described in ",
      a[ href[ "tiro" ], "Section 2" ], "..." ]
  ]
}
```

From such information, Xcerpt queries can extract information, e.g., as shown in Figure 1, articles that are at least partially covering a given topic or a table of content for an article. Often the information can be represented in varying ways, either due to schemata such as DocBook or due to different sources using different schemata. Such cases call for the powerful pattern matching constructs of Xcerpt such as descendant (qualified or unqualified) to traverse arbitrary length paths in the data, optional to express that certain parts of a query should be matched if occurring in the data, but may also be missing, or subterm negation (expressed using without) that requires that certain parts of a query do not occur in the data.

**Figure 1:** Visual (visXcerpt) Patterns for Xcerpt Queries
4. Current Research Activities

Currently, research around Xcerpt is focused on five core issues:

- The language constructs and syntax is continuously being refined based on growing experience with use cases [BBF05] and applications from various domains, e.g., bioinformatics [DFBS05]. For first results and remaining open issues refer to [FBS06].
- Efficient evaluation and optimization techniques are studied along an abstract machine for Xcerpt. The optimizations combine traditional complexity study and operator order optimizations from databases (cf. [BSFL06, Lin06, Sch05] for first results on their application to Xcerpt) with techniques from compiler construction.
- Both for optimization and query authoring, type checking and type information is helpful and sometimes even essential. Type systems for Xcerpt [BCDW05] are under development as part of the REWERSE working group on “Composition and Typing”.
- A reactive companion language for updates and event processing, called XChange [BPS04, BEP06], is under development in Munich.
- For visual rendering and authoring of data and query, a visualization of Xcerpt, called visXcerpt [BBS03, BBB04], has been developed and demonstrated at several international conferences.

In addition to these issues, Xcerpt’s integration with current advances in Web service description and deployment, a revised API for interfacing Xcerpt with general (object-oriented) programming languages similar to XQJ, the use and tighter integration of Xcerpt with rule-based policy languages for Web and Semantic Web applications, and the automated composition of Xcerpt programs are investigated.

Biography

http://rewerse.net/publications.html#REWERSE-RP-2004-43

http://rewerse.net/publications.html#REWERSE-DEL-2005-I4-D3

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2005-14

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2004-23

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2005-39

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2005-1

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2006-3

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2005-3

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2004-13

http://rewerse.net/publications.html#REWERSE-DEL-2005-I4-D5

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2005-7

http://www.pms.ifi.lmu.de/publikationen#DA_Oliver.Bolzer

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2004-11

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2002-7


http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2005-15
http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2005-38

http://rewerse.net/publications.html#REWERS-DEL-2006-I4-D6

http://rewerse.net/publications.html#REWERS-DEL-2004-I4-D1

http://www.pms.ifi.lmu.de/publikationen#DA_Benedikt.Linse

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2004-1

http://www.pms.ifi.lmu.de/publikationen#PMS-FB-2004-7

http://rewerse.net/publications.html#REWERS-DEL-2005-I4-D4

http://www.pms.ifi.lmu.de/publikationen#PMS-DISS-2004-1

http://www.pms.ifi.lmu.de/publikationen#DA_Andreas.Schroeder

Benedikt Linse holds a master’s degree in Computer Science from the University of Munich, Germany (2006), where he is currently employed as a research and teaching assistant. His present research interests include versatile query languages for the (Semantic) Web, automatic translation between XQuery and Xcerpt, reasoning and efficient rule chaining in Xcerpt, and efficient query evaluation.

Sebastian Schaffert holds a PhD in Computer Science from the University of Munich, Germany (2004). He is currently employed as a senior researcher at “Salzburg Research Forschungsgesellschaft”, Austria. His research interests are in semantic web technology, knowledge representation and reasoning, social and educational software, and programming languages. He has contributed to many scientific conferences as author and program committee member and has authored several publications on the rule-based Web query language Xcerpt and related topics.