Effective and Efficient Data Access in the Versatile Web Query Language Xcerpt

Sacha Berger, François Bry, Tim Furche, Benedikt Linse, and Andreas Schroeder

Institute for Informatics,University of Munich,
Oettingenstraße 67, 80538 München, Germany
http://pms.ifi.lmu.de/

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, rich client interfaces in desktop applications or, increasingly, in browser-based clients ease data access and allow more complex client processing based on XML or RDF data retrieved through Web service interfaces. Convenient specifications of the data processing on the client and flexible, expressive service interfaces for data access become essential in this context. Web query languages such as XQuery, XSLT, SPARQL, or Xcerpt have been tailored specifically for such a setting: declarative and efficient access and processing of Web data. Xcerpt stands apart among these languages by its versatility, i.e., its ability to access not just one Web format but many. In this demonstration, two aspects of Xcerpt are illustrated in detail: The first part of the demonstration focuses on Xcerpt's pattern matching constructs and rules to enable effective and versatile data access. It uses a concrete practical use case from bibliography management to illustrate these language features. Xcerpt's visual companion language visXcerpt is used to provide an intuitive interface to both data and queries. The second part of the demonstration shows recent advancements in Xcerpt's implementation focusing on experimental evaluation of recent complexity results and optimization techniques, as well as scalability over a number of usage scenarios and input sizes.

1 Introduction

Web querying has received considerable attention from academia and industry culminating in the recent development of the W3C Web query languages XQuery and SPARQL. These main-stream 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 to overcome the impedance mismatch between the query paradigms in the different languages. Xcerpt [8, 9] addresses this issue by garnering the entire language towards versatility in format, representation,

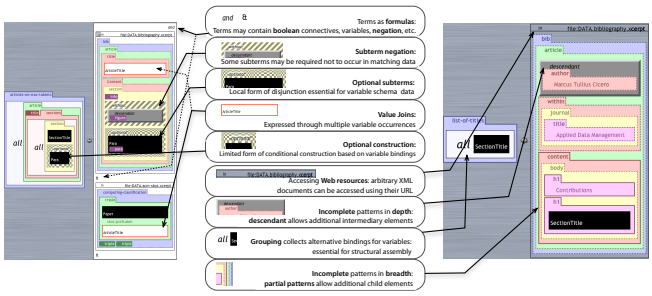


Fig. 1. Exemplary visXcerpt Query Patterns

and schema of the data, cf. [5]. It is a *semi-structured query language*, but very much unique among such languages (for an overview see [1]):

(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 [10] than mainstream XML query languages using navigational access.

(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) extensions for querying, among others, RDF, Xcerpt becomes a *versatile query language*, cf. [5].

(6) In its strict separation of querying and construction in *rules*, it makes programs more readable and optimization over intermediary results feasible.

visXcerpt [2] is Xcerpt's visual companion language related to it in an unusual way: visXcerpt is a *visual* query language obtained by mere rendering of Xcerpt without changing the language constructs or the runtime system for query evaluation. This rendering is mainly achieved via CSS styling of Xcerpt's constructs. The authors believe that this approach is promising, as it makes those languages easy to learn—and easy to develop.

This demonstration is split in two parts: first the novel language constructs for versatile pattern matching and rule-based data integration are illustrated along a practical demonstrator application using visXcerpt. Xcerpt's core features, especially the pattern-oriented queries and answer-constructors, its rules or views, and its specific language constructs for incomplete specifications are emphasized in this application. It is demonstrated (a) how incomplete specifications are essential for retrieving semi-structured data, (b) how access to both Web and Semantic Web data in the same query program is achieved and (c) how

visXcerpt complements and integrates with Xcerpt. Special emphasis is placed on recent advancements in language constructs and concepts.

The second part of this demonstration focuses on the evaluation and optimization of Xcerpt queries. In particular, it shows experimental confirmation of recent complexity results for various Xcerpt subsets. Furthermore, an impression of the effects of recent optimizations of complex queries involving negated or optional subterms is given.

2 Part I: Language Features and visXcerpt

Setting of the Demonstrator

Excerpts from DBLP[1] and from a computer science taxonomy form the base for the scenario considered in the application. DBLP is a collection of bibliographic entries for articles, books, etc. in the field of Computer Science. DBLP data is a representative for standard Web data using a mixture of rather regular XML content combined with free form, HTML-like information. A small Computer Science taxonomy has been built for the purpose of this demonstration. Very much in the spirit of SKOS, this is a lightweight ontology based on RDF and RDFS. Combining such an ontology as metadata with the XML data of DBLP is a foundation for applications such as community based classification and analysis of bibliographic information using interrelations between researchers and research fields. Realizing such applications is eased by using the integrated Web and semantic Web query language (vis)Xcerpt that also allows reasoning using rules.

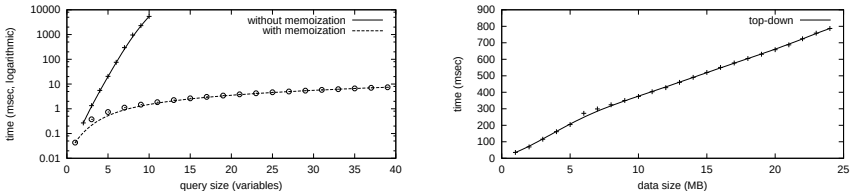
Realizing Versatility

Query and construction *patterns* in (vis)Xcerpt are used, both for binding variables in query terms and for reassembling the variables in so-called construct terms. The variable binding paradigm is that of Datalog: the programmer specifies patterns including variables. Interactive behavior of variables in visXcerpt highlights the relation between variables in query and construct terms. Arguably, pattern based querying and constructing together with the variable binding paradigm make complex queries easier to specify and read.

To cope with the semistructured nature of Web data, (vis)Xcerpt query patterns use a notion of incomplete term specifications with optional or unordered content specification. This feature distinguishes (vis)Xcerpt from query languages like Datalog and query interfaces like QBE [10]. Simple, yet powerful textual and visual constructs of incompleteness are presented in the demonstrator application, cf. Figure 1 showing two exemplary visual query patterns and a breakdown of used language constructs.

An important characteristic of (vis)Xcerpt is its rule-based nature: (vis)Xcerpt provides rules very similar to SQL views. Arguably, rules or views are convenient

[1] `http://www.informatik.uni-trier.de/~ley/db/`



(a) Effect of Memoization (b) Scalability on Real Data

Fig. 2. Experimental Evaluation of "Memoization Matrix" Approach

for a logical structuring of complex queries. Thus, in specifying a complex query, it eases the programming and improves the program readability to specify (abstract) rules as intermediate steps—very much like procedures in conventional programming. Another aspect of rules is the ability to solve simple reasoning tasks.

Referential transparency and answer closedness are essential properties of Xcerpt and visXcerpt, surfacing in various parts of the demonstration. They are two precisely defined traits of the rather vague notion of "declarativity". Referential transparency means that within a definition scope all occurrences of an expression have the same value, i.e., denote the same data. Answer-closedness means that replacing a sub-query in a compound query by a possible single answer always yields a syntactically valid query. Referentially transparent and answer-closed programs are easy to understand (and therefore easy to develop and to maintain), as the unavoidable shift in syntax from the data sought for to the query specifying this data is minimized.

3 Part II: Effectiveness and Efficiency

Currently, two main threads are considered in the Xcerpt project: (1) A careful review of language constructs is underway that aims at an improved effectiveness for query authoring, cf. [6]. Related is a better support for RDF, including proper handling of b-nodes in results and incomplete data specifications. Furthermore, a type system [3] for Xcerpt is under development that eases error detection and recovery. (2) Novel evaluation methods for Xcerpt, enabled by high-level query constructs, are being investigated. Xcerpt's pattern matching is based on simulation unification. An efficient algorithm of simulation unification that is competitive with current main-stream Web query languages both in worst-case complexity and practical performance is described in [4]. The demonstration shows that the employed evaluation algorithm, called "memoization matrix" scales over a large set of query scenarios, empirically confirming the theoretical complexity derived in [4]. Figure 2 shows on the left hand the effect of the memoization on query evaluation time. The right hand side illustrates that the algorithm scales quite nicely over large amounts of data, assuming realistic queries and data (here the Nasa XML dataset from the University of Wash-

ington XML Repository[2] is used). Furthermore, the scalability of basic pattern queries over a broad range of data sizes is illustrated. Finally, the effect of several advanced query constructs is investigated. It is shown that constructs such as `optional` or qualified descendant do not only make queries easier to express and understand, but in many practical cases also more efficient to evaluate. Effects of optionality, injectivity, order, totality, and subterm negation are shown in detailed evaluations.

In further work, optimizations of the rule chaining algorithm are investigated, partially based on dependency analysis provided by the above mentioned type system. Furthermore, rule unfolding and algebraic optimization beyond intermediary construction similar to optimization of nested construction in languages such as XQuery is investigated, cf. [7] for details on the relation of the two.

References

1. J. Bailey, F. Bry, T. Furche, and S. Schaffert, "Web and Semantic Web Query Languages: A Survey," in *Reasoning Web Summer School 2005*, J. Maluszinsky and N. Eisinger, Eds. Springer-Verlag, 2005.
2. S. Berger, F. Bry, and S. Schaffert, "A Visual Language for Web Querying and Reasoning," in *Proc. Workshop on Principles and Practice of Semantic Web Reasoning*, ser. LNCS, vol. 2901. Springer-Verlag, 2003.
3. S. Berger, E. Coquery, W. Drabent, and A. Wilk, "Descriptive Typing Rules for Xcerpt," in *Proc. of Workshop on Principles and Practice of Semantic Web Reasoning*. REWERSE, 2005.
4. F. Bry, A. Schroeder, T. Furche, and B. Linse, "Efficient Evaluation of n-ary Queries over Trees and Graphs," Submitted for publication, 2006.
5. F. Bry, T. Furche, L. Badea, C. Koch, S. Schaffert, and S. Berger, "Querying the Web Reconsidered: Design Principles for Versatile Web Query Languages," *Journal of Semantic Web and Information Systems*, vol. 1, no. 2, 2005.
6. T. Furche, F. Bry, and S. Schaffert, "Initial Draft of a Language Syntax," REWERSE, Deliverable I4-D6, 2006. [Online]. Available: http://rewerse.net/deliverables/m18/i4-d6.pdf
7. B. Linse, "Automatic Translation between XQuery and Xcerpt," Diplomarbeit/-Master thesis, Institute for Informatics, University of Munich, 2006. [Online]. Available: http://www.pms.ifi.lmu.de/publikationen#DA_Benedikt.Linse
8. S. Schaffert, "Xcerpt: A Rule-Based Query and Transformation Language for the Web," Dissertation/Ph.D. thesis, University of Munich, 2004. [Online]. Available: http://www.pms.ifi.lmu.de/publikationen/
9. S. Schaffert and F. Bry, "Querying the Web Reconsidered: A Practical Introduction to Xcerpt," in *Proc. Extreme Markup Languages*, 2004.
10. M. M. Zloof, "Query By Example: A Data Base Language," *IBM Systems Journal*, vol. 16, no. 4, pp. 324–343, 1977.

[2] `http://www.cs.washington.edu/research/xmldatasets/`