

# Querying the standard and Semantic Web using Xcerpt and visXcerpt

Sacha Berger  
University of Munich  
sacha.berger@ifi.lmu.de

François Bry  
University of Munich  
francois.bry@ifi.lmu.de

Sebastian Schaffert  
University of Munich  
wastl@wastl.net

## ABSTRACT

Query and transformation languages such as XPath, XQuery and XSLT have evolved to standard development tools for Web applications. Arguably those languages are not fully suited for Semantic Web applications. The query and transformation languages Xcerpt and visXcerpt have been conceived with both standard Web and Semantic Web applications in mind. They are twin languages both based on the same paradigms and principles. Xcerpt realizes these paradigms and principles textually, visXcerpt visually. A mixed standard Web and Semantic Web application scenario implemented in Xcerpt and visXcerpt is presented. Xcerpt and visXcerpt are ongoing research projects; prototypic implementations of the languages are available.

## 1. PRINCIPLES OF XCERPT AND VISXCERPT

The query and transformation language Xcerpt[2], is a declarative, logic based language. Its salient features are **pattern based query** and **construction** of graph-structured data, possibly **incomplete query patterns** reflecting the *heterogeneity* and the *semistructured* nature of web data, **rules** relating query and construction and **rule chaining** enabling simple inference and query modularization. visXcerpt[1] is merely a hypertext rendering of Xcerpt, a novel approach to realize a visual language.

The following paragraphs present salient features of Xcerpt and visXcerpt.

### 1.1 Twin Languages.

Xcerpt and visXcerpt are two languages based on the same query evaluation paradigms. Depending on the user or usage scenario either of them can be more appropriate – while experienced programmers often feel more comfortable with textual programming, novice programmers might prefer using a visual language. An application first developed using the visual language can without any need for conversion be further developed using the textual language and *vice*

*versa*. visXcerpt has been conceived as a mere Hypertext rendering of Xcerpt using (a slightly extended variant of) the styling language CSS. This approach to developing a visual language is fully new. To the best of the knowledge of the authors, there are no other examples of visual languages conceived this way. This approach yields a visual language tightly connected to a textual language it is a rendering of. Arguably, a visual language conceived as Hypertext application is especially convenient for the Web and the Semantic Web. Arguably this approach of developing visual languages is very promising for Web languages of all kinds.

### 1.2 Query Languages for both, Standard and Semantic Web Applications.

Xcerpt and visXcerpt, short (vis)Xcerpt, have been conceived for querying not only Web meta-data, but also arbitrary Web data. A working hypothesis of the (vis)Xcerpt projects is namely that many applications will refer to both, standard Web and Semantic Web data. Using a single query language well-suited for data of both kinds is preferable to using different languages. This reduces the programming effort and costs and avoids mismatches resulting from inter-operating systems and/or different query paradigms.

### 1.3 Languages Capable of Inference.

(vis)Xcerpt are query languages capable of inference. The inferences (vis)Xcerpt can perform are limited to simple inference like needed in querying database views, in logic programming, and in usual forms of Semantic Web reasoning. Offering both, inference and querying, in the same languages avoids the impedance mismatch commonly arising when querying and inferencing are performed separately.

### 1.4 Rule-based Languages.

Rules are understood here as means to specify novel, maybe virtual data in terms of queries, i.e., what is called “views” in (relational) databases, regardless of whether data are materialized or not. Views, i.e., rule-defined data are desirable for both, conventional and Semantic Web applications. There are three reasons for this. First, rules are a mean for achieving the so-called “separation of concern”, i.e., the stepwise specifications of data to retrieve and/or to construct. In other words, rules are a means for “procedural abstraction”, i.e. rules (view definitions, resp.) are the counterpart of functions and/or procedures. Second, rules and view definitions give rise to easily specifying inference methods needed, e.g., by Semantic Web applications. Third, rules are means for “data mediation”. Data mediation means translating to a common format data from different sources. Data media-

tion is needed on both the standard Web and the Semantic Web because of their heterogeneity. Data mediation is especially needed on the Semantic Web because of many formats e.g. for serializing RDF data.

## 1.5 Referentially Transparent and Answer-Closed Languages.

Referential transparency means that, within a definition scope, all occurrences of an expression have the same value, i.e., denote the same data. Referential transparency is an essential, precisely defined trait of the rather vague notion of “declarativity”. Referentially transparent programs are easier to understand and therefore easier to develop, to maintain, and to optimize. Answer-closedness means that replacing a sub-query in a compound query by a possible single answer always yields a syntactically valid query. Answer-closed query languages ensure in particular that every data item, i.e. every possible answer to some query, is a syntactically valid query. Answer-closedness is the distinguishing property of the milestone visual query language “Query By Example” [3]. Logic programming languages, such as Datalog are in general answer-closed, while SQL is not answer-closed. Answer-closedness eases the specification of queries because it keeps limited the unavoidable shift in syntax from the data sought for, i.e., the expected answer, and the query specifying these data.

The demonstration of (vis)Xcerpt will show how the principles given above are realized in a textual language, Xcerpt, and in its twin visual language visXcerpt.

## 2. DEMONSTRATIONS OF (VIS)XCEPT: ONTOLOGY-BASED DATA RETRIEVAL

The application scenario demonstrated is the organization of a party. A clique of friends has to be determined and some music has to be selected based on the music genres the friends prefer. The data are inspired by the “Friend of a Friend” application (cf. <http://xmlns.com/foaf/0.1/>). In addition RDF, OWL and XML documents list music genres, relationship between music genres, music and CDs owned by the friends. The demonstration illustrates the following aspects of (vis)Xcerpt: Standard Web and Semantic Web data are retrieved using the same query languages, (vis)Xcerpt. Meta-data formatted in various Semantic Web formats are conveniently retrieved using (vis)Xcerpt. (vis)Xcerpt queries and answer constructions are expressed using patterns that are intuitive and easy to express. Recursive (vis)Xcerpt programs are presented and evaluated demonstrating that (vis)Xcerpt gives rise to a rather simple expression of transitive closures of Semantic Web relations and of recursive traversal of nested Web documents. The following paragraphs sketch the (vis)Xcerpt demonstration.

### 2.1 Xcerpt – the Textual Query Language.

Assuming ontology-based inference and mediation rules for the “Friend of a Friend” data, a graph structured view of the network of friends is queried. The view consists of **person** elements (each containing a **name** element) referencing other **person** elements as friends. In XML graphs are represented as trees containing (e.g. ID/IDREF) references. In contrast to XQuery, (vis)Xcerpt has built-in dereferencing and reference circle detection. Therefore with (vis)Xcerpt other **persons** can be seen as sub-elements of a person and

no special treatment of circular structures is necessary. The rule can be compared to a Datalog rule using a variable to relate query and construction parts. Xcerpt has two textual syntaxes that are totally interchangeable: a compact user-friendly syntax and an XML syntax (not illustrated in this article). Experiences with e.g. XSLT have demonstrated the need for compact, non-XML syntax. Incompleteness in the query pattern (expressed by **desc** in depth, and double braces in breadth of the term structure) is necessary in queries, as the exact structure of **person** elements is varying.

### 2.2 visXcerpt – the Visual Query Language.

The visual counterpart of the textual rule uses nested rectangles with name tabs to represent the nested term structure. Colors depending on the nesting depth help recognizing structures while browsing complex patterns. visXcerpt has **interactive features** helping for a quick understanding of large programs: boxes representing XML elements can be folded and unfolded and semantically related portions of programs (e.g. different occurrences of the same variable), can be highlighted. References (e.g. ID/IDREF references) can be followed back and forth as Hypertext links. visXcerpt programs can be composed using a **novel Copy-and-Paste paradigm** specifically designed for tree (or term) editing. **Patterns** are provided as templates to support easy construction of visXcerpt programs without in-depth prior knowledge of visXcerpt’s syntax. The demonstration shows how one, while programming, can switch from Xcerpt to visXcerpt and *vice versa*.

### 2.3 Xcerpt and visXcerpt for the Semantic Web.

Inference based on statements and their properties is a common task for querying Semantic Web data. A music ontology consisting of statements like `statement["Swing", rdf:subClassOf, "Jazz"]` is part of a hierarchy of music genres. The transitive property of the `rdf:subClassOf` predicate is easily expressed with a rule. The demonstration aims at showing that Semantic Web properties are easily expressed by rules.

## 3. ACKNOWLEDGMENTS

This research has been funded by the European Commission and by the Swiss Federal Office for Education and Science within the 6th Framework Programme project REVERSE number 506779 (cf. <http://reverse.net>).

## 4. ADDITIONAL AUTHORS

Additional authors: Tim Furche (University of Munich, email: [tim@furche.net](mailto:tim@furche.net)) and Oliver Bolzer (University of Munich, email: [bolzer@cip.ifi.lmu.de](mailto:bolzer@cip.ifi.lmu.de)) and Christoph Wieser (University of Munich, email: [wieser@cip.ifi.lmu.de](mailto:wieser@cip.ifi.lmu.de)).

## 5. REFERENCES

- [1] S. Berger, F. Bry, S. Schaffert, and C. Wieser. Xcerpt and visXcerpt: From Pattern-Based to Visual Querying of XML and Semistructured Data. In *29th Intl. Conference on Very Large Data Bases*, 2003.
- [2] S. Schaffert and F. Bry. Querying the Web Reconsidered: A Practical Introduction to Xcerpt. In *Extreme Markup Languages*, 2004.
- [3] M. M. Zloof. Query-by-example: A data base language. *IBM Systems Journal*, 16(4):324–343, 1977.