



Doctoral Thesis

A meaning driven querying methodology end-user oriented querying

Author(s):

Kapetanios, Epaminondas

Publication Date:

2000

Permanent Link:

<https://doi.org/10.3929/ethz-a-004127788> →

Rights / License:

[In Copyright - Non-Commercial Use Permitted](#) →

This page was generated automatically upon download from the [ETH Zurich Research Collection](#). For more information please consult the [Terms of use](#).

Diss. ETH No. 13937

A Meaning Driven Querying Methodology: End-user Oriented Querying

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY (ETH) Zurich

for the degree of Doctor of Technical Sciences

presented by

EPAMINONDAS KAPETANIOS

Diploma in Computer Science,
Univ. of Athens, Hellas
born October 8, 1963
citizen of Athens, Hellas

accepted on the recommendation of

Prof. Dr. M. C. Norrie, examiner
Prof. Dr. H. Hinterberger, co-examiner

2000

Abstract

Providing information out of data has always been the major concern in information science. Addressing and disseminating the acquired information, however, has been the major focus of research activities in query languages and question answering systems. Despite the variety of approaches taken so far, two central questions can be raised when we ask for information: *how* to ask for information and *what* kind of queries/questions the system can answer.

Although the second question has been exhaustively examined by theories and/or practical solutions referring to model- or proof-theoretic semantics, closed versus open world assumptions, fuzziness or uncertainty of query results, etc., the way *how* to ask for available information still remains a major challenge, especially when end-users are involved in the querying process. The major presupposition of formulating a reasonable query in terms of database or Web data query languages used as programming tools is a) the familiarity with the syntax formalisms of the query languages which are rather complex for an end-user, b) understanding of the semantics of the model underlying the data/knowledge repository.

In order to improve usability and alleviate end-users from syntax-based formulation of queries, advanced (visual) query interfaces and/or languages as well as interactive query formulation techniques have been proposed which provide a syntax-free (visual) formalism for query formulation. The latter mostly relies on graphical presentations of conceptual models which turn out to be very tedious to handle and difficult to understand, especially when large knowledge repositories and/or complex data schemas are concerned. Additionally, the role of *semantics* defined as the relationship of linguistic symbols and their meaning as well as a constraints-based usage of query terms and results has not been the case. Furthermore, in many cases the end-user does not really know what is relevant to her/him or how to make use of the available knowledge in order to formulate a query.

In this sense, the *meaning driven querying methodology* presented in this thesis changes the perspective of end-user oriented query construction in that the *system guides the end-user* to the construction of a reasonable query through consideration of *semantics/meaning* of vocabulary terms. The meaning of queries is expressed in terms of a) constraints-based consideration of terms within a query, i.e. rejection of mutually exclusive terms, b) the *intensional meaning* or *interpretation* of query terms, i.e. talking about the same things but with different interpretations, c) the appearance of query terms in a particular natural language which enhance the understanding of the query terms. Meaning is represented in a *knowledge base* and provide the set of alphabet terms for the designated query language MDDQL (Meaning Driven Data Query Language). In addition to the knowledge base as a repository of the query alphabet terms, an *inference engine* and *visual query interface(s)* are used as an interaction platform or communication blackboard between end-user and machine.

The inference engine incorporates the interaction mode between system and end-user conceived as an *abstract machine* which receives as input the current query context (current state of a query) and produces a semantically meaningful output (subsequent/extended state of the query). The decision on which terms should be further considered for a potential extension of the query expression is based on the *semantics/meaning* of the application domain and/or operational terms which is represented in the knowledge base for the vocabulary. Furthermore, since terms might appear in more than one natural language, it is possible to construct a query expression in two or more natural languages and still receive the same query results. This is due to the separation between the world of symbols as used for the

representation and structuring of data/knowledge repositories and the world of multi-lingual vocabularies as used for the formulation of queries in MDDQL.

Given that the construction mode of a query resembles the definition of an abstract machine, a constructed query is an already parsed one. This is in contrast with other parsing approaches where a query needs to be formulated first and subsequently parsed. Therefore, no additional parsing after the interactive construction is needed, a feature which is particularly valuable when multi-lingual environments of end-users are considered. Otherwise, more than one parser needs to be implemented, according to the number of natural languages supported.

Abstract

Die Herleitung von Informationen aus Daten war immer einer der wichtigsten Schwerpunkte in den Informationswissenschaften. Das Adressieren und Verbreiten der gewünschten Information ist der Gegenstand der Forschungs- und Entwicklungsaktivitäten von Abfragesprachen und Fragenbeantwortende Systeme. Trotz der Vielfalt der bisher verfolgten Ansätze und Philosophien, zwei zentrale Aspekte stellen sich, wenn man nach Information sucht: a) *wie* soll man nach Information fragen und b) *welche* Fragen kann ein System beantworten.

Obwohl der zweite Aspekt ausreichend durch Theorien und praktische Lösungen bezüglich modell- oder Beweis-theoretischer Semantiken, "closed vs. open world assumptions", Fuzziness oder Unsicherheit von Abfrageresultaten, usw., untersucht wurde, der erste Aspekt, wie man nach verfügbaren Informationen fragt, stellt immer noch Herausforderungen, insbesondere wenn Endbenutzer beim Abfrageprozess involviert sind. Die wichtigsten Voraussetzungen bei der Formulierung einer bedeutungsvollen Abfrage in einer Datenbank- oder Web-daten-spezifischen Abfragesprache, die als Programmierwerkzeuge gedacht und benutzt werden, sind a) das Erlernen von Syntaxformalismen der jeweiligen Abfragesprache, die für den Endbenutzer komplex sind, b) das Verstehen der Semantiken des Datenmodells, das dem Daten- oder Wissensrepository zugrundeliegt.

Um die Nutzbarkeit eines Systems zu erhöhen und die Aufgabe der syntax-basierten Formulierung einer Abfrage von Endbenutzern zu erleichtern, es wurden sowohl visuelle Abfrageschnittstellen und/oder Programmiersprachen als auch interaktive Abfrageformulierungstechniken vorgeschlagen und entwickelt, die auf einen syntaxfreien (visuellen) Formalismus für die Abfrageformulierung basieren. Diese beziehen sich, in den meisten Fällen, auf graphische Darstellungen des konzeptuellen Modells, die relativ schwierig zu handhaben und zu interpretieren sind, insbesondere wenn grosse Wissensrepositorien und/oder komplexe Datenschemata zu berücksichtigen sind. Darüber hinaus ist die Rolle der Semantiken, die sowohl durch die Beziehung linguistischer Symbole und deren Bedeutung als auch durch die constraints-basierte Benutzung der Abfrageterme und Abfrageresultate definiert ist, nicht von zentraler Bedeutung. Diese Problematik ist auch durch die Tatsache verstärkt, dass Endbenutzer meistens nicht wissen was ist relevant oder wie sie verfügbares Wissen zu gebrauchen ist, um eine Abfrage zu formulieren.

Die *Bedeutungsorientierte Abfragemethodologie*, die in dieser Dissertation präsentiert wird, öffnet die Perspektive einer Endbenutzerorientierten Abfragekonstruktion, in dem das System den Endbenutzer bei der Konstruktion einer Bedeutungsvollen Abfrage durch Berücksichtigung von Semantiken/Bedeutungen der Begriffe des benutzten Vokabulars unterstützt. Die Bedeutung einer Abfrage

wird ausgedrueckt durch a) die constraints-basierte Einbettung der Begriffe in einer Abfrage, z.B. Ablehnung von bedeutungslosen Abfragen, b) die intensionale Bedeutung oder Interpretation der Abfragebegriffe, d.h. aehnliche Begriffe mit unterschiedlichen Interpretationen behandeln, c) die Erscheinung der Begriffe in der Abfrage in einer vorausgewaehlt natuerlichen Sprache, die das Verstehen der verwendeten Symbole des Implementierungsmodells ermoeglicht oder erhoehrt.

Begriffe und derer Bedeutung ist in einer Wissensbasis repraesentiert, die die Menge aller dem Alphabet der Abfragesprache MDDQL gehoerenden Begriffe bestimmt. Darueber hinaus existieren zwei weitere Komponenten, die den Abfragekonstruktionsmechanismus von MDDQL realisieren: ein Inferenzmechanismus und die visuellen Abfrageschnittstellen als Kommunikationsplattform zwischen einem Endbenutzer und dem System.

Der Inferenzmechanismus verkoerpert den zugrundegelegten Interaktionsmodus zwischen Endbenutzer und System, der als eine *abstrakte Maschine* aufgefasst und konzipiert ist. Diese erhaelt als Eingabe den aktuellen Abfragekontext (aktueller Zustand der Abfrage) und produziert eine semantisch bedeutungsvolle Ausgabe (anderer Abfragezustand, erweiterte Abfrage). Die Entscheidung, welche Begriffe sollen weiter fuer eine potentielle Erweiterung der Abfrage beruecksichtigt werden, haengt von den zugrundegelegten, in der Wissensbasis repraesentierten Semantiken/Bedeutungen sowohl der Begriffe einer Applikationsdomaene als auch der Begriffe, die Operationen zum Ausdruck bringen. Aufgrund der Tatsache, dass Begriffe in mehr als eine natuerliche Sprache formuliert werden koennen, es ist moeglich die selbe Abfrage in verschiedenen natuerlichen Sprachen zu formulieren ohne Verzerrung des Abfrageresultats. Diese Eigenschaft wird durch die Trennung zwischen Implementierungssymbolen in einer Datenbank oder Web-daten und dem Abfragevokabular ermoeglicht.

Eine durch den als abstrakte Maschine spezifizierten Interaktionsmodus eine in MDDQL konstruierte Abfrage gilt als eine Abfrage, die schon grammatikalisch analysiert ist. Diese Eigenschaft steht im Gegesatz zu den ueblichen Parsingmechanismen, die eine Abfrage erst nach dem Formulierungsabschluss parsen. Diesbezieglich, es wird kein zusaetzliches Parsen der Abfrage benoetigt, das, im Fall einer mehrsprachigen Benutzerumgebung, die Verfuegbarkeit mehrerer Parsingmechanismen, d.h. ein Parsingmechanismus pro natuerliche Sprache, fuer die selbe Abfrage erfordert haette.