



Doctoral Thesis

A generic agent framework for Internet information systems

Author(s):

Erni, Antonia

Publication Date:

2000

Permanent Link:

<https://doi.org/10.3929/ethz-a-004016455> →

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. 13645

A Generic Agent Framework for Internet Information Systems

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZÜRICH

for the degree of
Doctor of Technical Sciences

presented by

ANTONIA ERNI

Dipl. Informatik-Ing. ETH
born November 1, 1969
citizen of Scuol (GR), Trin (GR)

accepted on the recommendation of

Prof. Dr. M.C. Norrie, examiner
Prof. Dr. M. Reiser, co-examiner

2000

Abstract

With the increasing popularity of the World-Wide Web, more and more data is accessible over the Internet. The Web is no longer used only for information presentation, but also for real business applications. This results in the availability of more data-intensive Web sites with data managed in database systems. In order to present database information to users, Web documents are often generated dynamically on demand.

Having detected the necessity of providing Web access to database management systems (DBMS) quite a number of interfaces to commercial and research DBMS have already been developed. Nowadays it is no longer a question of *how* to provide Web database access, but rather the *time to market* is important in order to compete in the growing electronic market place. Additionally, given the typical delays in Web access to information currently experienced, it is another step to ensure an *effective information service* in terms of performance and user satisfaction. In spite of efforts to improve network communications, delays due to network and server overloading are likely to remain with us for some time to come.

In this thesis, we propose and implement an agent-based framework called *Internet OMS* for the development of integrated, multimedia information services for Web databases and contribute in the Web database area with three ways. First, the framework allows fast prototyping of Web information systems using a plug-and-play architecture and a graphical user interface design tool. Second, we gain in performance accessing database data through agents managing caches on different levels and providing value-added services to both casual and regular users. And third we validate our concepts by developing a performance model for our framework architecture and compare it with the most common gateway interfaces (CGIs) used to provide Web access to databases.

To support fast prototyping of Web information services, we provide developers with database components which can be downloaded and executed on the client side through a normal Web browser. These components allow developers to define a database schema, update and change the database as well as browse through the data by means of a generic browser. Furthermore, the appearance of the data can be defined and changed using the presentation editor of our framework thereby allowing the definition of user interfaces according to application requirements. The generic presentation of the data can be changed by drag and drop operations on objects, by changing colours and fonts and by specifying whether HTML code should be generated for showing the object in the browser or if it should be presented as a Java-Applet frame.

For effective Web database services, it is essential that the information requirements of reg-

ular users can be met without the typical delays currently experienced using Web browsers. We developed cooperating agents to manage both client and server caches, thereby bringing significant performance improvements. The caching and prefetching of information is based on user profiles and agents communicate to ensure the currency of client caches. According to specific application requirements, various forms of agents can be installed on the server and client sides to provide value-added services to both casual and regular users.

For framework validation, we use a quantitative approach which allows us to analyse our agent-based architecture and compare it with the most common architecture, the CGI approach, used to provide access to databases over the Web. The quantitative approach is based on the development of performance models used for capacity planning. It provides a uniform and formal way for dealing with performance issues and is based on the theory of queuing networks. For the architectures to be analysed, we provide a workload model which captures the resource demands and workload intensity characteristics for each component of the system, and a performance model which provides user response time as a function of the system description and the workload parameters. In addition, we present performance results for the different architectures measured in simulating the systems with collected user query traces.

Zusammenfassung

Im Zuge der immer wachsenden Popularität des World-Wide Web steigt die im Internet verfügbare Informationsmenge stark an. Das Web wird in zunehmendem Masse nicht nur für Informationsdarstellung, sondern für die Abwicklung und Abbildung realer Geschäftsvorgänge und -anwendungen verwendet. Als Ergebnis dieser gewaltigen Datenmengen wird immer mehr Information von Web Dokumenten in Datenbanksystemen abgespeichert und verwaltet. Um diese Datenbankinformation den Benutzern zur Verfügung zu stellen, wird eine steigende Anzahl von Web Dokumenten auf Anfrage dynamisch generiert.

Aus der Notwendigkeit, auf Datenbankmanagementsysteme (DBMS) über das Web zuzugreifen zu müssen, sind eine Vielzahl von Schnittstellen zu kommerziellen und wissenschaftlichen Datenbanken entstanden. Es ist nicht länger eine Frage, wie der Zugriff auf Datenbanksysteme über das Web erreicht wird, sondern wie sich die Prämisse “time to market” am effizientesten umsetzen lässt, um in dem regelrecht explodierenden elektronischen Markt wettbewerbsfähig agieren zu können. In Anbetracht der typischen Verzögerungen bei den Zugriffszeiten auf Information im Web ist es zudem eine absolute Notwendigkeit, einen hoch effizienten Informationsservice in Bezug auf Performance und Kundenzufriedenheit in Bedienung und Anwendung zu gewährleisten. Trotz den allgemeinen Anstrengungen, die Netzwerkverbindungen zu verbessern, ist es wahrscheinlich, dass noch geraume Zeit mit Verzögerungen wegen Überlastung der Netzwerke und Server gerechnet werden muss.

In dieser Dissertation schlagen wir ein Agenten-basiertes Framework namens Internet OMS für die Entwicklung von integrierten Multimedia Informationsdiensten für Web Datenbanken vor. Auf dem Gebiet von Web Datenbanken leisten wir drei verschiedene Beiträge: Erstens erlaubt unser Framework ein schnelles Prototyping von Web Informationssystemen unter Anwendung einer “plug and play” Architektur und eines graphischen Entwicklungstools für Benutzerschnittstellen. Zweitens gewinnen wir an Performance beim Zugriff auf Datenbanken durch den Einsatz von Agenten, welche Caches auf verschiedenen Ebenen verwalten und dabei Mehrwertdienste gegenüber gelegentlichen und regelmässigen Benutzern bieten. Drittens validieren wir unsere Konzepte, indem wir ein Performance Modell für unsere Framework Architektur entwickeln, welche wir mit dem am häufigsten verwendeten Interface, dem Common Gateway Interface (CGI), vergleichen.

Wir erreichen “fast prototyping” von Web Informationsdiensten, indem wir Entwicklern Datenbankkomponenten zur Verfügung stellen, welche auf der Kundenseite mit einem normalen Web Browser heruntergeladen und ausgeführt werden können. Diese Komponenten ermöglichen es dem Entwickler, ein Datenbank-Schema zu definieren, die Datenbank oder ihren Inhalt zu verändern, sowie die Daten durch einen generischen Browser dem Benutzer

direkt zur Verfügung zu stellen. Darüber hinaus ist es möglich, die Darstellung der Daten durch Einsatz eines Editors zu Präsentationszwecken zu definieren und so zu verändern, dass die Anforderungen der jeweiligen Applikation erfüllt werden. Diese generische Darstellung von Daten kann durch “drag and drop” Operationen an Objekten und durch das Wechseln von Farben und Schriftarten verändert werden. Zudem kann zwischen HTML Code zur Generation von Objekten im Browser und der Präsentation durch Java Applet Frames gewählt werden.

Für effiziente Web Datenbankdienste ist es essentiell, dass die Informationsanforderungen der regelmässigen Benutzer von Web Browsern ohne die heute typischen Verzögerungen erfüllt werden können. Wir haben kooperierende Agenten entwickelt, welche gleichzeitig Client und Server Caches verwalten und dadurch signifikante Performance Verbesserungen erzielen. Dieses Caching und Vorgeifen auf Information basiert auf einer intensiven Nutzung der Benutzerprofile und auf Agenten, welche untereinander kommunizieren, um die Aktualität der Client Caches zu gewährleisten. Um wertsteigernde Dienste gegenüber gelegentlichen und regulären Benutzern zu ermöglichen, können, gemäss spezifischen Applikationsanforderungen, verschiedene Formen von Agenten auf dem Server und beim Client installiert werden.

Zur Validierung des Frameworks bedienen wir uns einer quantitativen Vorgehensweise, welche es erlaubt, unsere auf Agenten basierende Architektur zu analysieren und sie anschliessend mit der am häufigsten angewandten Architektur, der CGI-Technologie, zu vergleichen. Diese quantitative Analyse nutzt Performance Modelle, wie sie in der Kapazitätsplanung zur Anwendung kommen. Diese Modelle ermöglichen eine uniforme und formelle Vorgehensweise, um mit Performance Problemen zu arbeiten und basieren auf der Warteschlangentheorie. Zur Analyse der Architektur bedienen wir uns eines “workload models”, welches die Anforderungen an die Ressourcen und die Charakteristiken im Bezug auf die Arbeitslastintensität für jede einzelne Komponente des Systems erfasst. Zudem kommt ein “performance model” zur Anwendung, welches die Antwortzeit als eine Funktion der Arbeitslast darstellt. Darüber hinaus stellen wir Ergebnisse von Performance-Messungen dar, welche durch die Simulation von Benutzerabfragen auf den verglichenen Web-Datenbankarchitekturen entstanden sind.