Brian Read (Ed.)

LNCS 2097

Advances in Databases

18th British National Conference on Databases, BNCOD 18
Chilton, UK, July 2001
Proceedings

Springer

Brian Read (Ed.)

Advances in Databases

18th British National Conference on Databases, BNCOD 18
Chilton, UK, July 9-11, 2001
Proceedings

Springer

Series Editors

Gerhard Goos, Karlsruhe University, Germany
Juris Hartmanis, Cornell University, NY, USA
Jan van Leeuwen, Utrecht University, The Netherlands

Volume Editor

Brian Read
CLRC Rutherford Appleton Laboratory
Information Technology Department
Chilton, Didcot, Oxfordshire OX11 0QX
United Kingdom
E-mail: Brian.Read@rl.ac.uk

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Advances in databases : proceedings / 18th British National Conference on Databases, BNCOD 18, Chilton, UK, July 9 - 11, 2001. Brian Read (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; Vol. 2097)
ISBN 3-540-42265-X

CR Subject Classification (1998): H.2-4

ISSN 0302-9743
ISBN 3-540-42265-X Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna
Printed on acid-free paper SPIN: 10839443 06/3142 5 4 3 2 1 0

Foreword

The ever-expanding growth of Information Technology continues to place fresh demands on the management of data. Database researchers must respond to new challenges, particularly to the opportunities offered by the Internet for access to distributed, semi-structured and multimedia data sources.

This volume contains the proceedings of the 18^{th} British National Conference on Databases (BNCOD 2001), held at the Rutherford Appleton Laboratory in July 2001. In recent years, interest in this conference series has extended well beyond the UK. In selecting just eleven of the submitted papers for presentation, the programme committee has included contributors from The Netherlands, Germany, Sweden, Canada and USA. In addition, two specially invited speakers address subjects of topical interest.

Our first invited speaker is Professor Dr. Rudi Studer from the University of Karlsruhe. At AIFB, the Institute for Applied Informatics and Formal Description Methods, he and his colleagues are in the forefront of work on the Semantic Web. This aims to make information accessible to human and software agents on a semantic basis. The paper discusses the role that semantic structures, based on ontologies, play in establishing communication between different agents. The AIFB web site has been developed as a semantic portal to serve as a case study.

The massive increase in data volumes from big science such as remote sensing and high energy physics means that we now contemplate the storage and processing of petabytes. Grid technology, specifically the „Data Grid“ is seen as attractive. It is thus timely that our second invited speaker addresses strategy in this field. He is Professor Tony Hey, now recently appointed as Director of the UK e-Science Core Programme and well placed to expound the vision.

The contributed papers are presented in four groups. The first of these addresses performance and optimisation. This issue has always been at the core of database technology. The first paper, by Regan and Delis, reports on a practical study of space management in logs. They evaluate a technique for reclaiming log space from short transactions while retaining recoverability for long running ones. The increasing popularity of XML presents new challenges. Zhu and Lü propose an algorithm for an effective storage placement strategy for XML documents that facilitates their efficient parallel processing. The trade off between data quality and performance is an interesting topic tackled by Caine and Embury. They study algorithms for integrity checking delayed from when the system is too busy to off-peak, „lights out“ hours.

The second group of papers concentrates on objects in databases and software engineering. The great variety of CASE tools prompt the adoption of standardised meta-models and transfer formats. In proposing an extension to OCL, Gustavsson and Lings further the interchange of models by defining a common, model independent notation for design transformations. Next, Zhang and Ritter investigate the state of database support for software development using object-oriented programming languages. They highlight the shortcomings in this respect of the current object-relational database paradigm and suggest how it might beneficially be enhanced. The third paper returns to the engineering design environment and tackles concurrent version control. Al-Khudair, Gray and Miles present a generalised object-oriented model that captures the evolution of design configurations and their components by supporting versioning at all levels.

In the third group of papers, we again consider optimisation. More specifically, contributors consider efficient querying in the newer domains of multimedia and distributed data sources. The requirements and techniques of the worlds of information retrieval and transactional databases are very different. The Dutch team of Blok, de Vries, Blanken and Apers present a case study on the „top-N" queries familiar in content retrieval in the context of a database approach to the management of multimedia data. The key issues addressed, such as speed and quality of answers and the opportunities for scalability are supported by experimental results. A similar problem is of concern to Sattler, Dunemann, Geist, Saake and Conrad. They seek control over the potentially excessive data returned from a query over heterogeneous data sources. By extensions to multi-database languages, they explore ways of asking for just the „first n" results, or of asking for a *sample* of the complete result. Still with the theme of information systems relying on database technology, Waas and Kersten are concerned with a web multimedia portal based on the Monet database system. Here the optimisation challenge is query throughput. The authors report on the performance of a simple and robust scheme for the scheduling of queries in a large, parallel, shared-nothing database cluster.

The two papers in our final group are both about querying objects. However, they are very different. Trigoni and Bierman present an inference algorithm for OQL that identifies the most general type of a query in the absence of schema type information. This is relevant to where heterogeneity is encountered – for example, in any open, distributed, or even semi-structured, database environment. Distributed databases and virtual reality are combined in the ambitious work reported by Ammoura, Zaiane and Ji. They explore data mining in a virtual data warehouse. Rendering multi-dimensional data aggregates as objects, the user flies through the data to explore and query different views.

Acknowledgements

The members of the programme committee, under the energetic leadership of Carole Goble, reviewed the submitted papers with critical thoroughness. Their enthusiastic commitment to the continued success of BNCOD is much appreciated. Susan Hilton and her team played an essential part in the practical organisation. Thanks are also due to Alex Gray and the steering committee for the invitation to host the conference and valuable guidance.

April 2001 Brian Read

Conference Committees

Programme Committee

Carole Goble (Chair)	University of Manchester
David Bell	University of Ulster
Richard Connor	University of Strathclyde
Richard Cooper	University of Glasgow
Suzanne Embury	University of Wales, Cardiff
Carole Goble	University of Manchester
Alex Gray	University of Wales, Cardiff
Mike Jackson	University of Wolverhampton
Anne James	Coventry University
Keith Jeffery	CLRC Rutherford Appleton Laboratory
John Keane	UMIST
Graham Kemp	University of Aberdeen
Jessie Kennedy	Napier University
Brian Lings	University of Exeter
Rob Lucas	Keylink Computers Ltd
Mark Levene	University College London
Nigel Martin	Birkbeck College, University of London
Werner Nutt	Heriot-Watt University
Norman Paton	University of Manchester
Alex Poulovassilis	Birkbeck College, University of London
Brian Read	CLRC Rutherford Appleton Laboratory
Nick Rossiter	University of Newcastle upon Tyne
Michael Worboys	Keele University

Additional Referees

Carsten Butz	Heriot-Watt University
Alvaro Fernandes	University of Manchester
Tony Griffiths	University of Manchester
Paulo Pinheiro da Silva	University of Manchester
Sandra Sampaio	University of Manchester

Organising Committee

Brian Read (Chair)	CLRC Rutherford Appleton Laboratory
Susan Hilton	CLRC Rutherford Appleton Laboratory
Kevin O'Neill	CLRC Rutherford Appleton Laboratory

Steering Committee

Alex Gray (Chair)	University of Wales, Cardiff
Nick Fiddian	University of Wales, Cardiff
Carole Goble	University of Manchester
Peter Gray	University of Aberdeen
Roger Johnson	Birkbeck College, University of London
Jessie Kennedy	Napier University
Brian Lings	University of Exeter
Michael Worboys	Keele University

Table of Contents

Invited Papers

Performance and Optimisation

Objects - Design & Development

Query Optimisation

Querying Objects

SEAL — A Framework for Developing SEmantic Web PortALs

Alexander Maedche[1,3], Steffen Staab[1,2], Nenad Stojanovic[1], Rudi Studer[1,2,3], and York Sure[1,2]

[1] Institute AIFB, University of Karlsruhe, D-76128 Karlsruhe, Germany
http://www.aifb.uni-karlsruhe.de/WBS
{ama,sst,nst,rst,ysu}@aifb.uni-karlsruhe.de
[2] Ontoprise GmbH, Haid-und-Neu Straße 7, 76131 Karlsruhe, Germany
http://www.ontoprise.de
[3] FZI Research Center for Information Technologies, Haid-und-Neu Straße 10-14, 76131 Karlsruhe, Germany
http://www.fzi.de/wim

Abstract. The core idea of the Semantic Web is to make information accessible to human and software agents on a semantic basis. Hence, web sites may feed directly from the Semantic Web exploiting the underlying structures for human and machine access. We have developed a generic approach for developing semantic portals, *viz.* SEAL (SEmantic portAL), that exploits semantics for providing and accessing information at a portal as well as constructing and maintaining the portal.
In this paper, we discuss the role that semantic structures make for establishing communication between different agents in general. We elaborate on a number of intelligent means that make semantic web sites accessible from the outside, *viz.* semantics-based browsing, semantic querying and querying with semantic similarity, and machine access to semantic information at a semantic portal. As a case study we refer to the AIFB web site — a place that is increasingly driven by Semantic Web technologies.

1 Introduction

The widely-agreed core idea of the Semantic Web is the delivery of data on a semantic basis. Intuitively the delivery of semantically apprehended data should help with establishing a higher quality of communication between the information provider and the consumer. How this intuition may be put into practice is the topic of this paper.

We discuss means to further communication on a semantic basis. For this one needs a theory of communication that links results from semiotics, linguistics, and philosophy into actual information technology. We here consider *ontologies* as a sound semantic basis that is used to define the meaning of terms and hence to support intelligent access, *e.g.* by semantic querying [5] or dynamic hypertext views [19].

Thus, ontologies constitute the foundation of our SEAL (SEmantic portAL) approach. The origins of SEAL lie in Ontobroker [5], which was conceived for

B. Read (Ed.): BNCOD 2001, LNCS 2097, pp. 1–22, 2001.

semantic search of knowledge on the Web and also used for sharing knowledge on the Web [2]. It then developed into an overarching framework for search and presentation offering access at a portal site [19]. This concept was then transferred to further applications [1,21,24] and is currently extended into a commercial solution[1].

We here describe the SEAL core modules and its overall architecture (Section 3). Thereafter, we go into several technical details that are important for human and machine access to a semantic portal.

In particular, we describe a general approach for semantic ranking (Section 4). The motivation for semantic ranking is that even with accurate semantic access, one will often find too much information. Underlying semantic structures, *e.g.* topic hierarchies, give an indication of what should be ranked higher on a list of results.

Finally, we present mechanisms to deliver and collect machine-understandable data (Section 5). They extend previous means for better digestion of web site data by software agents. Before we conclude, we give a short survey of related work.

2 Ontology and Knowledge Base

For our AIFB intranet, we explicitly model relevant aspects of the domain in order to allow for a more concise communication between agents, *viz.* within the group of software agents, between software and human agents, and — last not least — between different human agents. In particular, we describe a way of modeling an ontology that we consider appropriate for supporting communication between human and software agents.

2.1 Ontologies for Communication

Research in ontology has its roots in philosophy dealing with the nature and organisation of being. In computer science, the term ontology refers to an engineering artifact, constituted by a specific vocabulary used to describe a particular model of the world, plus a set of explicit assumptions regarding the intended meaning of the words in the vocabulary. Both, vocabulary and assumptions, serve human and software agents to reach common conclusions when communicating.

Reference and meaning. The general context of communication (with or without ontology) is described by the meaning triangle [15]. The meaning triangle defines the interaction between symbols or words, concepts and things of the world (*cf.* Figure 1).

The meaning triangle illustrates the fact that although words cannot completely capture the essence of a reference (= concept) or of a referent (= thing), there is a correspondence between them. The relationship between a word and

[1] cf. http://www.time2research.de

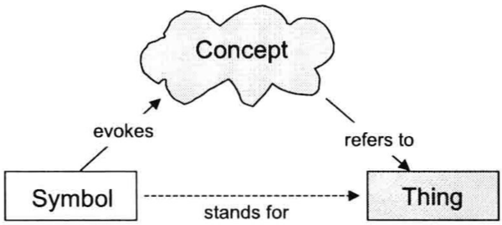

Fig. 1. The Meaning Triangle

a thing is indirect. The correct linkage can only be accomplished when an interpreter processes the word invoking a corresponding concept and establishing the proper linkage between his concept and the appropriate thing in the world.

Logics. An ontology is a general logical theory constituted by a vocabulary and a set of statements about a domain of interest in some logic language. The logical theory specifies relations between signs and it apprehends relations with a semantics that restricts the set of possible interpretations of the signs. Thus, the ontology reduces the number of mappings from signs to things in the world that an interpreter who is committed to the ontology can perform — in the ideal case each sign from the vocabulary eventually stands for exactly one thing in the world.

Figure 2 depicts the overall setting for communication between human and software agents. We mainly distinguish three layers: First of all, we deal with things that exist in the real world, including in this example human and software agents, cars, and animals. Secondly, we deal with symbols and syntactic structures that are exchanged. Thirdly, we analyze models with their specific semantic structures.

Let us first consider the left side of Figure 2 without assuming a commitment to a given ontology. Two human agents HA_1 and HA_2 exchange a specific sign, *e.g.* a word like "jaguar". Given their own internal model each of them will associate the sign to his own concept referring to possibly two completely different existing things in the world, *e.g.* the animal *vs.* the car. The same holds for software agents: They may exchange statements based on a common syntax, however, they may have different formal models with differing interpretations.

We consider the scenario that both human agents commit to a specific ontology that deals with a specific domain, *e.g.* animals. The chance that they both refer to the same thing in the world increases considerably. The same holds for the software agents SA_1 and SA_2: They have actual knowledge and they use the ontology to have a common semantic basis. When agent SA_1 uses the term "jaguar", the other agent SA_2 may use the ontology just mentioned as background knowledge and rule out incorrect references, *e.g.* ones that let "jaguar" stand for the car. Human and software agents use their concepts and their inference processes, respectively, in order to narrow down the choice of referents (*e.g.*, because animals do not have wheels, but cars have).

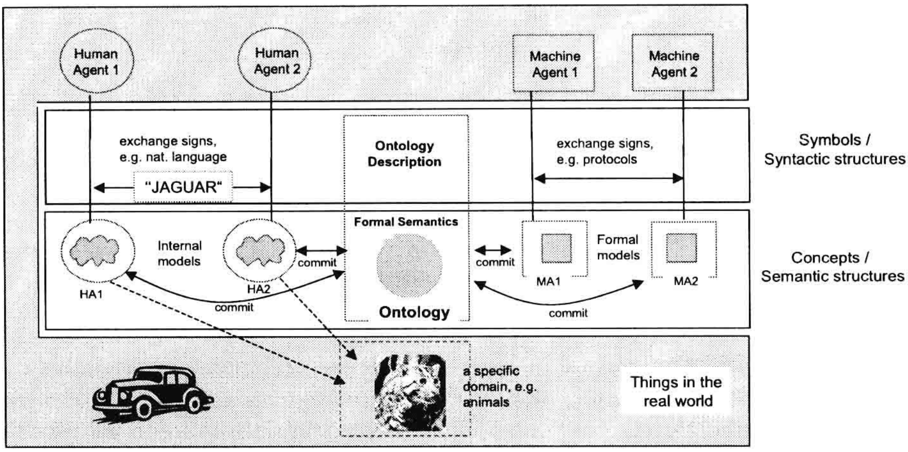

Fig. 2. Communication between human and/or software agents

A new model for ontologies. Subsequently, we define our notion of ontology. However, in contrast to most other research about ontology languages it is not our purpose to invent a new logic language or to redescribe an old one. Rather what we specify is a way of *modeling* an ontology that inherently considers the special role of signs (mostly strings in current ontology-based systems) and references.

Our motivation is based on the conflict that ontologies are for human and software agents, but logical theories are mostly for mathematicians and inference engines. Formal semantics for ontologies is a *sine qua non*. In fact, we build our applications on a well-understood logical framework, *viz.* F-Logic [10]. However, in addition to the benefits of logical rigor, user and developer of an ontology-based system profit from ontology structures that allow to elucidate possible misunderstandings.

For instance, one might specify that the sign "jaguar" refers to the union of the set of all animals that are jaguars and the set of all cars that are jaguars. Alternatively, one may describe that "jaguar" is a sign that may either refer to a concept "animal-jaguar" or to a concept "car-jaguar". We prefer the second way. In conjunction with appropriate GUI modules (*cf.* Sections 3ff) one may avoid presentations of 'funny symbols' to the user like "animal-jaguar", while avoiding 'funny inference' such as may arise from artificial concepts like the union of the sets denoted by 'animal-jaguar' and 'car-jaguar'.

2.2 Ontology *vs.* Knowledge Base

Concerning the general setting just sketched, the term ontology is defined — more or less — as some piece of formal knowledge. However, there are several properties that warrant the distinction of knowledge contained in the ontology *vs.* knowledge contained in the so-called *knowledge base*, which are summarized in Table 1.

Table 1. Distinguishing ontology and knowledge base

	Ontology	Knowledge base
Set of logic statements	yes	yes
Theory	general theory	theory of particular circumstances
Statements are mostly	intensional	extensional
Construction	set up once	continuous change
Description logics	T-Box	A-Box

The ontology constitutes a general logical theory, while the knowledge base describes particular circumstances. In the ontology one tries to capture the general conceptual structures of a domain of interest, while in the knowledge base one aims at the specification of the given state of affairs. Thus, the ontology is (mostly) constituted by *intensional* logical definitions, while the knowledge base comprises (mostly) the *extensional* parts. The theory in the ontology is one which is mostly developed during the set up (and maintenance) of an ontology-based system, while the facts in the knowledge base may be constantly changing. In description logics, the ontology part is mostly described in the T-Box and the knowledge base in the A-Box. However, our current experience is that it is not always possible to distinguish the ontology from the knowledge base by the logical statements that are made. In the conclusion we will briefly mention some of the problems referring to some examples of following sections.

The distinctions ("general" *vs.* "specific", "intensional" *vs.* "extensional", "set up once" *vs.* "continuous change") indicate that for purposes of development, maintenance, and good design of the software system it is reasonable to distinguish between ontology and knowledge base. Also, they describe a rough shape of where to put which parts of a logical theory constraining the intended semantic models that facilitate the referencing task for human and software agents. However, the reader should note that none of these distinctions draw a clear cut borderline between ontology and knowledge base in general. Rather, it is typical that in a few percent of cases it depends on the domain, the view of the modeler, and the experience of the modeler, whether she decides to put particular entitities and relations into the ontology or into the knowledge base.

Both following definitions of ontology and knowledge base specify constraints on the way an ontology (or a knowledge base) should be modeled *in a particular logical language* like F-Logic or OIL:

Definition 1 (Ontology). *An ontology is a sign system* $\mathcal{O} := (\mathcal{L}, \mathcal{F}, \mathcal{G}, \mathcal{C}, \mathcal{H}, \mathcal{R}, \mathcal{A})$, *which consists of*

- *A* ***lexicon****: The lexicon contains a set of signs (lexical entries) for concepts,* $\mathcal{L}^c$, *and a set of signs for relations,* $\mathcal{L}^r$. *Their union is the lexicon* $\mathcal{L} := \mathcal{L}^c \cup \mathcal{L}^r$.
- *Two* ***reference functions*** $\mathcal{F}$, $\mathcal{G}$, *with* $\mathcal{F} : 2^{\mathcal{L}^c} \mapsto 2^{\mathcal{C}}$ *and* $\mathcal{G} : 2^{\mathcal{L}^s} \mapsto 2^{\mathcal{S}}$. $\mathcal{F}$ *and* $\mathcal{G}$ *link sets of lexical entries* $\{L_i\} \subset \mathcal{L}$ *to the set of concepts and relations they refer to, respectively, in the given ontology. In general, one lexical entry*

may refer to several concepts or relations and one concept or relation may be refered to by several lexical entries. Their inverses are $\mathcal{F}^{-1}$ *and* $\mathcal{G}^{-1}$*.*
In order to map easily back and forth and because there is a n to m mapping between lexicon and concepts/relations, $\mathcal{F}$ *and* $\mathcal{G}$ *are defined on sets rather than on single objects.*
- *A set* $\mathcal{C}$ *of* ***concepts****: About each* $C \in \mathcal{C}$ *exists at least one statement in the ontology, viz. its embedding in the taxonomy.*
- *A* ***taxonomy*** $\mathcal{H}$*: Concepts are taxonomically related by the irreflexive, acyclic, transitive relation* $\mathcal{H}$*,* $(\mathcal{H} \subset \mathcal{C} \times \mathcal{C})$*.* $\mathcal{H}(C_1, C_2)$ *means that* C_1 *is a subconcept of* C_2*.*
- *A set of binary* ***relations*** $\mathcal{R}$*:* $\mathcal{R}$ *denotes a set of binary relations.*[2] *They specify pairs of domain and ranges* (D, R) *with* $D, R \in \mathcal{C}$*.*
  The functions d and r applied to a binary relation Q yield the corresponding domain and range concepts D and R, respectively.
- *A set of ontology axioms,* $\mathcal{A}$*.*

The reader may note that the structure we propose is very similar to the WordNet model described by Miller [14]. WordNet has been conceived as a mixed linguistic / psychological model about how people associate words with their meaning. Like WordNet, we allow that one word may have several meanings and one concept (synset) may be represented by several words. However, we allow for a seamless integration into logical languages like OIL or F-Logic by providing very simple means for definition of relations and for knowledge bases.

We define a knowledge base as a collection of object descriptions that refer to a given ontology.

Definition 2 (Knowledge Base). *We define a knowledge base as a 7-tupel* $\mathcal{KB} := (\mathcal{L}, \mathcal{J}, \mathcal{I}, \mathcal{W}, \mathcal{S}, \mathcal{A}, \mathcal{O})$*, that consists of*

- *a* ***lexicon*** *containing a set of signs for instances,* $\mathcal{L}$*.*
- *A* ***reference function*** $\mathcal{J}$ *with* $\mathcal{J} : 2^{\mathcal{L}} \mapsto 2^{\mathcal{I}}$*.* $\mathcal{J}$ *links sets of lexical entries* $\{L_i\} \subset \mathcal{L}$ *to the set of instances they correspond to.*
  Thereby, names may be multiply used, e.g. "Athens" may be used for "Athens, Georgia" or for "Athens, Greece".
- *a set of* ***instances*** $\mathcal{I}$*. About each* $I_k \in \mathcal{I}, k = 1, \ldots, l$ *exists at least one statement in the knowledge base, viz. a membership to a concept C from the ontology* $\mathcal{O}$*.*
- *A* ***membership function*** $\mathcal{W}$ *with* $\mathcal{W} : 2^{\mathcal{I}} \mapsto 2^{\mathcal{C}}$*.* $\mathcal{W}$ *assigns sets of instances to the sets of concepts they are members of.*
- ***Instantiated relations****,* $\mathcal{S}$*, are described, viz.* $\mathcal{S} \subseteq \{(x, y, z) | x \in \mathcal{I},\ y \in \mathcal{R},\ z \in \mathcal{I}\}$*.*
- *A set of knowledge base axioms,* $\mathcal{A}$*.*
- *A reference to an ontology* $\mathcal{O}$*.*

Overall the decision to model some relevant part of the domain in the ontology *vs.* in the knowledge base is often based on gradual distinctions and driven by the needs of the application. Concerning the technical issue it is sometimes

[2] Here at the conceptual level, we do not distinguish between relations and attributes.

even useful to let the lexicon of knowledge base and ontology overlap, *e.g.* to use a concept name to refer to a particular instance in a particular context. In fact researchers in natural language have tackled the question how the reference function J can be dynamically extended given an ontology, a context, a knowledge base and a particular sentence.

3 SEAL Infrastructure and Core Modules

The aim of our intranet application is the presentation of information to human and software agents taking advantage of semantic structures. In this section, we first elaborate on the general architecture for SEAL (SEmantic PortAL), before we explain functionalities of its core modules.

3.1 Architecture

The overall architecture and environment of SEAL is depicted in Figure 3:

The *backbone* of the system consists of the *knowledge warehouse*, *i.e.* the data repository, and the *Ontobroker* system, *i.e.* the principal inferencing mechanism. The latter functions as a kind of middleware run-time system, possibly mediating between different information sources when the environment becomes more complex than it is now.

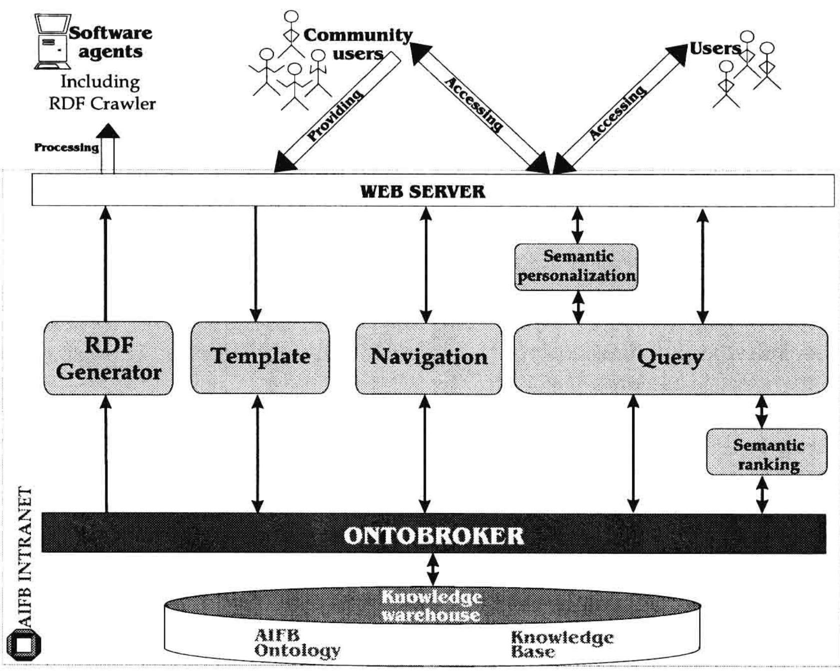

Fig. 3. AIFB Intranet - System architecture

At the front end one may distinguish between three types of *agents*: *software agents*, *community users* and *general users*. All three of them communicate with the system through the *web server*. The three different types of agents correspond to three primary modes of interaction with the system.

First, remote applications (*e.g.* software agents) may process information stored at the portal over the internet. For this purpose, the *RDF generator* presents RDF facts through the web server. Software agents with *RDF crawlers* may collect the facts and, thus, have direct access to semantic knowledge stored at the web site.

Second, community users and general users can access information contained at the web site. Two forms of accessing are supported: navigating through the portal by exploiting hyperlink structure of documents and searching for information by posting queries. The hyperlink structure is partially given by the portal builder, but it may be extended with the help of the *navigation* module. The navigation module exploits inferencing capabilities of the inference engine in order to construct conceptual hyperlink structures. Searching and querying is performed via the *query* module. In addition, the user can personalise the search interface using the *semantic personalization* preprocessing module and/or rank retrieved results according to semantic similarity (done by the postprocessing module for *semantic ranking*). Queries also take advantage of the Ontobroker inferencing.

Third, only community users can provide data. Typical information they contribute includes personal data, information about research areas, publications, activities and other research information. For each type of information they contribute there is (at least) one concept in the ontology. Retrieving parts of the ontology, the *template* module may semi-automatically produce suitable HTML forms for data input. The community users fill in these forms and the template modules stores the data in the knowledge warehouse.

3.2 Core Modules

The core modules have been extensively described in [19]. In order to give the reader a compact overview we here shortly survey their function. In the remainder of the paper we delve deeper into those aspects that have been added or considerably extended recently, *viz.* semantic ranking (Section 4), and semantic access by software agents (Section 5).

Ontobroker. The Ontobroker system [6] is a deductive, object-oriented database system operating either in main memory or on a relational database (via JDBC). It provides compilers for different languages to describe ontologies, rules and facts. Beside other usage, in this architecture it is also used as an inference engine (server). It reads input files containing the knowledge base and the ontology, evaluates incoming queries, and returns the results derived from the combination of ontology, knowledge base and query.

The possibility to derive additional factual knowledge from given facts and background knowledge considerably facilitates the life of the knowledge providers and the knowledge seekers. For instance, one may specify that if a person belongs