Intelligent Database Systems

Barbara Catania
University of Genova (Italy)

IIWAS 2001 - Linz, Austria
Outline

• Introduction to Intelligent Database Systems (IDBs)
• Fundamental IDB approaches
• IDBs and their role in Web applications
• An IDB approach for metadata representation and retrieval
• Conclusions
• Bibliography
Introduction
What is an IDB?

**DB technology:**
- limited modeling capabilities
- new data management applications

**AI techniques:**
- often toy systems
- no persistent management of data

Late '80s/’90s

DB techniques can aid an AI system to deal with large amount of information

AI techniques can provide semantic support to a DB system

IDB Technology
Characteristics of IDBs

- Architecture based (at least implicitly) on an organization in the Expert Systems (ESs) style
  - Fact DataBase (FDB) + Rule Base (RLB)
- Use of AI techniques
  - Knowledge representation techniques
    - semantic data representation
  - Inference techniques
    - improved reasoning about data
  - Intelligent user interfaces
    - help users to make requests and receive replies
- persistency of the FDB
A traditional taxonomy of IDBs

Efforts originated in a DB context

Static extensions
extending the expressive power of traditional DB data models

Dynamic extensions
introducing some form of reasoning inside DBMSs

Efforts originated in a AI context

Basic solutions
coupling knowledge-based systems and DBMSs

Advanced solutions
try to use AI systems to deal directly with large amount of information
Fundamental IDB approaches
Efforts originating in a DB context

Dynamic extensions

Relational model

75  80  85  90  95  00

Static extensions

Semantic models

Hyper-semantic models

Nested model

Hyper-semantic models

OODBMS Active DBMS

UML

OMT

Barbara Catania
Static extensions

Extensions of the relational model

- Nested relational model

Conceptual models

- Semantic data models
  - Hyper-semantic data models
  - OO data models

Other data models

- Relational
- Functional
- Semantic networks

KL-ONE - AI
Dynamic extensions

Introducing programming languages constructs

Introducing active rules

OODBMS

Active DBMS

Frame-based systems - AI

Production rules - AI

semantic data models
Efforts originating in an AI context

- Based on the notion of Knowledge Based system (KBS)
- KBSs typically contain:
  - explicitly represented rules RLB
  - simple facts FDB
  - components which can make inferences over the Knowledge Base KB = RLB + FDB
- the information dealt with by the KBS consists therefore of:
  - explicitly stored facts and rules
  - derived facts
Knowledge Based System types

- Pure rule-based representations supporting inference by resolution
  - systems developed in a logic programming context
  - ES shells based only on the production rule paradigm
- Pure frame- or object-based representations supporting inference by inheritance
  - frame systems
  - terminological (description logic) systems
  - KESE: hybrid systems, commercially available, supporting alternative inference methods and representation schemes (SPOKE, KEE)
### An overall view

<table>
<thead>
<tr>
<th>KBS type</th>
<th>Formalism</th>
<th>Reasoning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logic programming systems</td>
<td>logical clauses</td>
<td>resolution principle</td>
</tr>
<tr>
<td>Expert Systems (ES)</td>
<td>production rules</td>
<td>procedural, logic</td>
</tr>
<tr>
<td>Inheritance-based systems</td>
<td>ontology of concepts, inheritance hierarchies</td>
<td>inheritance</td>
</tr>
<tr>
<td>Frame-based systems</td>
<td>structured concepts</td>
<td>inheritance</td>
</tr>
<tr>
<td>Terminological systems</td>
<td>terminological knowledge, assertional knowledge, based on description logic</td>
<td>classification</td>
</tr>
<tr>
<td>KESE</td>
<td>various approaches</td>
<td>various approaches</td>
</tr>
</tbody>
</table>
Tradeoff

- Tradeoff between: computational complexity, expressive power, and completeness
  - sound, complete, tractable but limited expressive power
    - KRYPTION, CLASSIC
  - sound, complete, intractable
    - KRISL
  - sound, higher expressive power, intractable and incomplete (more efficient)
    - BACK, NIKL, LOOM
KBS and DBMS

• Conventional KBSs are inadequate for supporting new data/knowledge-intensive applications

• Problems:
  – KBSs usually deal with knowledge bases of small size, in volatile memory
  – KBSs provide only limited DBMS services

• Possible solution: coupling KBSs with DBMSs
  – Coupling of logic programming systems with DBMSs
    • Deductive databases
  – coupling of ES shells and KESEs with DBMSs
    • five classes of approaches
Deductive databases

- Intensional database (IDB), containing logic formulas
  - IDB: parent(X,Y) <- father(X,Y)
  - IDB: parent(X,Y) <- father(X,Z), parent(Z,Y)

- Extensional database (EDB), containing base relations
  - EDB: father(ann,john)
  - EDB: father(john, mark)

- Through logic inference mechanisms, derive, from base relations, information not explicitly stored in the EDB
  - father(ann,john), father(john, mark),
  - parent(ann,john), parent(john, mark), parent(ann,mark)

- Language typically used for IDB:
  - Datalog (restriction of Prolog, set-oriented)

- formal theoretical foundation
Homogeneous approach: a pure logic system

Homogeneous approach: an enhanced system

Heterogeneous approach
Coupling ES shells and KESEs with DBMSs

• No theoretical foundations
• mismatch between ES shell/KESEs and DBMS
  • semantic, impedance, and granularity mismatch
• most proposals for KESEs does not give rise to real IDBs
  – useful for historical motivations
  – they represent the basic approaches of IDB architecture
• DB used to store AI objects:
  – AI objects are translated into and out of DB objects
  – AI objects are stored in their native format in the DB (for example, as LOB)
Coupling ES shells and KESEs with DBMSs

- Full-bridge solution
- Often flat file as intermediate medium
- The control of the interactions and the processing can be located on the central bridge or distributed
- Such architecture does not scale up well
- Examples:
  - DIFEAD (ESs, rel. DBMSs, intermediate data dictionary)
  - KADBASE (ESs, rel. DBMSs, distributed)
  - Europe-Bruke approach (BACK, rel. DBMSs)
Coupling ES shells and KESEs with DBMSs

- Extension of a KB with components proper to a DBMS
- used mainly for KBs based on the logical approach
- adopted by the vendors of the main ES tools to provide their systems with some elementary possibilities of extracting information from a database
- Examples: ROCK, KBMS, SPOKE
- no standard approach exists for realizing the access functions
Coupling ES shells and KESEs with DBMSs

- Extension of a DBMS with components proper to a KBS
- Two possible interactions:
  - explicit access procedure: an explicit call to the KBS is inserted in the application program
  - implicit access procedure: the access to the inference engine is through the same query interface used to access data
- Similar to rule based systems and OODBMSs
Coupling ES shells and KESEs with DBMSs

- DB and KBS systems are strongly integrated
  - only one environment
  - no semantic mismatch problems
- Architecture d):
  - construction of a DB system after (or during) the set up of the KBS
  - integral approaches
- Architecture e):
  - the DBMS technology is more stable and mature than the KBS technology, and the installed base of DBs is definitely larger than the KBSs base
  - DBs are probably a better place for incorporating ES functionalities than vice versa
  - Examples: ARCHES, KBase
Integral approach

• Example of type d) architecture:
  – only one pure KBS environment
  – data model is some sort of AI knowledge representation language
  – all sort of inference techniques are used
  – KBS environment should be able to support DBMS services

• attempt to use some kind of AI system to deal directly, in a DBMS style, with large quantities of persistent information

• no theoretical foundation

• Example: TELOS, CYC, NKRL, lexical approaches (WordNet)

• Limitations:
  – great variety of knowledge representation models
  – complexity of the used formalisms
  – lack in supporting DBMS functionalities
IDBs and their role in Web applications
Some applications

• Metadata representation

• Integration of heterogeneous sources

• Web application design
Metadata representation: problem

unstructured heterogeneous textual or multimedia documents over the Web

Metadata generation

1

Query mechanisms

2

3
Metadata representation: an IDB approach

- Unstructured (possibly multimedia) document
  - General solution for the mixed media access problem
    - texts
    - images
    - pictures
    - ...
  - support similarity-based indexing
    - similar caption = similar documents
Metadata representation: examples

• Solutions based on the illustrated approach have been proposed, among the others, in:
  – CYC
  – NKRL (see later)

• a solution based on TELOS has also been proposed to construct and manage an API for a metadata repository
Integration: problem

• Problems:
  – how is it possible to represent a global domain model?
  – how is it possible to represent the local knowledge?
  – how is it possible to map global queries into local queries and merging results?
Integration: an IDB approach

- Usage of knowledge representation languages for representing:
  - domain model
  - heterogeneous sources
  - query mapping
- important role played by ontologies
- advantages:
  - clear formal and declarative foundation
  - powerful reasoning facilities
Integration: examples

• Carnot project (MCC):
  – integration of heterogeneous sources using a set of articulation axioms that describe how to map SQL queries and domain concepts
  – articulation axioms built in CYC

• SIMS (University of Southern California):
  – LOOM is used both to represent the global domain model and the local heterogeneous sources characteristics

• TSIMMIS (Stanford University):
  – inheritance-based language (OEM) to describe sources
  – a logic OO-language is used to specify mediators as views upon OEM sources (LOREL)

• Garlic (IBM)
  – ODMG as model for sources and programming interfaces
Web application design: problem

- Web applications are characterized by three main design dimensions:
  - structure
  - navigation
  - presentation

- Problems:
  - which models can be used to support the development of Web applications in all the lifecycle steps?
Web application design: approach

• Conceptual level:
  – Structural modeling:
    • semantic/hypersemantc data models
    • OO models
  – Navigation:
    • techniques proposed for the more general problem of human-computer interaction specification
      – first-order logic, Petri Nets, finite state machines, ...
  – Presentation:
    • software tools and formal methods

• Design level:
  – structured or semi-structured data models
Web application design: examples

- WebML [Politecnico of Milano, Italy]
- Araneus [University of Rome, Italy]
- Strudel [At&T, INRIA, Univ. Washington]
An IDB approach for metadata representation and retrieval

Joint work with E. Bertino and G.P. Zarri
To better explain ... 

• A specific problem concerning Web applications
• a concrete approach
  – example of an integral approach (NKRL)
  – example of KESE (CONCERTO)
  – example of type b) architecture (Knowledge Manager)
  – important problems (standardization, DBMS facilities)
Metadata

- Machine-understandable knowledge that describes the properties and the relationships of Internet resources
- To be used to get information about the structure and the contents of these resources
- Different classes:
  - Structure-based metadata: external characteristics of the support (color, shape, texture, motion, etc.)
  - Content-specific metadata: representing the meaning of documents
    - keywords
    - conceptual annotations
Conceptual annotations

- Structured information, describing in depth the semantic meaning of a document
- several proposals:
  - UNTANGLE, MIHMA, Information Manifold, Ontobroker
- often based on description logic
- limitations:
  - often unable to describe complex events
  - not always adequate to describe actions, facts, events
  - automatic extraction quite difficult
- alternative approach: NKRL [Zarri, ‘94-’00]
Why NKRL ?

• NKRL: Narrative Knowledge Representation Language
• Ability to represent, through ontologies, both:
  – the important notions of a given application domains (concepts)
  – mutual relationships between concepts (facts, events)
• ability to (partially) automatically extract conceptual annotations in NKRL by using tools developed in two European projects:
  – NOMOS (Esprit P5330)
  – COBALT (LRE P61011)
• the proposed solution fluctuates between:
  – very simple, low-level rule-based techniques making use of elementary semantic categories like those included in WordNet
  – very complex inference-intensive applications of CYC
The proposed approach

Unstructured (possibly multimedia) document

- General solution for the mixed media access problem
  - texts
  - images
  - pictures
  - ...
- support similarity-based indexing
  - similar caption = similar documents

Natural Language (NL) caption

Annotation in NKRL
On March 1st, 2000, Barbara will go to London.
Definitional component

- Supplies the tools for representing the important notions (concepts) of a given domain
- A concept is a frame-based structure composed of
  - OID
  - Symbolic label like physical_entity, human_being, city, etc.
  - A set of characteristics features
- Concepts are represented by using an ontology of terms, called HCLASS
- General concepts belonging to the upper levels of are represented inside a catalogue and are assumed to be invariable
- Similarities with terminological languages
Enumerative component

- It is composed of all the instances of sortal concepts, called individuals
- non sortal concepts does not admit direct instances
- similarly to concepts, individuals are represented as frame based structures
- Example:
  - chair27
  - paris_
  - lucy_
Descriptive component

- It contains the description of the events proper to a given domain
- supplies the tools used to produce the formal representations (predicative templates) of general classes of narrative events, like ‘moving a generic object’, ‘formulate a need’, ‘be present somewhere’
- Templates are structured into an inheritance hierarchy, HTEMP, corresponding to a taxonomy of events
- Basic templates (more than 150) are described in a catalogue
- By means of proper specialization operations, it is possible to obtain from the basic templates the derived templates needed to implement a particular application
Descriptive component

• Templates are characterized by a threefold format:
  \((P_i (R_1 a_1 ) (R_2 a_2 ) \ldots (R_n a_n ))\)
  
  – \(P_i\) denotes the symbolic label identifying the template (class of events)
  
  – \(R_k\), \(k = 1, \ldots, n\), denote generic roles
  
  – \(a_k\), \(k = 1, \ldots, n\), denote the arguments associated with the roles (concepts, instances, pred. occ.)

• Predicates: BEHAVE, EXIST, EXPERIENCE, MOVE, OWN, PRODUCE, RECEIVE

• Roles: SUBJ(ect), OBJ(ect), SOURCE, DEST(ination), MODAL(ity), TOPIC, CONTEXT
Factual component

• Concerns the instances (predicative occurrences) of the predicative templates
  – representation of single, specific events

• Examples:
  – Tomorrow, I will move the wardrobe
  – Lucy was looking for a taxi
Example

Milan, October 15, 1993. The financial daily Il Sole 24 Ore reported Mediobanca had called a special board meeting concerning plans for capital increase.

c1) MOVE SUBJ (SPECIF sole_24_ore financial_daily): (milan_)
   OBJ #c2
date-1: 15_october_93
date-2:

c2) PRODUCE SUBJ mediobanca_
   OBJ (SPECIF summoning_
       (SPECIF board_meeting_1 mediobanca_ special_))
   TOPIC (SPECIF plan_1 (SPECIF cardinality_several_)
          capital_increase_1)
date-1: circa_15_october_93
date-2:
Advanced representation facilities

- Structured arguments built up making use of a specialized sublanguage (AECS), including four expansion operators:
  - disjunctive (ALTERNative = A)
  - distributive (ENUMeration = E)
  - collective (COORDination = C)
  - attributive (SPECIFication = S)
- ability to bind predicative occurrences together
  - binding occurrences
Application to multimedia documents

Three nice girls are lying on the beach

C1) EXIST SUBJ (SPECIF girl_1 nice_ (SPECIF cardinality_ 3)): (beach_1)
MODAL lying_position

[girl_1
  InstanceOf: girl_
  HasMember: 3]
Queries in NKRL

- Query are expressed through search patterns
- It must be possible to specify:
  - perfect match (identical structure)
  - perfect match apart from cardinality (identical structure apart from the cardinality of AECS lists)
  - subsumed match (information globally congruent from a semantic point of view - e.g., additional SPEFIC lists are possible-)
- automatic transformation of queries into similar queries
Example

Which was the theme of the recent board meeting called out by Mediobanca?

((?w IS-OCCURRENCE :predicate PRODUCE :SUBJ mediobanca_ :OBJ (SPECIF ?x (SPECIF ?y mediobanca_)) :TOPIC ?z) (1_october_93, 20_october_93) ((?x IS-A (:OR assembly_ adjournment_ dissolution_)) (?y IS-A board_meeting) (?z IS-A planning_activity)))
The CONCERTO Esprit Project

• The previous ideas have been implemented in the context of the CONCERTO Esprit Project
• only textual, possibly semi-structured (HTML, XML) documents
• the architecture can be extended to deal with multimedia documents
The CONCERTO KESE architecture

Communication through XML

Acquisizione e pre-processing

XML Document Translator

BSE Extraction

Conceptual Annotation Builder

Knowledge Management

Template Manager

Concept Manager

Knowledge Manager

Inference Engine

Document Repository

Concept Repository

Template Repository

Conceptual Annotation Repository

Document Repository

Implementation in JAVA

Interfaces

Document Acquisition Interface

Conceptual Annotation Builder Interf.

Query Environment Interface

XML format

Barbara Catania

Pag. 53

IIWAS 2001 - Linz, Austria
The KM architecture

Example of type b) architecture

TCP/IP protocol

KMIL document through HTTP protocol

CLIENT

Knowledge Manager

Web server

Servlet engine

Java API

Repository

Ontologies

Conceptual annotations

Documents

n

1

n
Technological choices

• How to represent conceptual annotations

• How to implement the repositories

• How to communicate with the Knowledge Manager
Ontologies and Conc. Ann. Representation

- Ontologies:
  - linearization of the hierarchies in a set of tables

- Conceptual annotations:
  - Traditional implementation: three-layered approach:
    - Common Lisp + a frame/object oriented environment + NKRL
  - To increase the standardization:
    - Java + RDF (Resource Description Format)
    - implemented in XML
RDF

- RDF (W3C): proposal for defining and processing WWW metadata
- model based on directed labelled graphs
  - nodes represent Web resources
  - described by using attributes
  - edges represent relationships between resources
- no predefined vocabulary (ontologies, keywords,...) exists
- model implemented in XML
Problems mapping NKRL in RDF

- **RDF structures**: dyadic
  - two resources are linked by a binary conceptual relation under the form of a property
- **NKRL structures**: threefold relationship
  - symbolic label
  - predicate
  - one or more roles and fillers
- **NKRL structures have been transformed in dyadic structures and mapped in RDF**
An example of RDF representation

```xml
<?xml version=1.0 ?>
<!DOCTYPE DOCUMENTS SYSTEM CA_RDF.dtd>
<CONCEPTUAL_ANNOTATION>
  <rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
           xmlns:ca=http://projects.pira.co.uk/concerto#>
    <rdf:Description about=occ11824>
      <rdf:type resource=ca:Occurrence/>
      <ca:instanceOf>Template43</ca:instanceOf>
      <ca:predicateName>Move</ca:predicateName>
      <ca:subject rdf:ID=Subj43 rdf:parseType=Resource>
        <ca:filler>barbara_</ca:filler>
      </ca:subject>
      <ca:object rdf:ID=Obj43 rdf:parseType=Resource>
        <ca:filler>london</ca:filler>
      </ca:object>
      <ca:listOfModulators>
        <rdf:Seq><rdf:li>begin</rdf:li></rdf:Seq>
      </ca:listOfModulators>
      <ca:date1>01/03/2000</ca:date1>
    </rdf:Description>
  </rdf:RDF>
</CONCEPTUAL_ANNOTATION>
```
Repository implementation

• Conceptual annotations are represented in XML
• Two possible usages of XML documents:
  – Data Centric: such documents represent the tool by which traditional data are transferred over the Web
    • XML as a vehicle for data transfer
    • Example: sales orders, flights scheduling, ...
  – Document Centric: the information is represented by the document itself
    • XML as a model for data representation
    • Example: books, textual documents, metadata

• In CONCERTO:
  – document centric XML documents
XML and DBMS

- Two categories of DBMS:
  - XML-Native DBMS: architecture designed for totally supporting management of XML documents
    - not yet very robust
    - useful for Document Centric documents
    - Example: eXcelon (Object Design Inc.)
  - XML-Enabled DBMS: all DBMS that extend their architecture with functionalities proper to the management of XML documents
    - Object-Relational (DB2, Oracle8i, …), relational (Microsoft SQL Server)
    - useful for Data Centric and partially for Document Centric documents
Communication protocol

• Standard communication protocol
  – Knowledge Manager Interface Language (KMIL) for interacting with the KM
  – implemented in XML

• The Knowledge Manager can be hosted on a generic machine, becoming independent from the other modules of the architecture
An example of KMIL input

```xml
<?xml version="1.0"?>
<!DOCTYPE KMIL-SESSION SYSTEM KmILIn.dtd>
<KMIL-SESSION>
  <KMIL-ACTION serial_number=1>
    <KMIL-INSERT-PredOcc IdPO=occ11824 Doc=doc132>
      <TEXT> RDF Text </TEXT>
    </KMIL-INSERT-PredOcc>
  </KMIL-ACTION>
  <KMIL-ACTION serial_number=2>
    <KMIL-INSERT-PredOcc IdPO=occ11845 Doc=doc133>
      <TEXT> RDF Text </TEXT>
    </KMIL-INSERT-PredOcc>
  </KMIL-ACTION>
</KMIL-SESSION>
```
An example of KMIL output

<?xml version="1.0"?>
<!DOCTYPE KMIL-SESSION SYSTEM KmILOut.dtd>
<KMIL-SESSION>
  <KMIL-ACTION-OUTPUT serial_number=1
    action_status = OK>
  </KMIL-ACTION>
  <KMIL-ACTION serial_number=2
    action_status = ERROR>
    <ERROR-CODE code = KMIL-ERR-08/>
  </KMIL-ACTION>
</KMIL-SESSION>
On-going work

- Efficient management of conceptual annotations on persistent storage
  - clustering
  - optimization/indexing
  - security
- Strongly related to XML document management
  - initial work on clustering and caching
Conclusions
Ideal IDBs

• Like a DBMS:
  – persistent storage management
  – support of query and update languages
  – support of indexing and query optimization techniques
  – concurrency control and recovery
  – security

• Like advanced data models:
  – like nested models: non-atomic attribute values
  – like semantic data models and OODBMS: abstraction, inheritance
  – like hyper-semantic data models: no real distinction between data and knowledge
  – like active DB: reaction
Ideal IDBs

• Like more advanced KESEs:
  – support of various inference techniques: deductive, abductive, nonmonotonic, probabilistic, analogical

• Like any state-of-the-art DBMS, ES shell or KESE:
  – sophisticated user interfaces as well as knowledge and application engineering tools

• Like (some) advanced systems in the AI style:
  – uniform, high-level type of representation (frames, objects, semantic networks, hybrid representation schemata …) in both the Rule Base and the Fact DataBase
Recent trends

• The techniques analyzed before are now at the basis of several research directions

• macroscopic directions:
  – Advanced data models
  – advanced reasoning
  – advanced architectures
  – advanced index and retrieval techniques
An extended taxonomy of IDBs

- Advanced data models
  - ... 
  - temporal DBMS 
  - semistructured and unstructured data representation 
  - Ontologies

- advanced architectures
  - heterogeneous systems 
  - cooperative systems 

- advanced reasoning
  - ... 
  - temporal DBMS 
  - query languages for semistructured and unstructured data 
  - data mining

- advanced indexing techniques
  - indexing and retrieving advanced data 
  - Internet indexing and retrieval techniques
Bibliography
Efforts in the DB context


Efforts in the DB context


Efforts in the AI context


ECAI. Proc. of the European Conference on Artificial Intelligence.

Efforts in the AI context


Efforts in the AI context


Coupling DB and AI efforts


**Coupling DB and AI efforts**


KRDB WORKSHOPS. Proc. of the Workshop on Knowledge Representation Meets Databases.


Coupling DB and AI efforts


IDBs and Web


IDBs and Web


NKRL for semistructured data representation


