



# OntoDDB

## Ontology Driven Data Base

### Esther Lozano

*Contact: Raimundo Lozano {rlozano@clinic.ub.es}  
Informática Médica, Corporación Sanitaria Clínic,  
Villarroel 170, 08036 Barcelona, Spain*

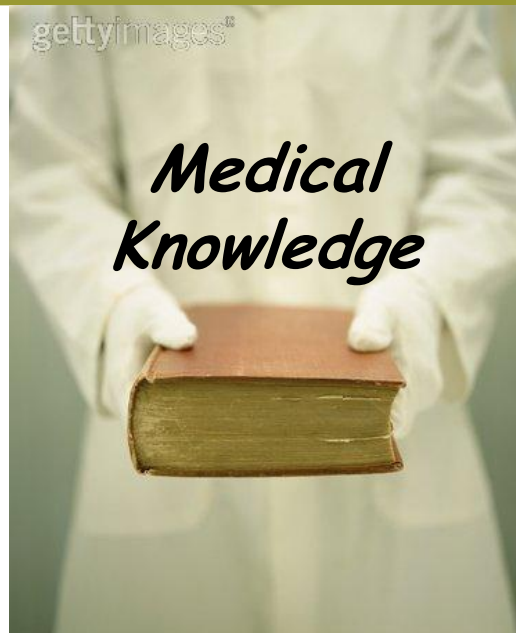


# INTRODUCTION



Patient and  
their problems

Monitoring  
anamnesis,  
exploration



*Medical  
Knowledge*

register

**Data**

transformation

Syndromic  
diagnosis

**Information**

Differential  
diagnosis

**Integration**

combination

inference

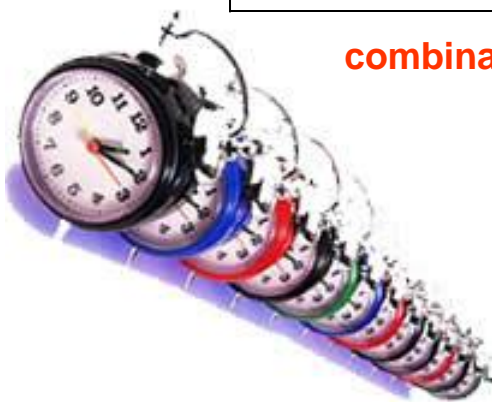
Diagnostic  
hypothesis  
**Decisions  
making**

Medical  
orders

planning

**Actions**

execution



# PREMISES

---

- Information / Knowledge division
  - Making explicit the knowledge
- Explicit knowledge can be managed
- Knowledge management  $\Rightarrow$  Knowledge-driven systems
- More interaction in systems construction
- OntoDDB born as a system for integrating different DBs in RDF and RDF/S models
- Improvement of the system through the years

# USE CASE: DB for research

- Data register and storage
- Reflection of structure of data
- Data extraction for analysis
- Context
  - Distributed access
  - Constant changes in knowledge: i.e. new relevant information to be included in research study

# INCONVENIENTS

- Cost
  - In development: Database, Web Application
  - In maintenance: database modifications, web pages modifications
- Long development time
- Very short amortization time
- Need of very technical resources
- Distance between researcher and developer
- Heterogeneity between different applications
- No reuse

# OBJECTIVES

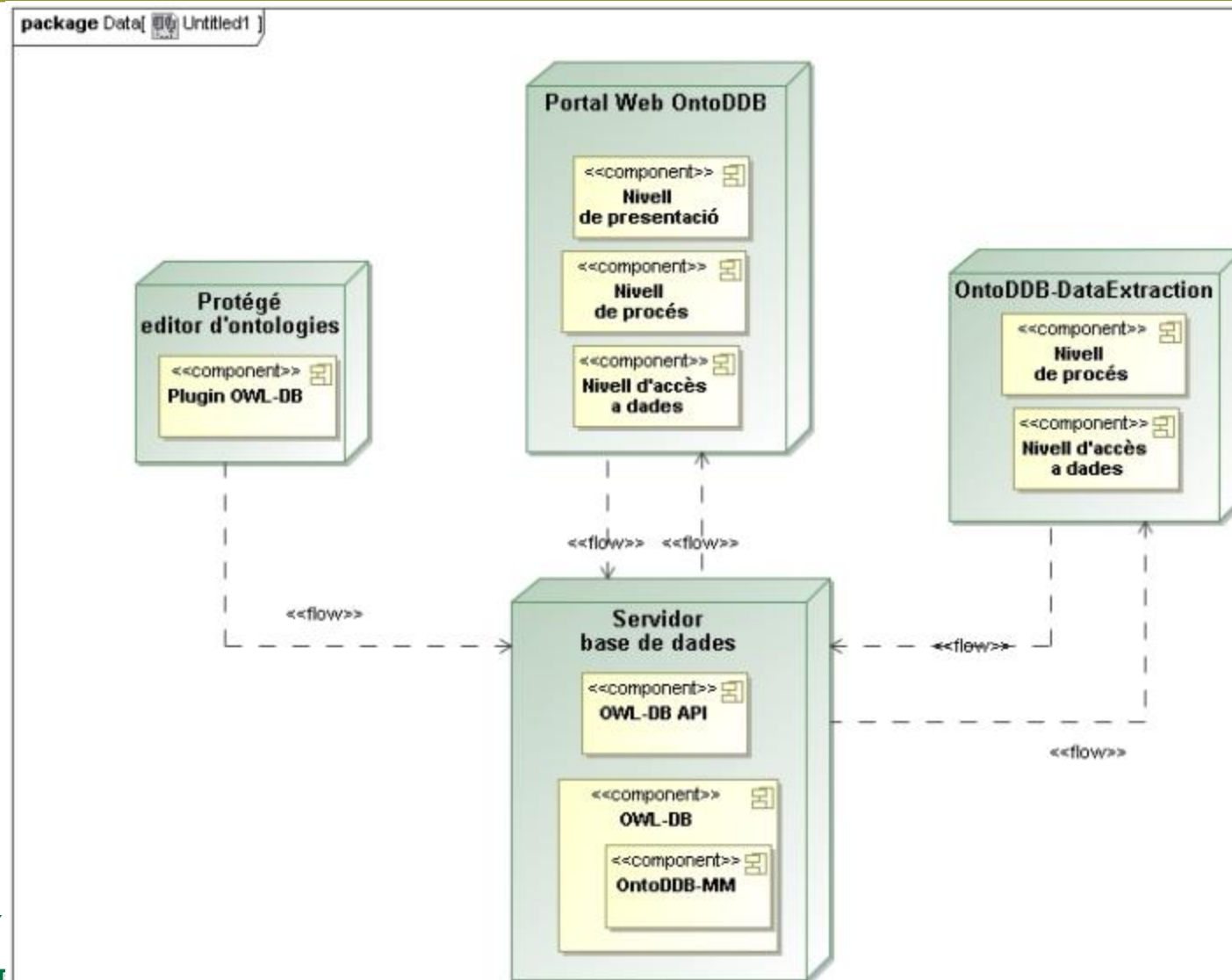
## KNOWLEDGE

- Specification of data model by ontologies
- Specification of user interface by ontologies
- Automatic storage of data and user interface
- On-line modifications
- Data extraction
- Distributed access



## DATA

# COMPONENTS



# OWL STORAGE

- Requirements
  - Wide scope, not limited to any project
  - Conceptual representation, not attached to any format
  - Portable between different DBMS
  - Efficiency retrieving concepts
- No good models proposed
  - Very simple
  - Not efficient
- Solution
  - Design a new storage model taking advantage of relational capabilities
    - Make explicit all OWL components defined in the OWL specification: classes, properties, literals, etc.



# ONTOLOGY REPOSITORIES

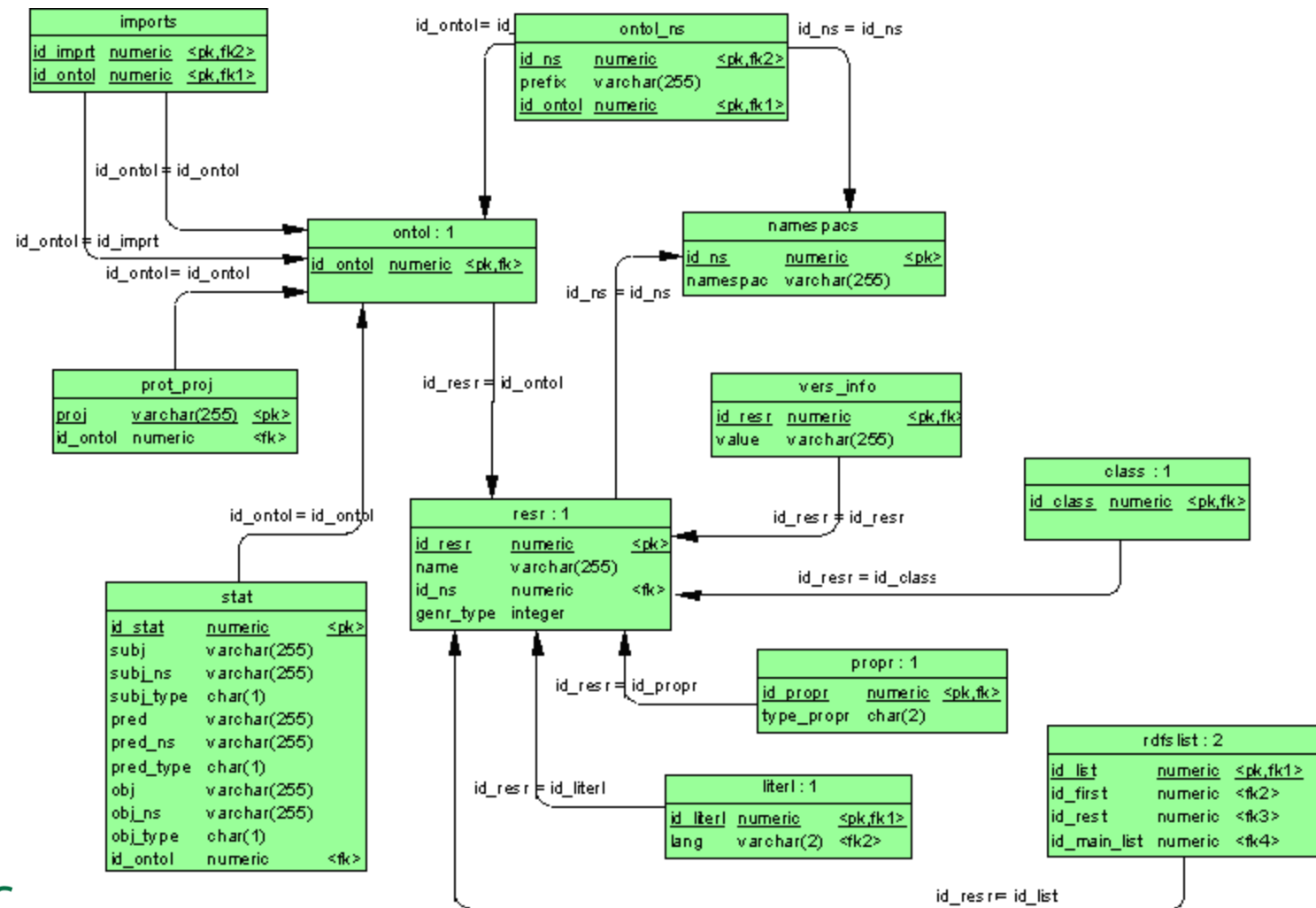
- Three popular database representations:
  - Schema-aware storage
    - Also called *specific* or *binary*
    - One table per RDF/S schema property or class
  - Schema-oblivious storage
    - Also called *generic* or *vertical*
    - One table to store triples (*subject-predicate-object*)
  - Hybrid of schema-aware and schema-oblivious representations
    - A ternary relation for every different property range
    - A binary relation for all classes instances

Theoharis, Y., Christophides, V., Karvounarakis, G. *Benchmarking Database Representations of RDF/S Stores*. In: Gil, Y., Motta, E., Benjamins, V.R., Musen, M.A. (eds.) ISWC 2005. LNCS, vol. 3729, pp. 685–701. Springer, Heidelberg (2005)

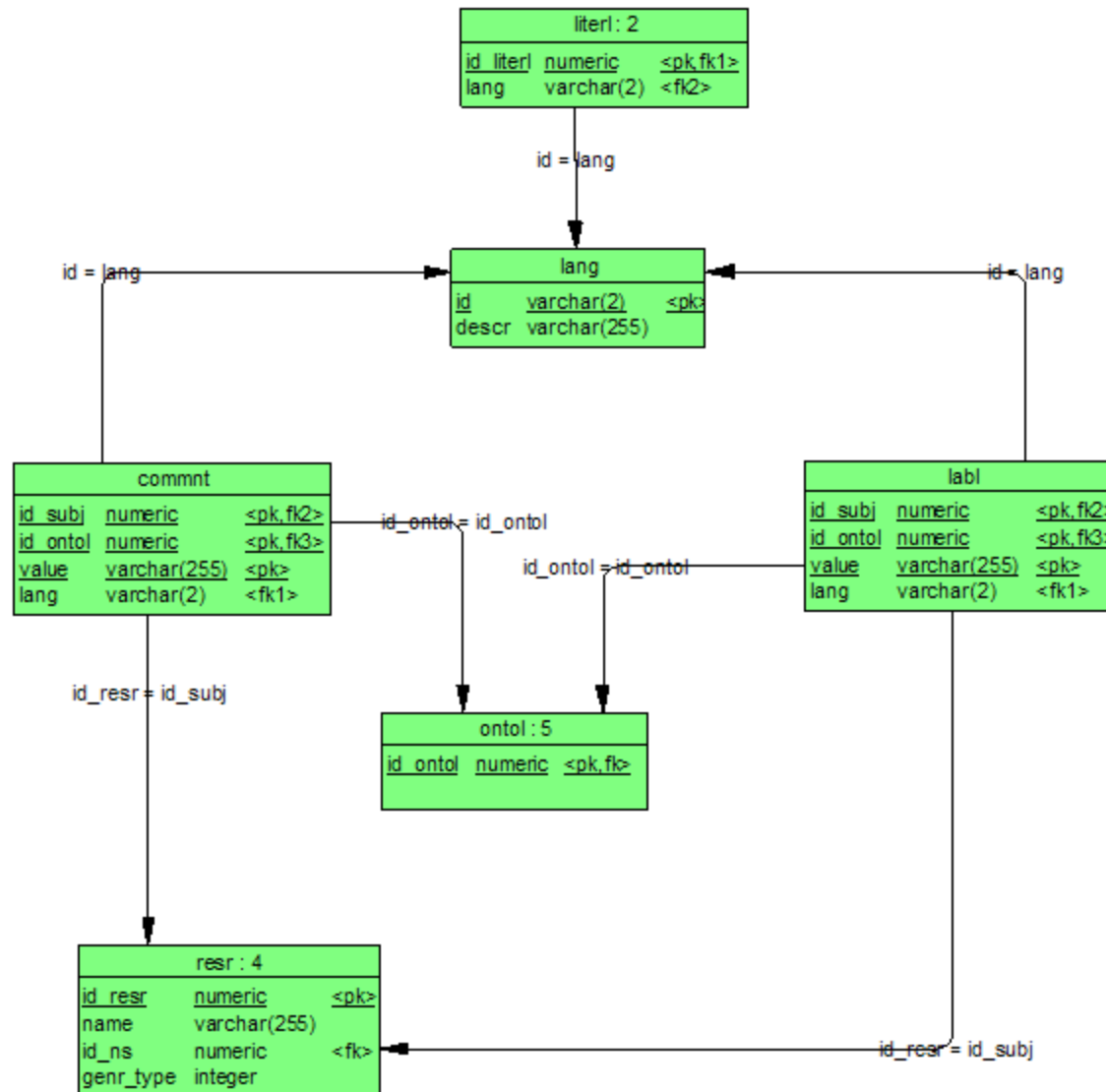
# OWL-DB: Characteristics

- EAV-based database
  - Tables for all OWL elements
  - Resources identified by internal ID
- Statements table
  - Resources identified by URI
  - No order dependent
- Statements as access point to the system
  - Can be used by any application managing OWL statements
- Propagation of information from statements table to rest of tables to ensure consistency
  - Triggers and procedures
- API for external applications

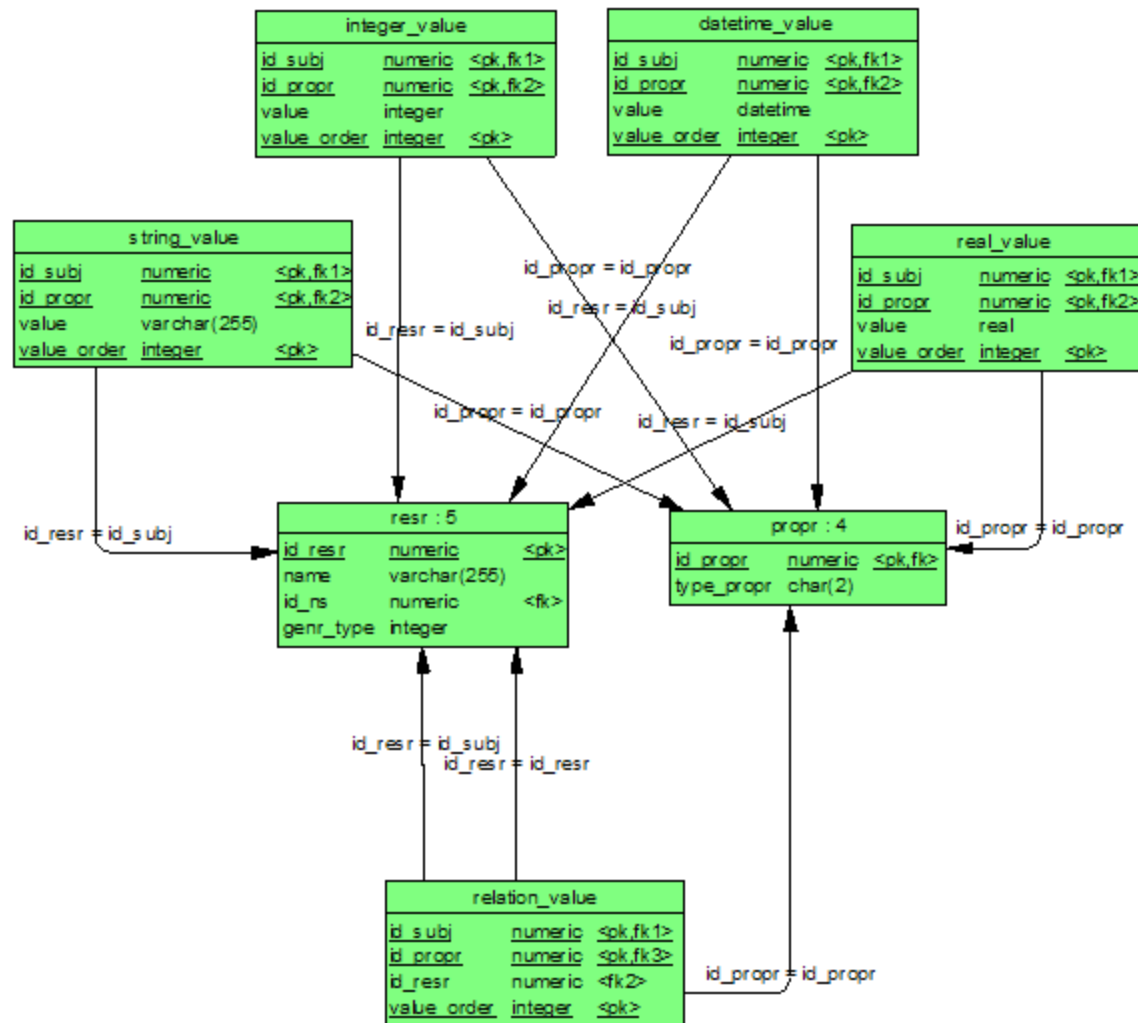
# OWL-DB: Database structure



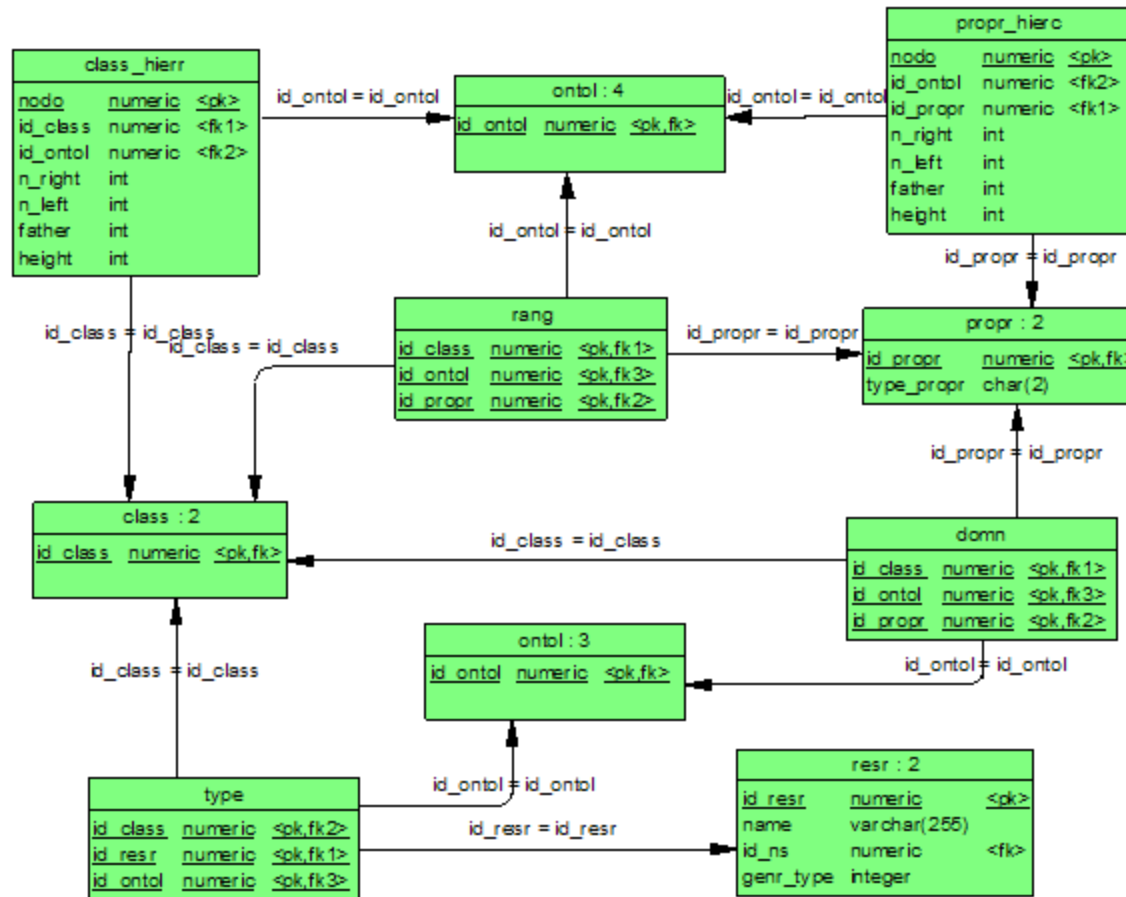
# OWL-DB: Database structure



# OWL-DB: Database structure



# OWL-DB: Database structure

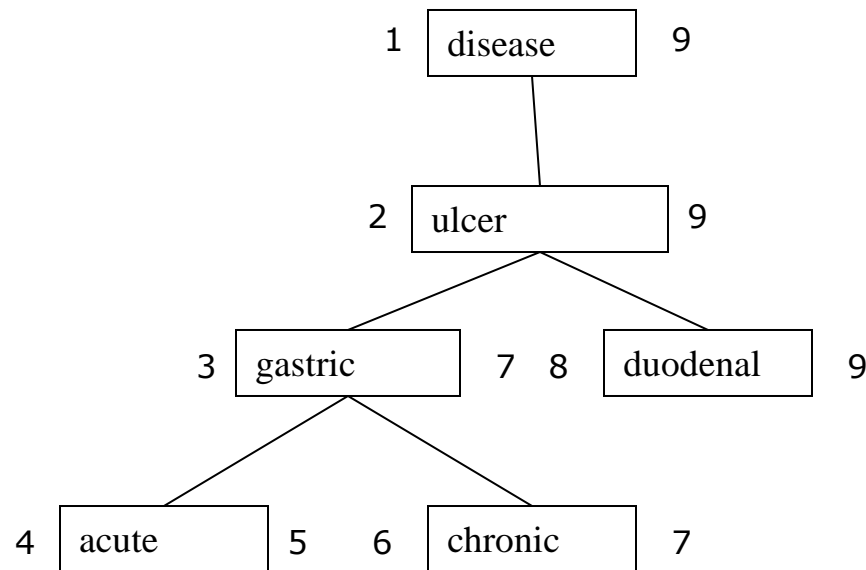


# OWL-DB: Class Hierarchy

- Classes organized in a tree with indexes

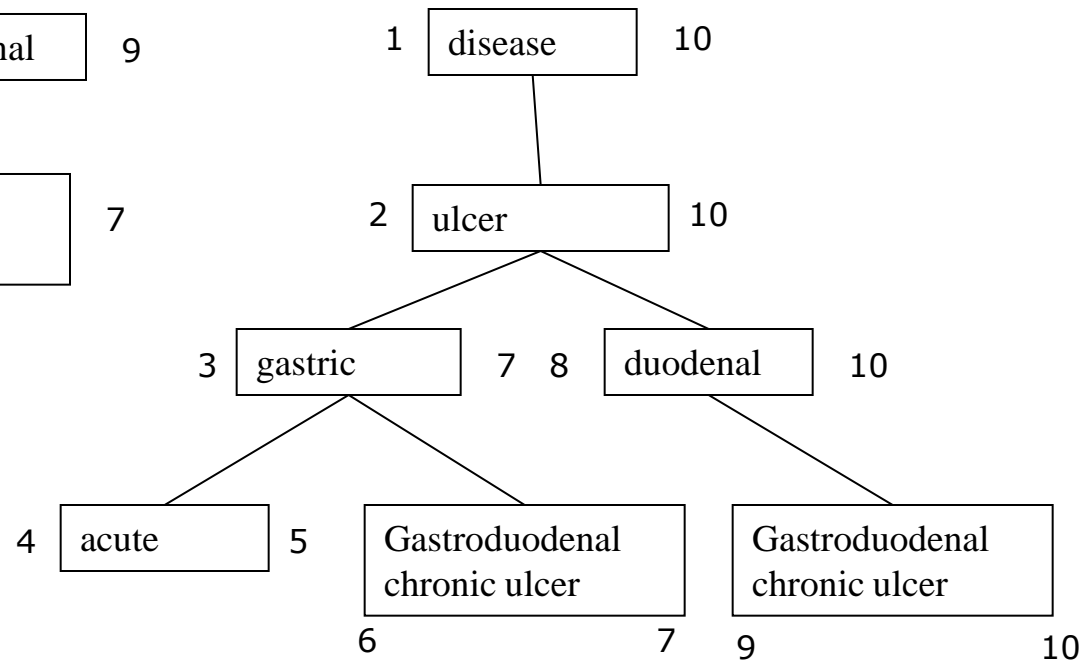
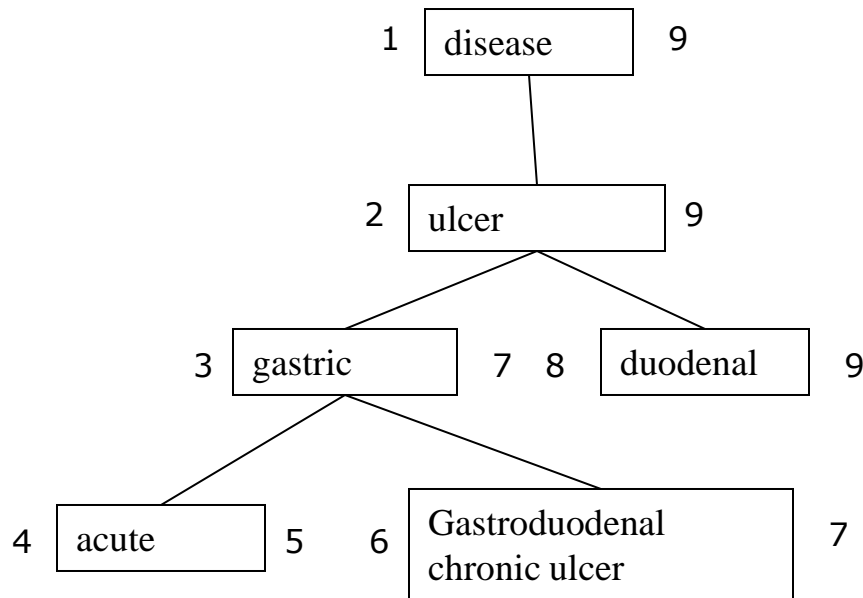


very fast searches of subclasses





# OWL-DB: Multiple inheritance

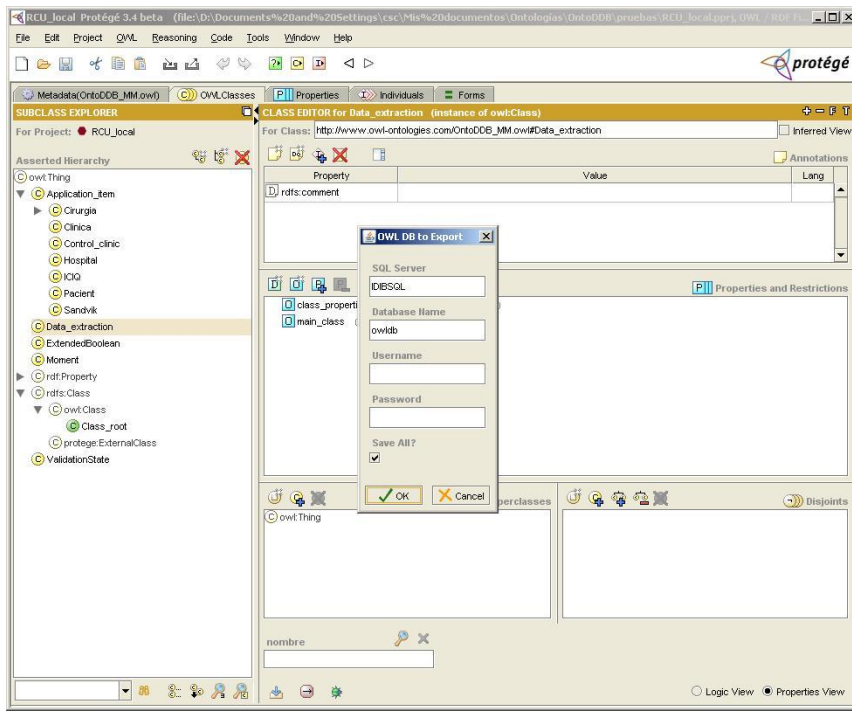




# PROTÉGÉ - PLUGIN OWL-DB

- Backend plug-in for Protégé: change in storage format
  - Ontology repository instead of OWL file
  - Changes in loading and storing processes
- Statements as means of communication between Protégé and repository
- Two storing modes:
  - Save all
    - Removal of previous statements from the database
    - Insertion of all the statements of the ontology
  - Save changes
    - Detection of changes made in the ontology
    - Update of names in the database
    - Insertion of new statements (statements comparison)
    - Removal of old statements

# PROTÉGÉ - PLUGIN OWL-DB



OWL-XML file

Model comparison and  
statements extraction

Stored  
procedures on  
the database

.OWL

JENA  
Java API for  
OWL

OWL-  
DB

# PROBLEMS: Name changes

- Resource name: common identifier between database and Protégé
- User allowed to modify the name
- Solution
  - ChangeListener over Protégé frames
  - List of modified elements
    - Old and new names of each element
  - Names update on DB before storing
    - Update trigger on *resr* table → update of *stat* table

# PROBLEMS: Blank nodes

- Name (URI) as identifier for resources
  - In statements communication with external applications
  - In repository (statements table)
- Anonymous resources have no name
  - Internal names change in each Protégé session
  - Impossible to link old and new names
- Solution:
  - Anonymous resources considered as changed resources in *save changes* mode
  - Namespace: *Anonymous*
  - Penalty on the first storing of each session

# OWL-DB OntoLoad

---

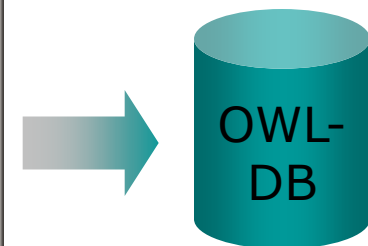
- Module for loading NEW ontologies on database
- OWL file as input
- Use of Jena API for ontology management
- Statements as means of communication with repository
- Capacity to load large ontologies
  - NCI ontology: 500.000 statements

# OWL-DB OntoLoad



The screenshot shows a Windows-style application window titled "OWL-DB OntoLoad". It contains several input fields: "OWL File:" with a path "D:\Documents and Settings\csc\Mis documentos\Ontologías\OntoDDB\pr", "SQL Server:" with "IDIBSQL", "Database Name:" with "owldb", "Username:" (empty), and "Password:" with "\*\*\*\*\*". There is a "Browse..." button next to the OWL File field and a "Start" button below the password field. At the bottom, a text area displays "FINISHED!!!" and "Ontology loaded into database."

Stored  
procedures on  
the database



# OntoDDB-MM: Metamodel

AISEB Protégé 3.1.1 (file:\Q:\UIM\PROYECTES\AISEB\Area%20de%20trabajo\Diseño\O

File Edit Project OWL Code Window Tools Help

OWLClasses Properties Forms Individuals Metadata

**SUBCLASS RELATIONSHIP**

For Project: AISEB

Asserted Hi

- owl:Thing
  - Application\_item
    - Hospital
    - Paciente
    - Analitica
  - Data\_extraction
  - ValidationState

**CLASS EDITOR**

For Class: Paciente (instance of Class\_root)

Name SameAs DifferentFrom

Paciente

rdfs:comment

**Properties and Restrictions**

- 1.1 - id\_paciente (single xsd:string)
- 1.2 - edad (single xsd:int)
- 2.1 - sexo (single owl:oneOf{Masculino Femenino})
- 3.1 - hospital (single Hospital)
- 4.1 - observaciones (single xsd:string)
- 5.1 - texto\_1 (single rdf:XMLLiteral)
- 6.1 - analitica (multiple Analitica)

# OntoDDB-MM: Metamodel

AISEB Protégé 3.1.1 (file:\Q:\UIM\PROYECTES\AISEB\Area%20de%20trabajo\Diseno\Ontologia\AISEB.pprj, OWL DB and Files (.owl or .rdf))

File Edit Project OWL Code Window Tools Help

OWLClasses Properties Forms Individuals Metadata

### CLASS BROWSER

For Project: AISEB

Class Hierarchy

- owl:Thing
  - rdfs:Class (32)
    - rdf:Property (38)
      - owl:DatatypeProperty (16)
        - webDataProperty (28)
          - webLiteralProperty (8)
          - WebMultilineStringProperty
        - owl:ObjectProperty (10)
          - WebObjectProperty (7)
      - Application\_item
      - Data\_extraction
      - Data\_extraction\_inactive (1)
      - ValidationState (2)

### INSTANCE BROWSER

For Class: webDataProperty

Asserted Inferred

multiple slot

- 1.1 - cod\_EP
- 1.1 - cod\_organizacion
- 1.1 - cod\_UP
- 1.1 - descripcion\_campo
- 1.1 - fecha
- 1.1 - id\_hospital
- 1.1 - id\_paciente
- 1.1 - tipo
- 1.2 - decrp\_corta\_EP
- 1.2 - decrp\_organizacion
- 1.2 - edad
- 1.2 - nombre\_hospital
- 1.2 - tag\_obligatorio
- 1.2 - up\_descrp
- 1.3 - valor\_obligatorio
- 1.4 - campo

Types

- webDataProperty
- owl:FunctionalProperty

### PROPERTY EDITOR

For Property: 1.1 - id\_hospital (instance of webDataProperty, owl:FunctionalProperty, internal name is id...)

Name: id\_hospital

rdfs:comment

Annotations

Property	Value	Language
rdfs:label	Id de hospital	es
rdfs:label	Hospital id	en

Domain: Hospital

Range: xsd:int

Allowed values

webDescriptionProperty: false

webIdProperty: false

webRow: 1

webColumn: 1

Functional: ☒ Functional

InverseFunctional: ☐ InverseFunctional



# OntoDDB: Web Application

**Aiseb Project** riozano | INICIO | SALIR | MANTENIMIENTO USUARIOS

Paciente >> 10010 >>

**Navegacion**

Mensajes  
Paciente

**Otras properties**

Analítica

Values must be saved before leaving any page or you will lose them.

**Detalles de: 10010**

**Propiedades basicas**

Id de paciente	10010	Iniciales	RLR
Sexo	Masculino	Edad	45
Observaciones	Ejemplo de campo multilinea		

Update

**Propiedades**

Hospital	Hospital 1
Ejemplo de literal	

Update

© 2007 Fundació Clínic per a l'Estudi i l'Assistència en Oncologia. All rights reserved.

Class root

Botón para actualizar

WebLiteralProperty

WebObjectProperty

Web DataProperty

Ejemplo de aplicación en OntoDDB.



# OntoDDB-DataExtraction

VALID Protégé 3.1.1 (file:\Q:\UIM\RECERCA\VALID\Area%20de%20trabajo\Diseño\Ontología\VALID.pprj, OWL DB and Files (.owl or .rdf))

File Edit Project OWL Code Window Tools Help

owl:Thing  
owl:Class (32)  
owl:Property (38)  
Application\_item  
Data\_extraction (1)  
Data\_extraction\_inactive (4)  
ValidationState (2)

CLASS BROWSER

For Project: VALID

Class Hierarchy

INSTANCE BROWSER

For Class: Data\_extraction

Asserted Inferred

:NAME

de\_patient

Types

Data\_extraction

INDIVIDUAL EDITOR

For Individual de\_patient (instance of Data\_extraction)

Name SameAs DifferentFrom

de\_patient

rdfs:comment

main\_class

Patient

class\_properties

hospital ↔ patients

Annotations

Property

# OntoDDB-DataExtraction

OntoDDB - DataExtraction

Directory: D:\Documents and Settings\rlozano\Escritorio Browse...

SQL Server: IDIBSQL

Database Name: OWL

Model: VALID

Username:

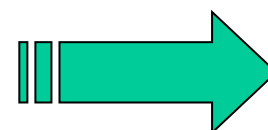
Password: \*\*\*\*\*

Start the Data Extraction

FINISHED!!!

Generated Files:

de\_patient.csv



de\_patient.csv

id	patient_id	baseline_crf	country	study	bi
2404	011073	2406	France	09/30/1968	10/27/20
2429	011079	2431	France	02/10/1964	05/06/20
2456	011082	2458	France	10/03/1983	01/18/20
2481	012091	2483	France	04/04/1970	08/01/20
2530	011083	2532	France	03/01/1940	02/08/20
2628	011076	2630	France	06/01/1945	02/06/20
2653	012096	2655	France	01/21/1947	07/05/20
2874	011087	2876	France	08/25/1985	05/24/20
2899	011086	2900	France	04/18/1977	04/17/20
3068	012090	3070	Germany	11/15/1968	04/28/2
3238	012069	3240	France	12/15/1941	11/29/20
3359	011080	3360	France	09/02/1947	11/24/20

# ADVANTAGES

- Simplification in development
  - Reduction in development cost
  - Reduction in development time
  - No need to programming, so no need of technical resources for doing it
  - Applications construction is reduced to analysis and design phases
  - Prototypes available from the beginning
- Ease of maintenance
  - Great flexibility respect to later modifications
  - Reduction in costs
- Allows to take advantage of economies of scale

# ADVANTAGES

- Use of ontological technologies
  - Ontological analysis clarifies the knowledge structure
  - Processable
  - Integration with Semantic Web
  - Reuse of models
- Use of standards: OWL, Protégé (de facto)
  - Models communication
  - Models sharing
  - Easily extensible with new functionalities
- Multilingualism
- Help to establish homogeneous criteria in organization
- Domain independent



# FUTURE WORK

---

- Upcoming version 2.0
  - Web services
  - OntoDDB API improvements (waiting time, ...)
- Extract more explicit knowledge
  - Division between data model and presentation
  - Incorporation of more web functionality
  - Incorporation of procedures
- Changes in ontology editing process
  - Progressive loading of information
  - Introduction to *views* of an ontology
- Future applications:
  - Clinical repository for patients data collection (not research)

