PELLUCID

a new platform for
organisationally
mobile public employees

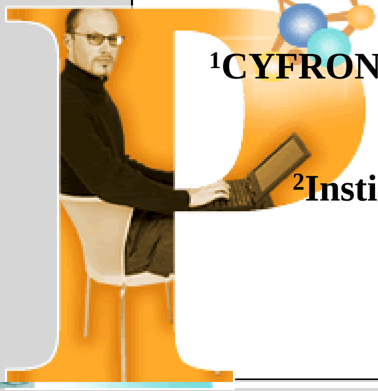
Experiences with Using Protégé in a Knowledge Management Application: The Pellucid Study

Krzysztof Krawczyk[1], Mariusz Dziewierz[1],
Michal Laclavik[2], Alvaro E. Arenas[3]

[1]CYFRONET - Academic Computer Centre of the University of Mining and Metallurgy, Cracow, Poland

[2]Institute of Informatics, Slovak Academy of Sciences, Bratislava, Slovakia

[3]CCLRC Rutherford Appleton Laboratory, Oxfordshire, UK



Overview

1. The Pellucid System
2. Generic Ontology
3. Contact Ontology
4. Document Ontology
5. Deployment Module
6. Experiences Using Protégé
7. Future Work and Open Questions

PELLUCID
a new platform for
organisationally
mobile public employees

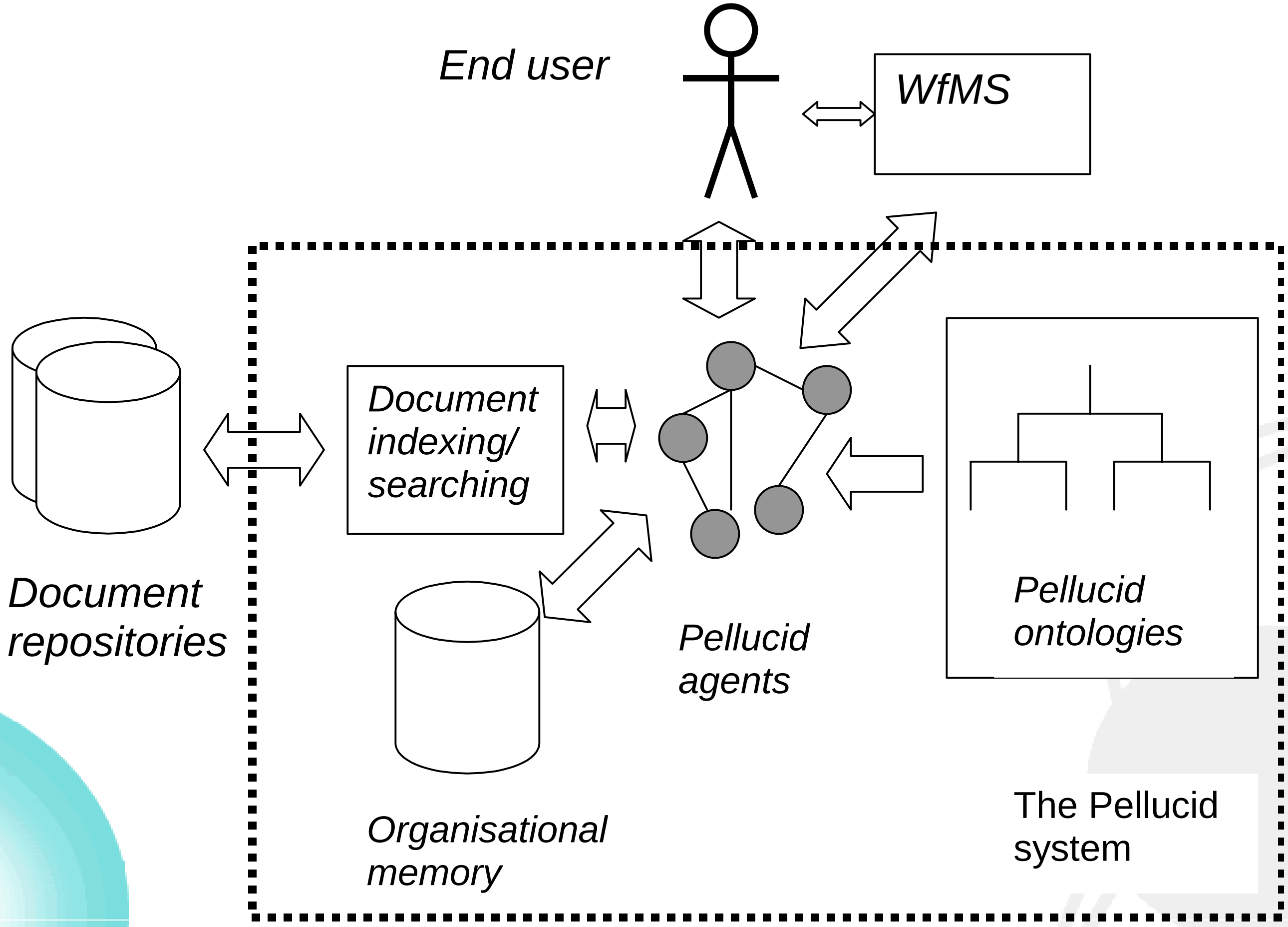
The Pellucid System

- Pellucid aims to enhance **knowledge management for public employees**.
  - It is developing a customisable software platform …
    - … which is based on software agents …
    - … and which will be put into place at three pilot sites (local or regional governments in Italy and Spain).
- The scope of Pellucid
  - Pellucid is concentrating on ‘experience management’
  - The innovation lies in relating experience management to the working context.
- The Pellucid System provides:
  - Access to relevant documents, contacts and other advice
  - It is also able to offer advice spontaneously as well as respond to request for help

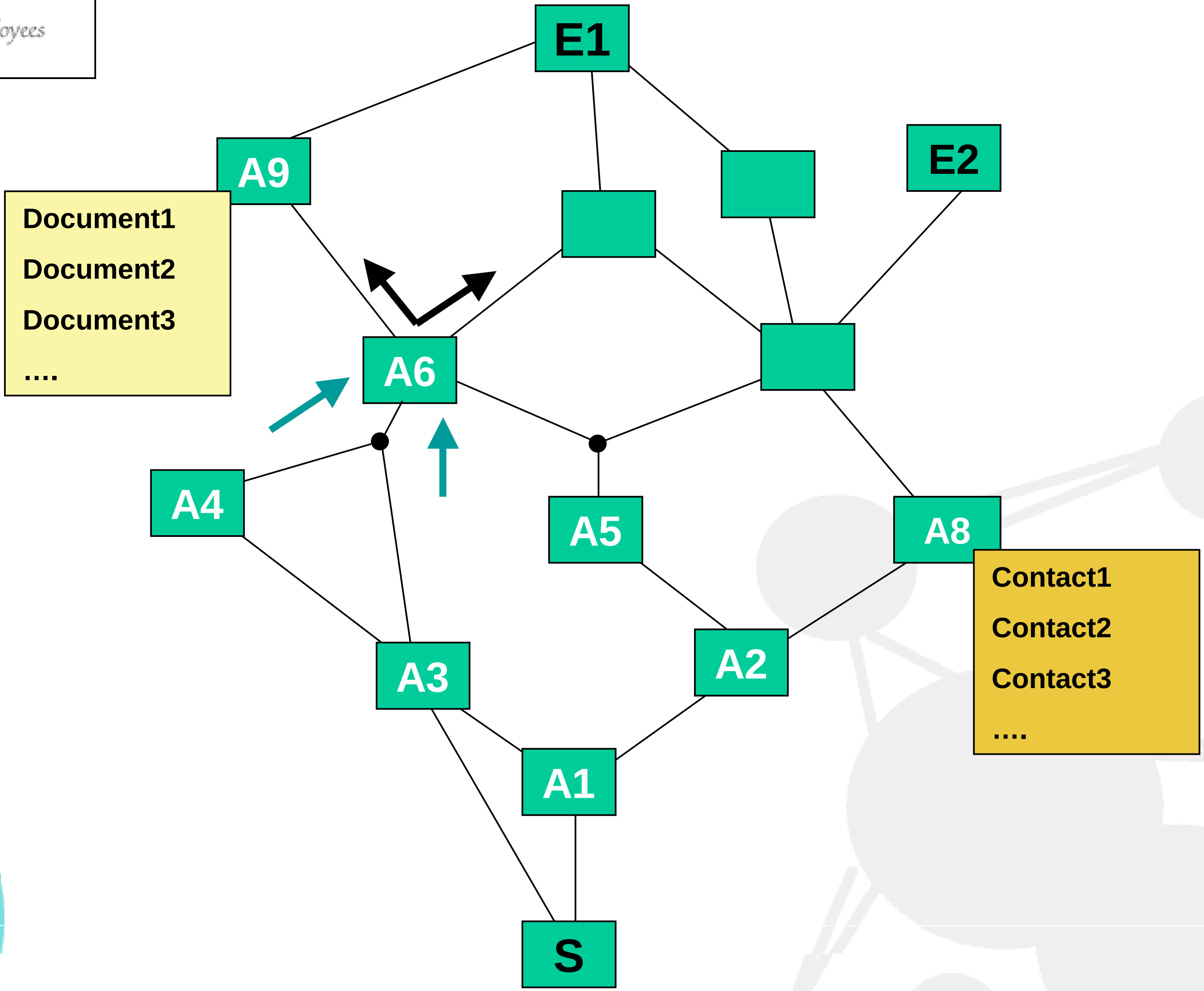
PELLUCID

a new platform for organisationally mobile public employees

The Pellucid System Conceptual View



Workflow-Based Knowledge Management



Document1

Document2

Document3

....

Contact1

Contact2

Contact3

....

Pellucid Agents

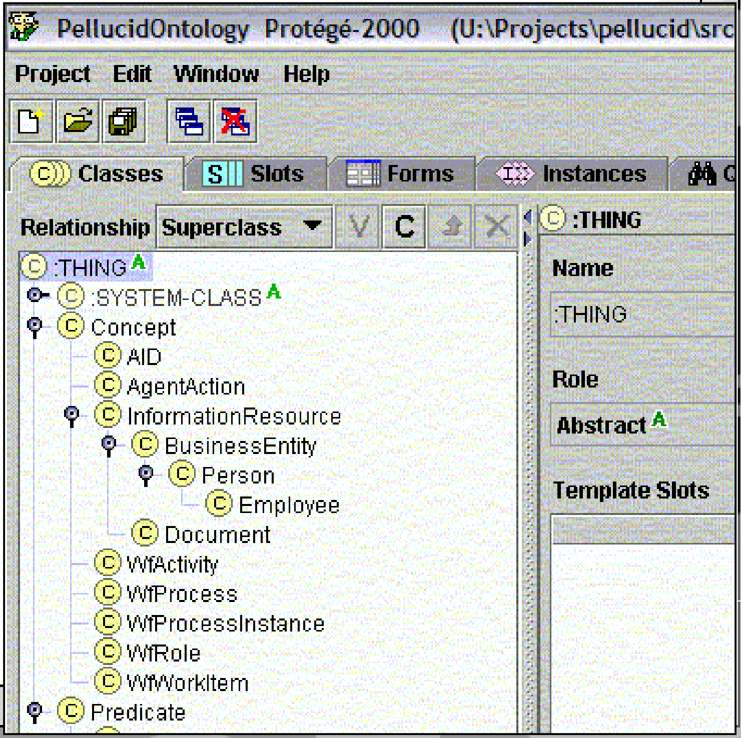
- Interaction Layer (IL)
  - The Personal Assistant Agent (PAA)
- Process Layer (PL)
  - The Monitoring Agent (MA)
  - The Role Agent (RA)
  - The Task Agent (TA)
  - The Capitalization Agent (CA)
- Access Layer (AL)
  - The Information and Search Agent (ISA)

Organisational Memory

- Store all relevant knowledge gathered, created and accumulated during the system life cycle
  - Ontologies and instances
  - Indexes
- The technology – object database
  - Ontology instances stored as objects using JDO (Java Data Objects)
  - Lucene as text indexing engine

Ontologies

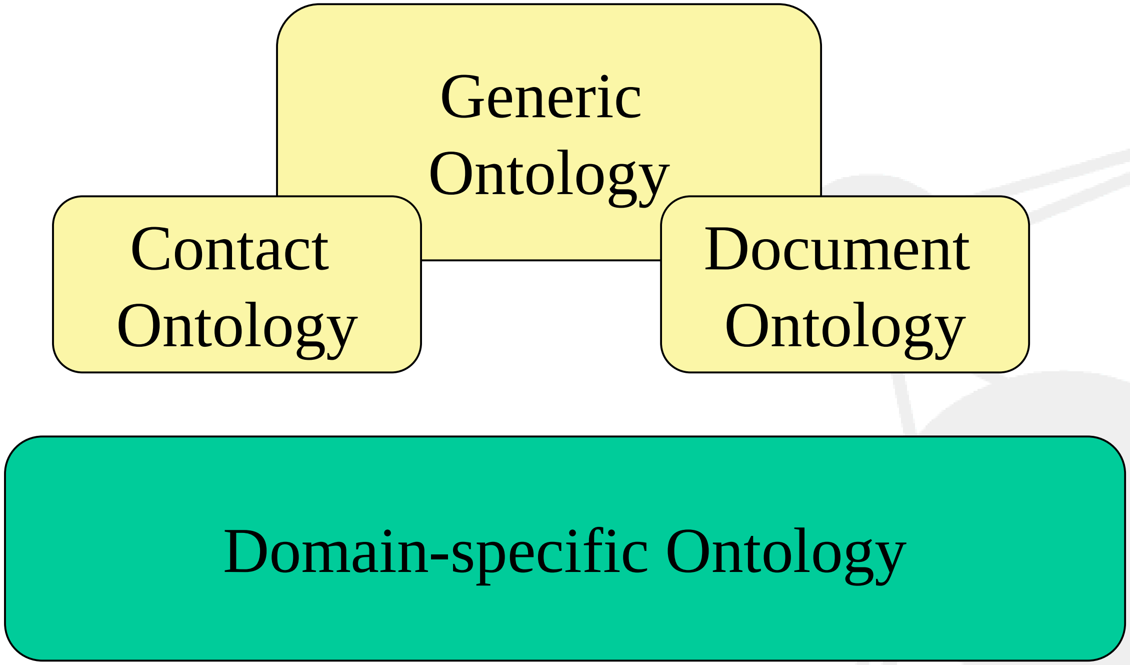
- Ontology = formal conceptualisation of a domain
  - CLASSES of things
  - PROPERTIES they have
- Why use an ontology?
  - Allows structured communication of knowledge
  - Guarantees mutual understanding
- Ontologies are basis for agent communication
  - JADE agents talk in FIPA-SL with user-defined ontologies.



- :THING
  - :SYSTEM-CLASS
  - Concept
    - AID
    - AgentAction
    - InformationResource
      - BusinessEntity
        - Person
          - Employee
      - Document
    - WfActivity
    - WfProcess
    - WfProcessInstance
    - WfRole
    - WfWorkItem
  - Predicate

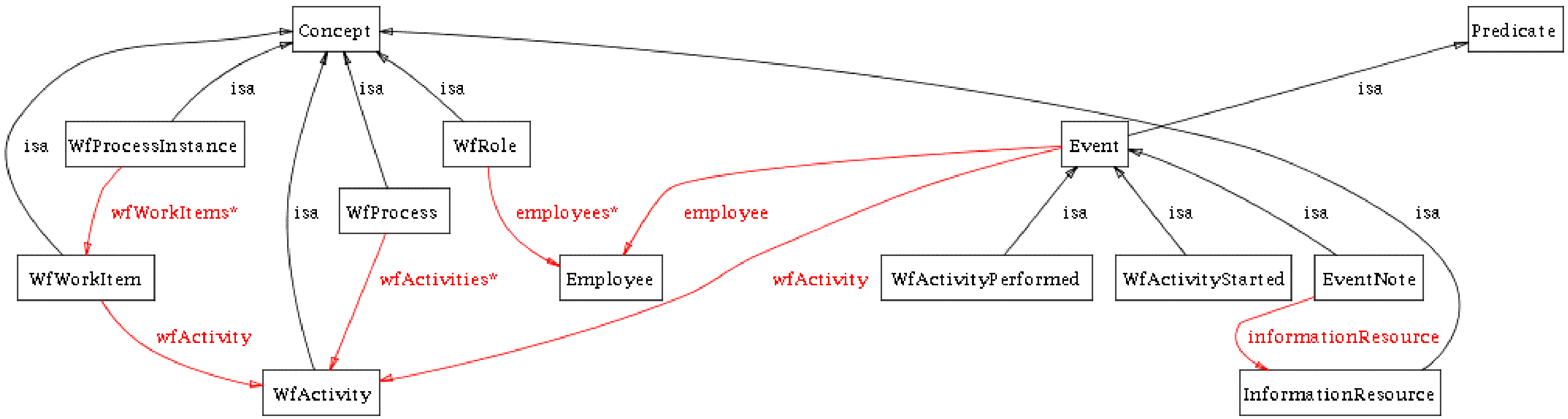
Ontologies

- Generic ontology provides core concepts
- Extended into Contact and Document ontologies
- Support extension into Domain-Specific ontologies



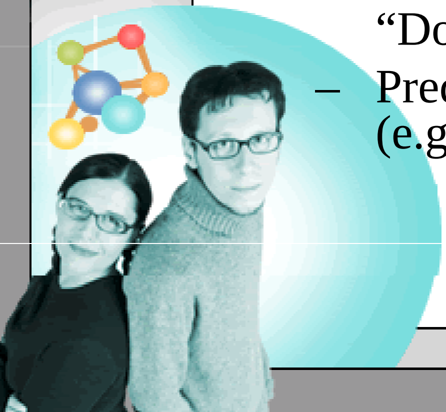
PELLUCID
a new platform for organisationally mobile public employees

Generic Ontology



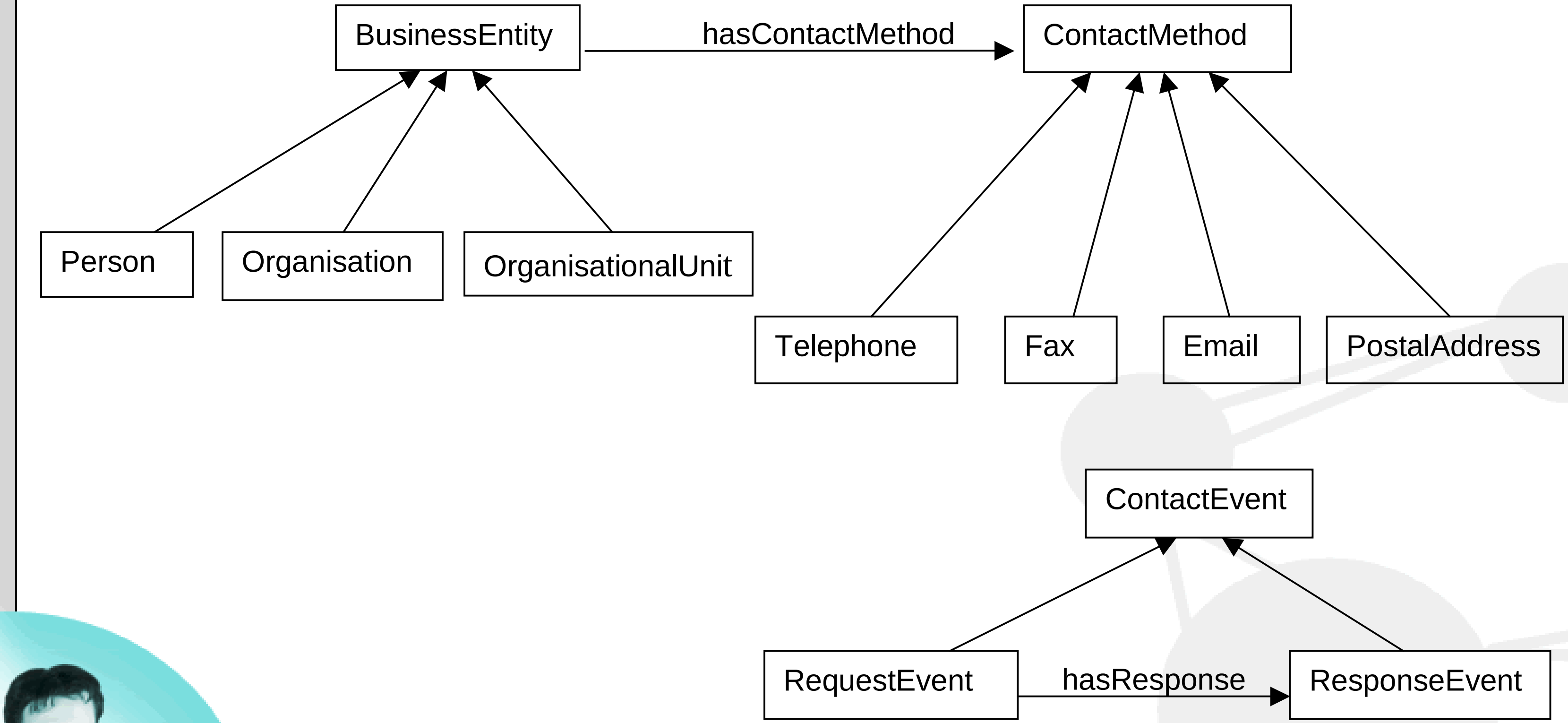
Ontologies in Pellucid consist of:

- Concepts – concrete elements (e.g.: “Employee” or “WfActivity” or “Document”)
- Predicates – relations between two or more Concepts (e.g.: “DocumentUsed”)



PELLUCID
a new platform for
organisationally
mobile public employees

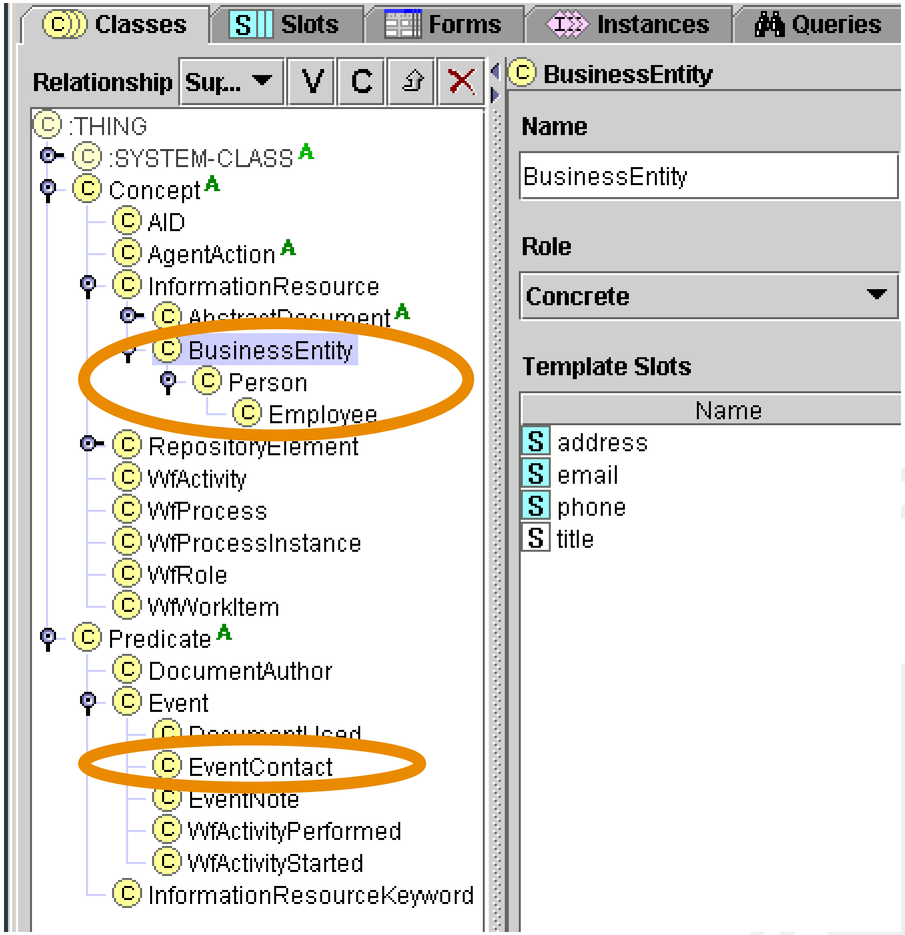
Contact Ontology



PELLUCID

a new platform for organisationally mobile public employees

Implementing Contact Ontology in Protégé



- :THING
  - :SYSTEM-CLASS A
  - Concept A
    - AID
    - AgentAction A
    - InformationResource
      - AbstractDocument A
      - BusinessEntity
        - Person
          - Employee
    - RepositoryElement
    - WfActivity
    - WfProcess
    - WfProcessInstance
    - WfRole
    - WfWorkItem
  - Predicate A
    - DocumentAuthor
    - Event
      - DocumentUsed
      - EventContact
      - EventNote
      - WfActivityPerformed
      - WfActivityStarted
    - InformationResourceKeyword

BusinessEntity

Name

BusinessEntity

Role

Concrete

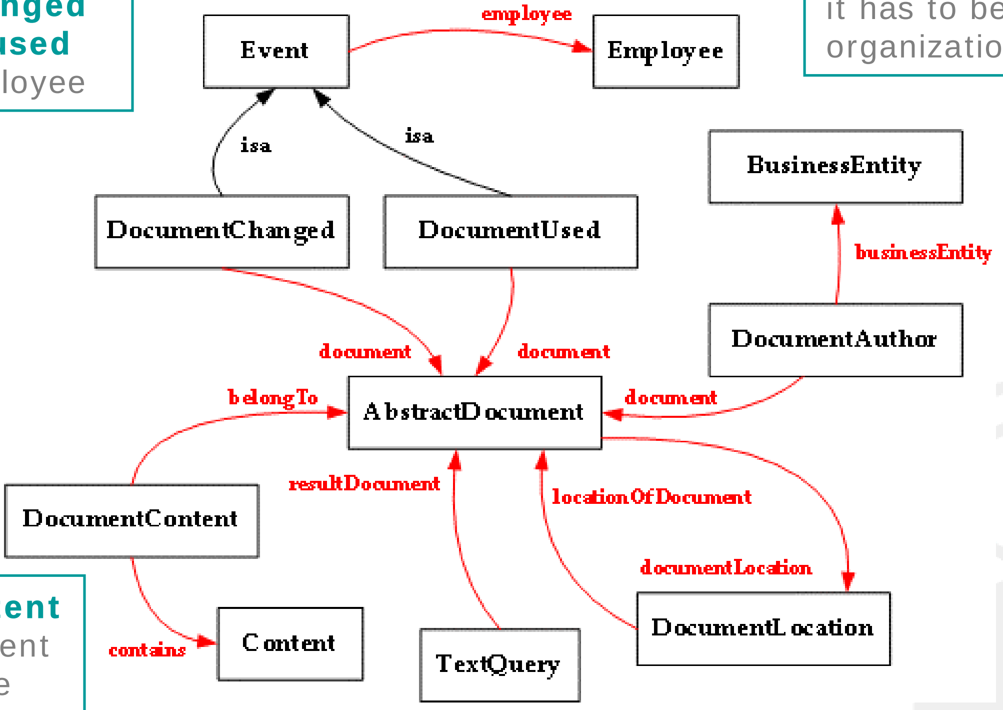
Template Slots

Name
S address
S email
S phone
S title

PELLUCID
a new platform for
organisationally
mobile public employees

Document Ontology

The document can be **changed** as well as **used** by the employee



The author of the document can be any **BusinessEntity** which means that not necessary it has to be Pellucid organization employee

The **Content** of document allows the Pellucid to know what is inside of the document

TextQuery is predicate to perform **full text searching** within indexes

PELLUCID

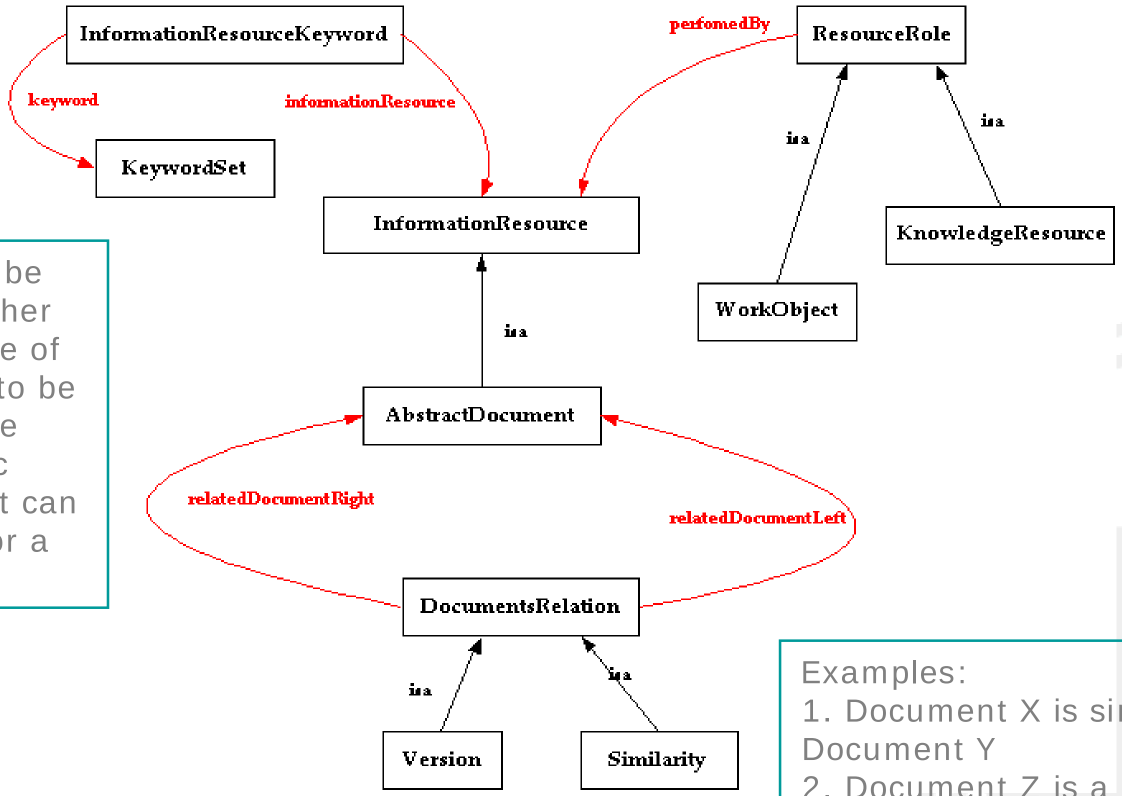
a new platform for organisationally mobile public employees

Document Ontology

Document inherits Relation from InformationResource

InformationResource can be described using **keywords**

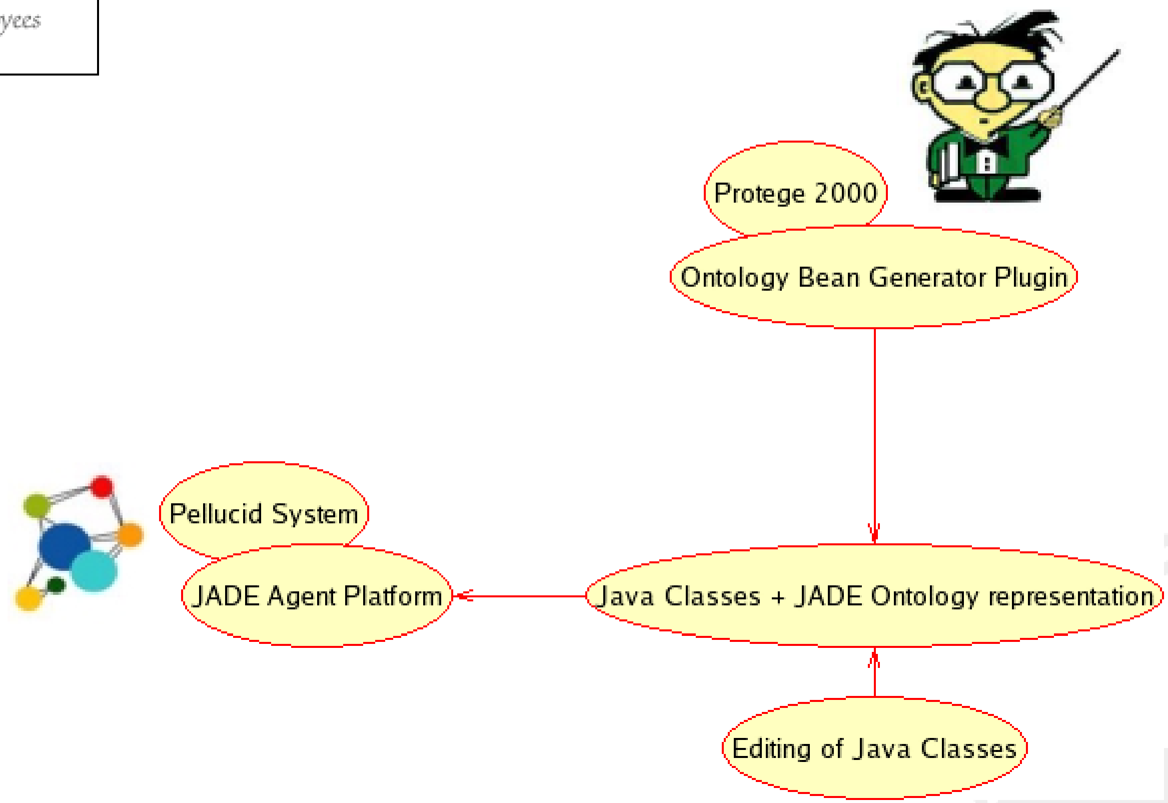
And **plays** different **roles** in organization



Document can be related to another one. The nature of the relation is to be described in the domain specific ontology. But it can be a **version** or a **similarity**.

Examples:
1. Document X is similar to Document Y
2. Document Z is a version of Document U

Ontology Conversion to JADE



Protege 2000

Ontology Bean Generator Plugin

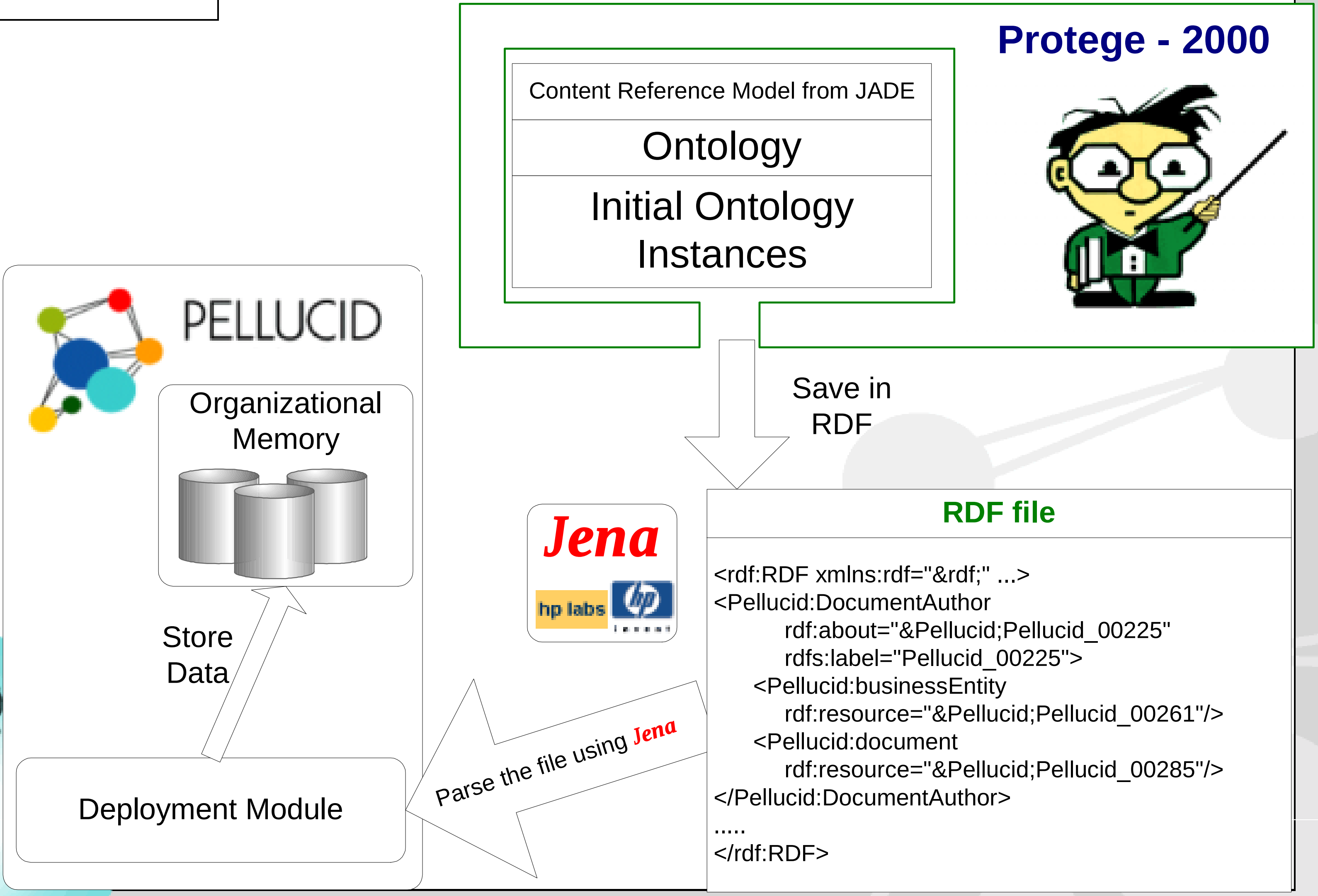
Java Classes + JADE Ontology representation

Editing of Java Classes

Pellucid System

JADE Agent Platform

Deployment Module



```
<rdf:RDF xmlns:rdf="&rdf;" ...>
<Pellucid:DocumentAuthor
        rdf:about="&Pellucid;Pellucid_00225"
        rdfs:label="Pellucid_00225">
    <Pellucid:businessEntity
        rdf:resource="&Pellucid;Pellucid_00261"/>
    <Pellucid:document
        rdf:resource="&Pellucid;Pellucid_00285"/>
</Pellucid:DocumentAuthor>
.....
</rdf:RDF>
```

Experiences using Protégé

- We have used the following Protégé extensions:
  - RDF storage model
  - Ontology Bean Generator Tab plug-in
  - Ontoviz Tab plug-in

- Some extra features in development:
  - User-friendly way to display ontology instances (knowledge)
    - We did this in the JADE implementation by adding a method to each ontology class. So Instances were displayed as string.
  - User-friendly way to display ontology information
    - Assigning human readable text to ontology elements and slots
    - Multi-language support for presenting ontology

Future Work and Open Questions

- More reasoning in Pellucid
  - Domain-specific reasoning
  - Experience knowledge discovery where ontology instances are input for:
    - Classification – categorization into previously defined groups
    - Clustering – finding groups of experiences
    - Association rules – detecting trends, relations
  - Experience generalization and aggregation
- Extend the work to other contexts
  - Contact and Document management are only instances of a more general case
  - Time management
- Explore other facilities associated to Protégé

Thank you !