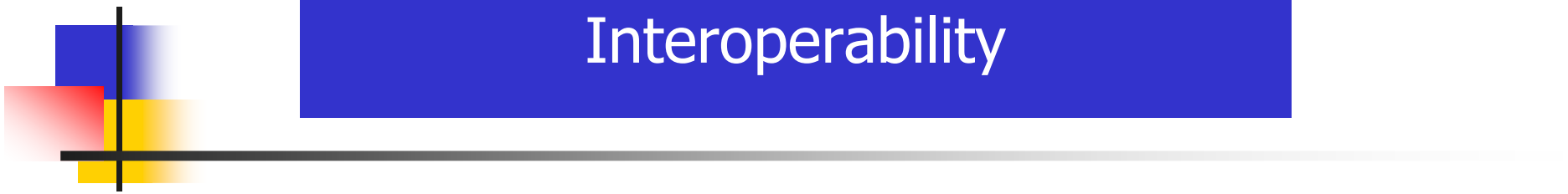


Semantic Annotations for Interoperability

Ontologies, a smart way towards interoperability (IAE, Paris, April 13 2006)

Nacer.Boudjlida@loria.fr

UHP Nancy 1-LORIA, http://www.loria.fr/~nacer

Presentation Outline

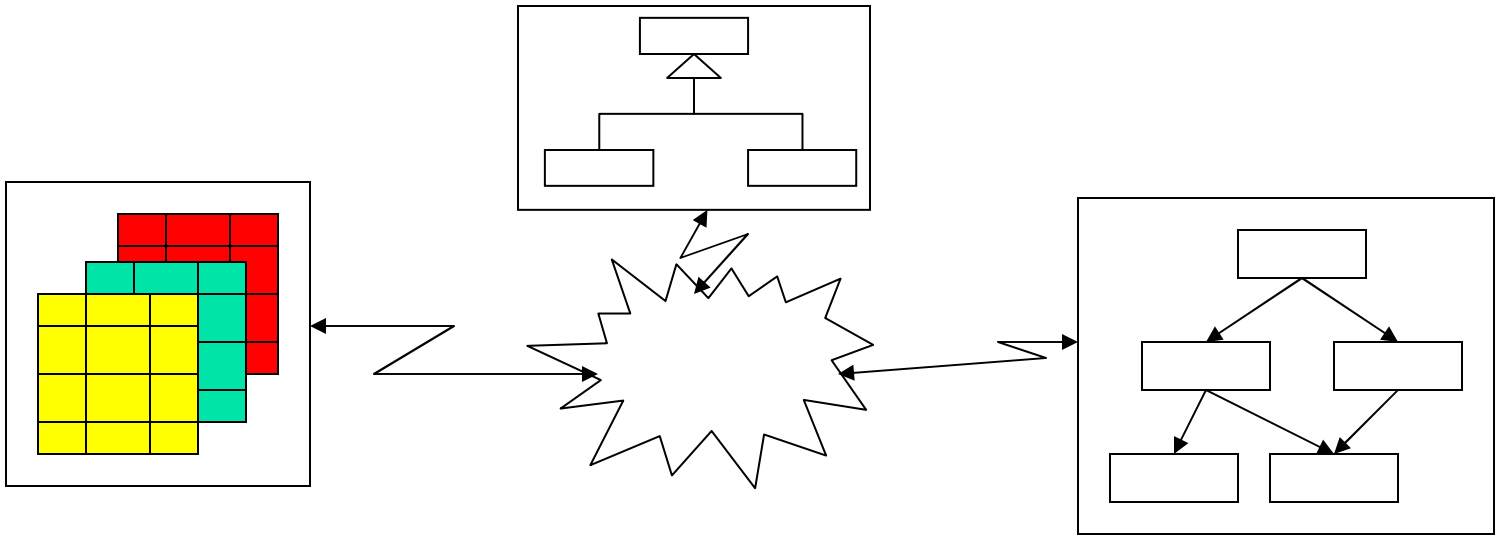
I. Integration and Interoperability
   1. Problem
   2. Approaches (« process » vs « objets »)

II. Ontology and Semantic Interoperability

III. Enterprise Modeling and Semantic Interoperability

Integration and Interoperability

- *Large scale*
  - Locally/widely distributed enterprises
  - Federated enterprises, Virtual Enterprises, etc.
  - Enterprise Alliances, Merges, etc.



Integration and Interoperability

- *Small Scale* : Illustrative *Example*
  - concat : string x string → string
  - string1: < size, value > (<4, ABCD>)
  - string2: < value, ending mark > (<EF, #>)
  - Result:
    - <5, ABCDEF>
    - OR <ABCDEF, #>

- *Solutions:*
  1. Process-Oriented Interoperability
  2. "Object" Oriented Interoperability

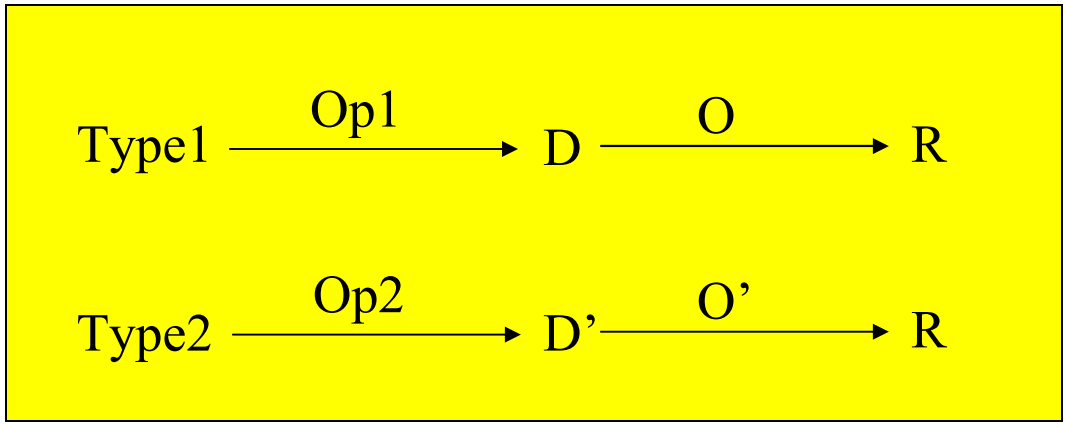
Process vs Object-Oriented Interoperability

- *Back to the example*:
  - Concat : string x string → string
  - string1: < size, value >
  - string2: < value, ending mark >

- *Solution 1* : Process-Oriented Interoperability
  - Concat1 : string1 x string1 → string1
  - Concat2 : string2 x string2 → string2
  - One or Two implementations of concat

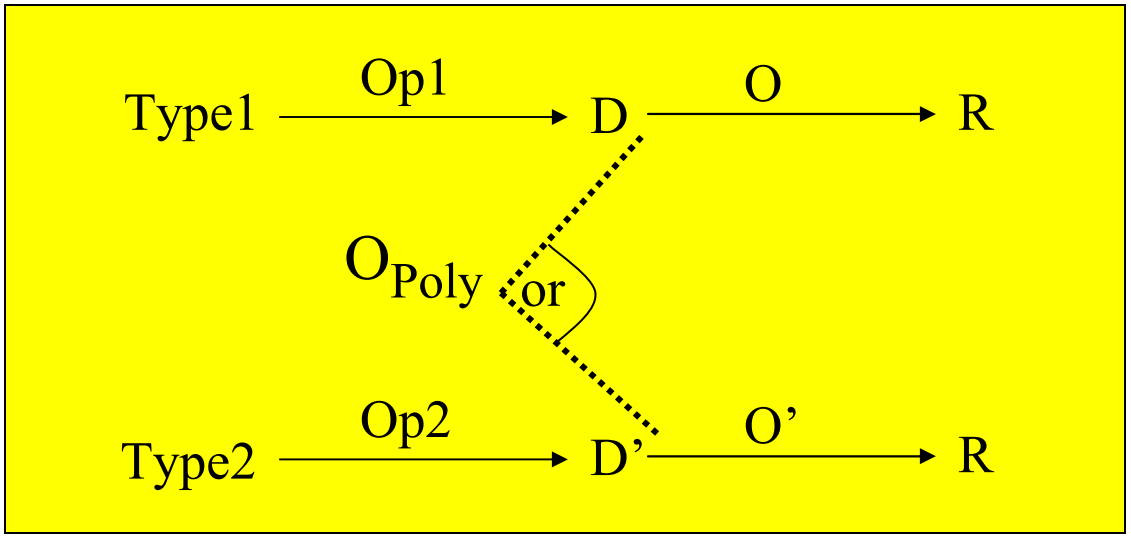
1/3. *Polymorphism with no overloading nor coercion*

- Op1: Type1 $\rightarrow$ D
- Op2: Type2 $\rightarrow$ D'
- O: D $\rightarrow$ R
- How to apply O on Op2(x : Type2)?
- Two implementations of O



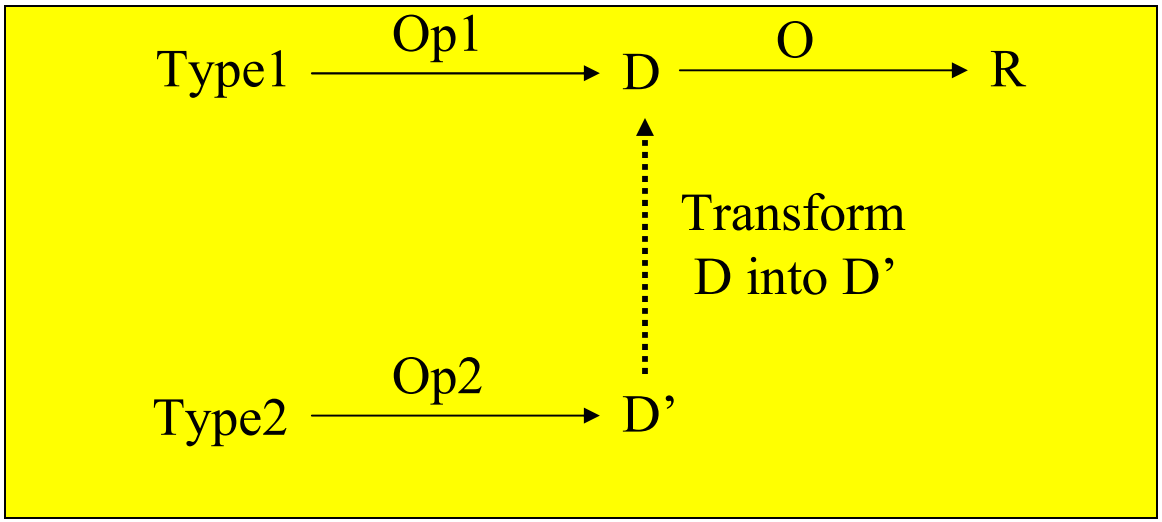
2/3. *Polymorphism without coercion, with overloading*

- Apply O on Op2(x : Type2): A unique name (O_{poly})
- Invoke O or O' depending on the type of parameter



3/3. *Polymorphism with type coercion*

- Apply O to Op2(x : Type2)



- Note: Similar situation when changing O into O'

Integration and Interoperability

- *Solution 2* : Object-Oriented interoperability
  - Concat : string1 x string1 → string1
  - Concat : **g**(string2) x **g**(string2) → **f**(string1)
  - **g** : string2 → string1
  - **f** : string1 → string2

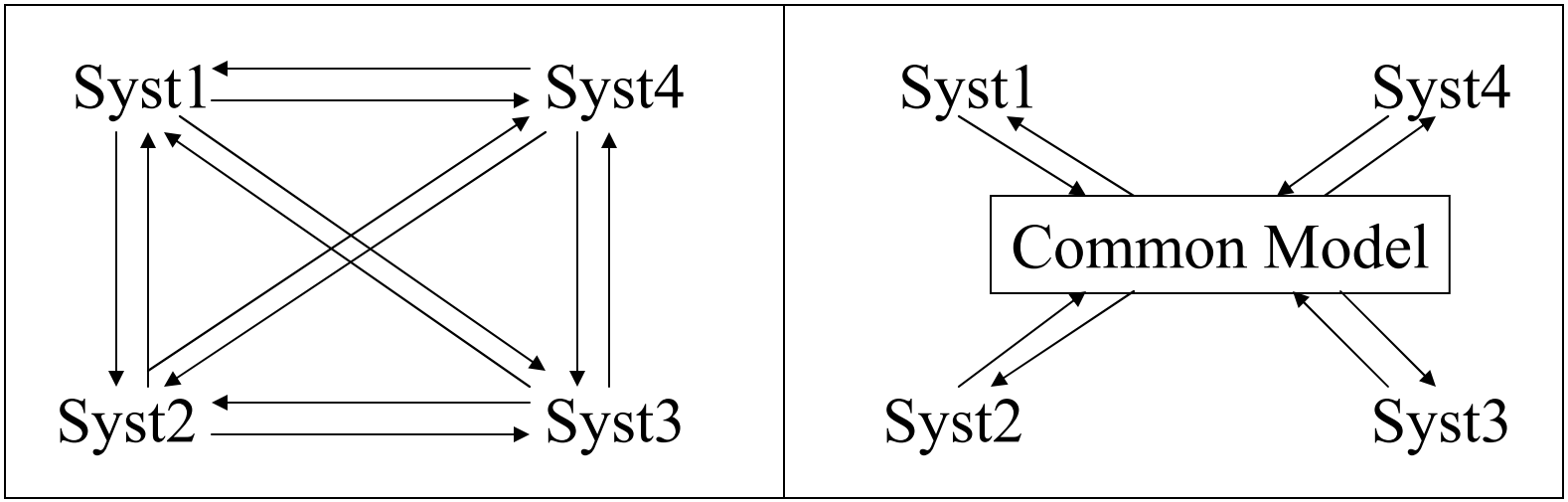
- A single implementation
  - Structure Transformation/Mediation

Object-Oriented Interoperability

- How to determine g and f?
- Data structure complexity?
- Structured vs (semi-)/unstructured?
- Data Semantics?
- Meta-data availability?

Object-Oriented Interoperability

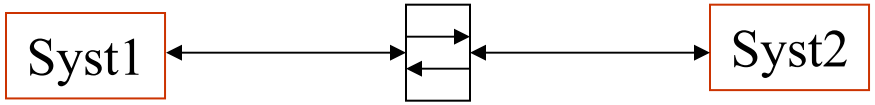
1. Naive Approach : n*(n-1) converters
2. Common Model: 2*n converters



- What a "common model"?
  - Ad hoc
    - *Example* : Data Model of a common repository
  - Specific
    - *Example* : Relational, JDBC, ODBC, UEML
  - Standard
    - OMG/ODMG Object Model
    - XML
    - WfMC Interoperability Interface
    - OMG/CORBA IDL

- What an "architecture"?
  - Encapsulation/Wrapping

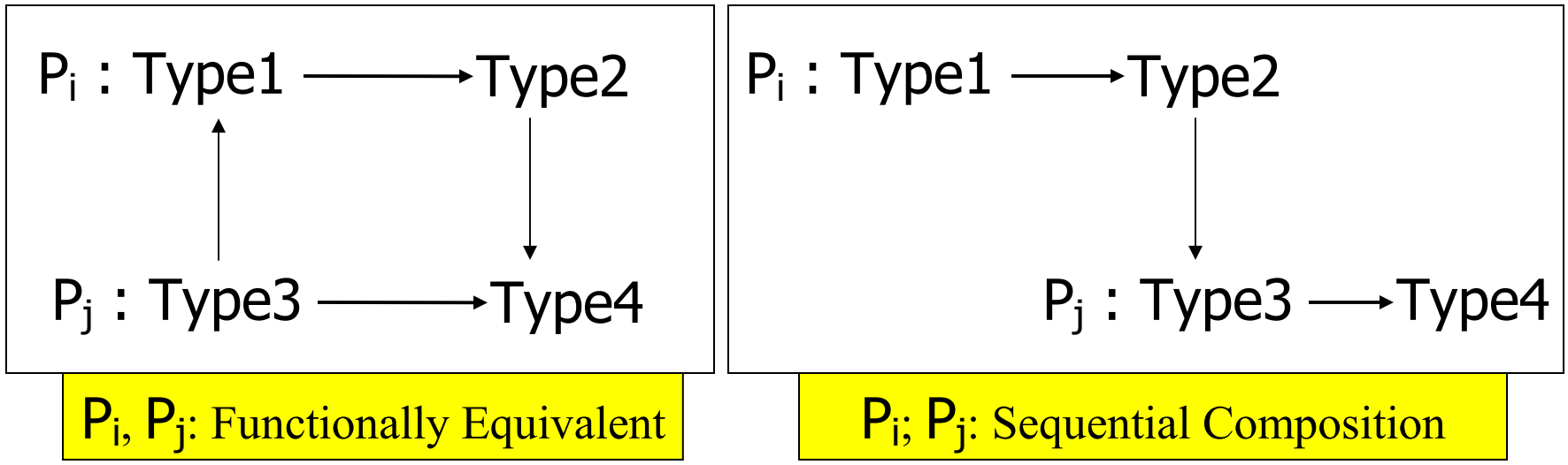
  - Mediation ("Format negociators")



Integration and Interoperability

- *As a summary*:

  A Program/System P_i (on a site S_i) wants to use data/objects/services produced/offered by a Program/System P_k (on a site S_k)



P_i, P_j: Functionally Equivalent

P_i; P_j: Sequential Composition

Integration and Interoperability

- Software System : {Components}
  - Possibly heterogeneous and distributed components
  - That communicate
    - Data, Results, Events, Requests for services, etc.
  - AND Cooperate in task achievement
    - Synchronization
    - Sharing
    - etc.

Integration and Interoperability

- Dimensions:
  1. Interfaces : *look and feel*
  2. Plateforms : Transparency
     - Technologies, Operating Systems
     - Distribution, Protocols, etc.
  3. Data :
     - Communication and sharing (despite the possible heterogeneity)
  4. Control : Notification/Reaction to events
  5. Process Models : Rules that govern the behaviour of the software systems

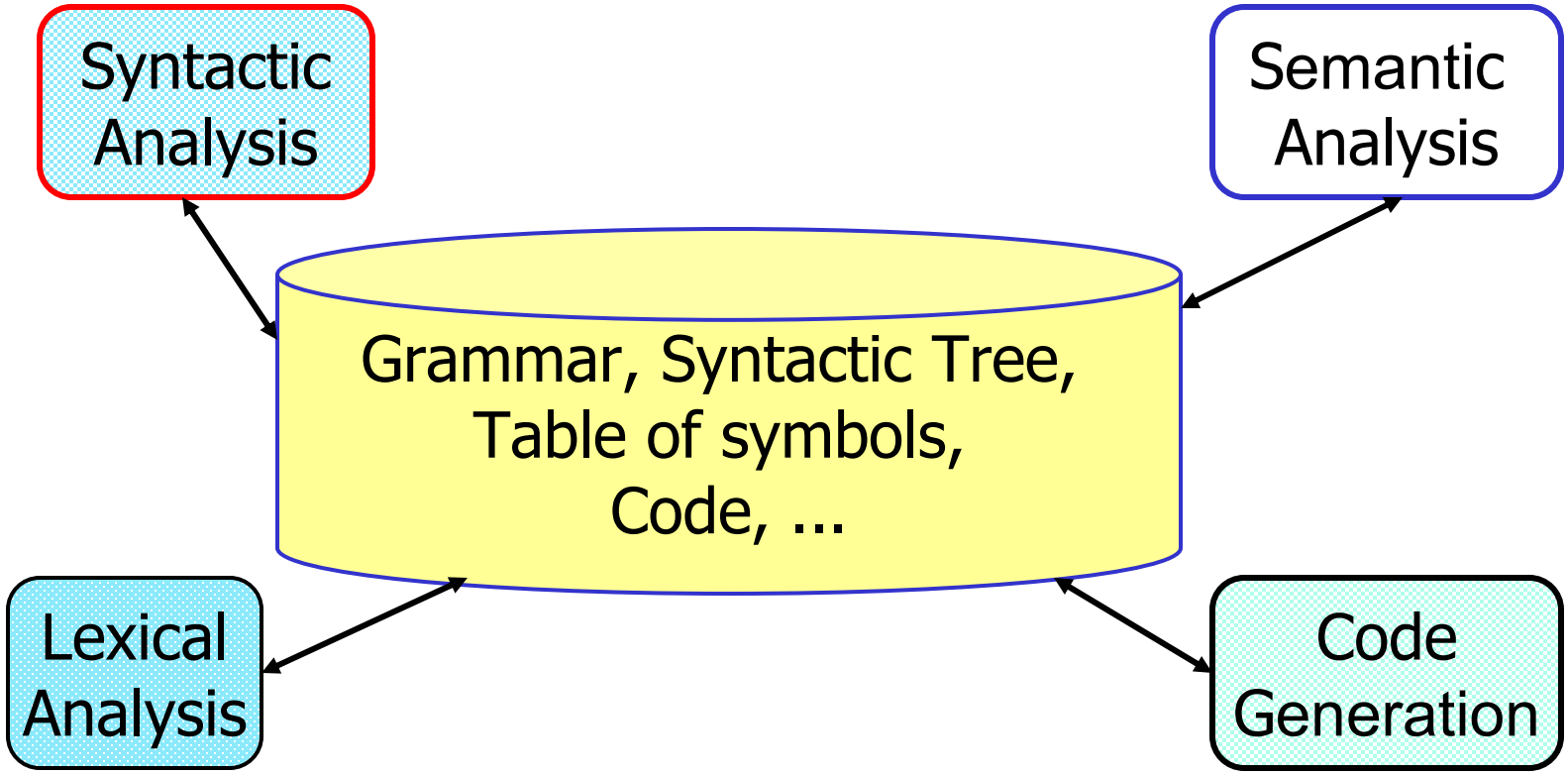
Models for sharing and exchange

- *(Some) Models for the* communication and the sharing of data/information

1. Data flow *(pipe and filter)*
2. Repository (private, shared)
3. Client/Server

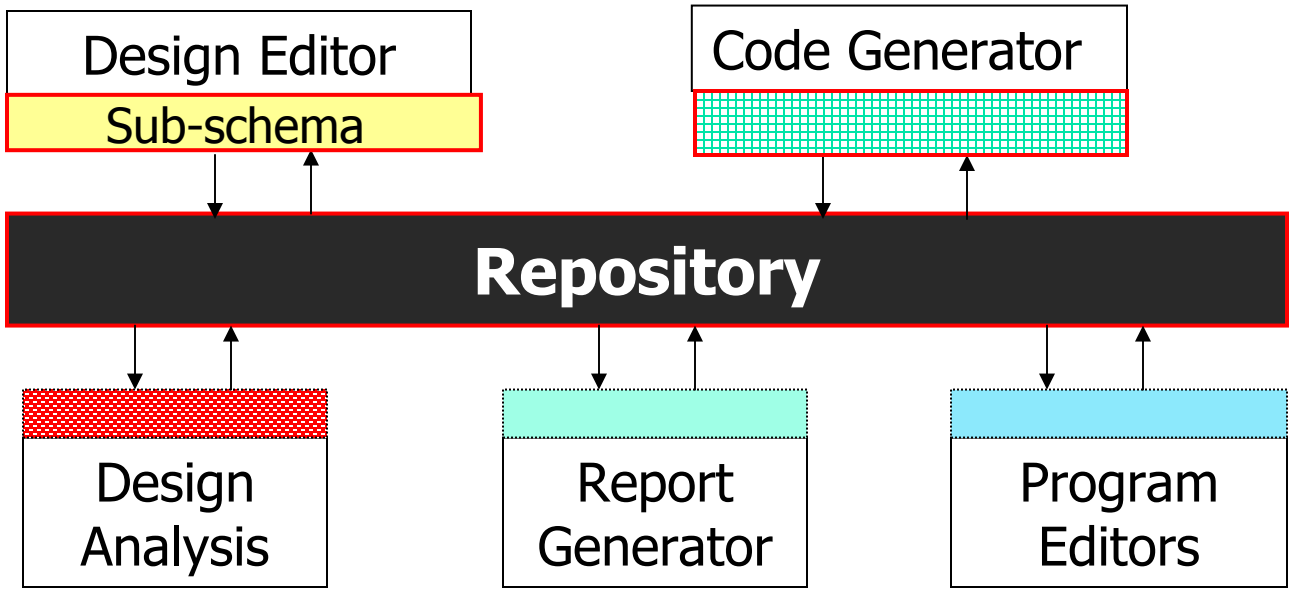
Repository-based Model

1. *Repository* : Common « Database »
   - Example: Compiler (tight integration)



Shared Repository Model

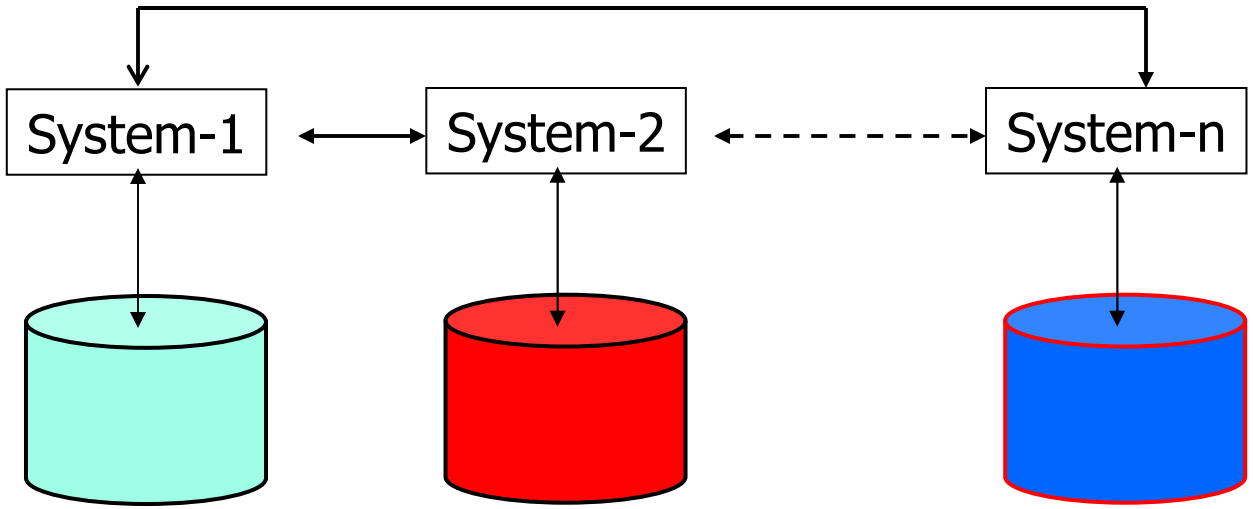
- Example: CASE Environment
- 'Enveloppes' (Wrappers)



Private Repository Model

2. Private Repository

- Communication by message passing



Private Repository Model (cont'd)

Private Repository

(+) Independent sub-systems: production of data without carying about their usage by others

(-) Negociation/Mediation of formats (often) required

Repository-based Model (cont'd)

- *Commun repository*

(+) Efficient mean for exchange (no transfer)

(+) Protection/Sharing thanks to sub-schemas

(+) Repository Manager services

- Back-ups
- Recovery in case of failure
- Security
- Access Control

Repository-based Model (end)

- *Commun repository*

(-) Common Data Format/Model or mediators

(-) Specific security, back-up, etc. policies

(-) Evolutivity : Integration of new systems

(-) Distribution of the Repository

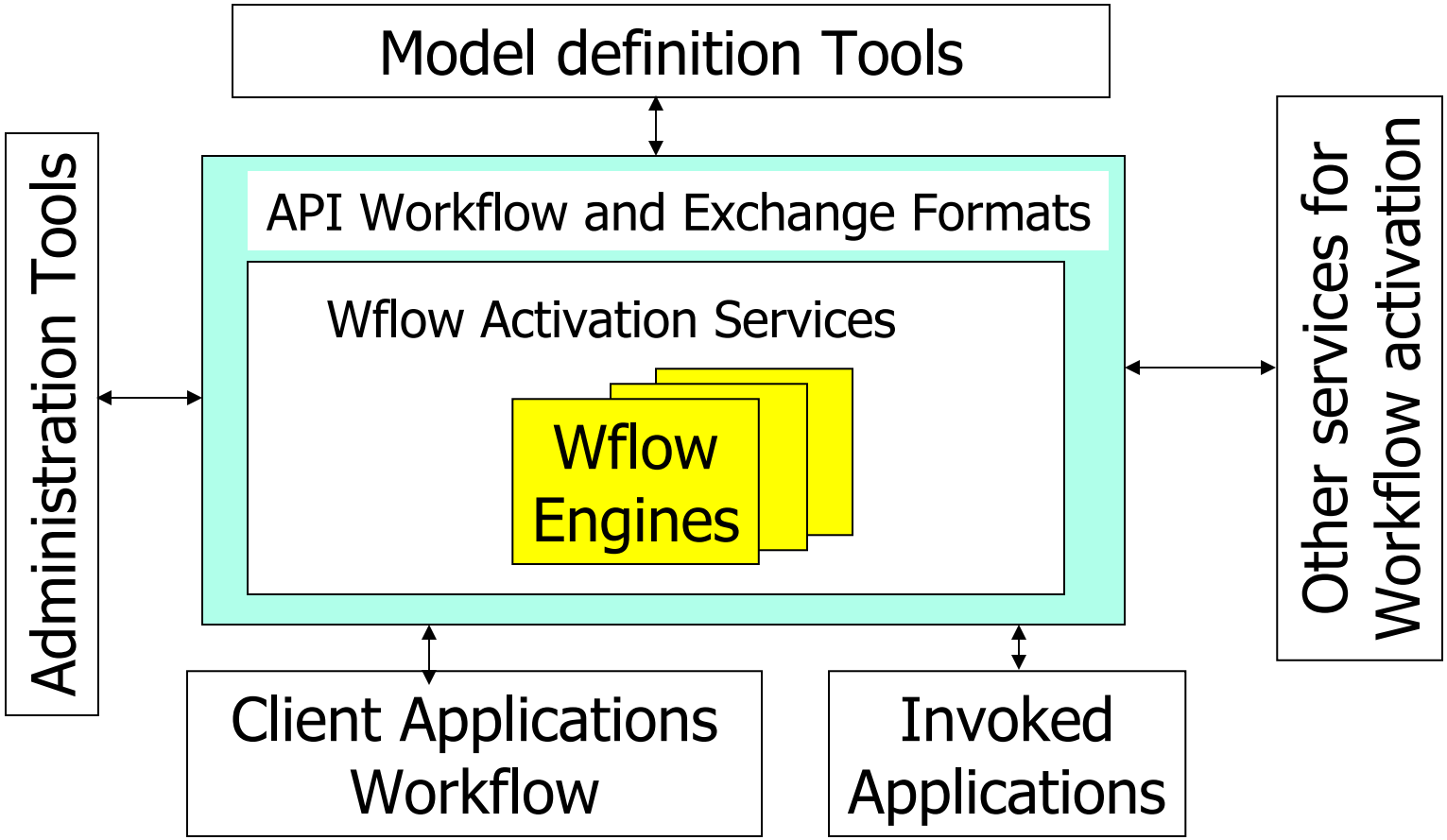
Architectures for interoperability

- Generic Architectures

- Reference Architectures:
  - To describe a class of systems
  - To be able to compare the systems of a domain

2/2. Reference Architectures: Example

- WfMC : Workflow Management Coalition



Reference Architectures (end)

- Other examples: CORBA, UDDI, etc.
- Syntactic Definitions
  - Op1 : Type1 $\rightarrow$ Type2
  - Op2 : Type1 $\rightarrow$ Type2
  - Op1 :: Op2 ?
- No semantics
  - For operations/services
  - For objects (content)
- Currently (over the Web):
  - Ontology
  - Semantic Annotations

Integration and Interoperability : Concluding Remarks

- Exchange, Communication between heterogeneous systems
- Middlewares, Formats and protocols
  - common
  - ad hoc
  - standards
- Do not prevent from mediation
- Lack of semantics: 'understand' (and consistently interpret) not only the structure but also the content

Presentation Outline

I. Integration and Interoperability
   1. Problem
   2. Approaches (« process » vs « objets »)

Questions?

Presentation Outline

I. Integration and Interoperability
   1. Problem
   2. Approaches (« process » vs « objets »)

II. Ontology and Semantic Interoperability

III. Enterprise Modeling and Semantic Interoperability

Semantic Interoperability

- An approach: Use of ontologies
- Ontology = branch of philosophy about 'essence of things' that exist in world, in a domain

- Definition :
  1. [Gruber] : Formal and explicit Specification *of a shared conceptualization*
  2. [W3C] : defines the terms used to describe and represent a domain of knowledge

Semantic Interoperability

- *Purpose of an ontology*: to be used by persons, applications, information bases, … that need to share information about a domain

- Examples:
  - Biology, Genomics
  - Automotive industry
  - Finances
  - …

Ontology Anatomy : Types of components

1. Concepts or classes:
   - Represent a set of entities within a domain
   - Organized as taxonomy(s)
2. Relations:
   - Represent inter-actions among concepts
   - May be organized as a taxonomy

Ontology Anatomy : Types of components

3. Axioms:
   - 'Sentences/Assertions' always true
   - Constrain class values
   - Define arguments of the relations

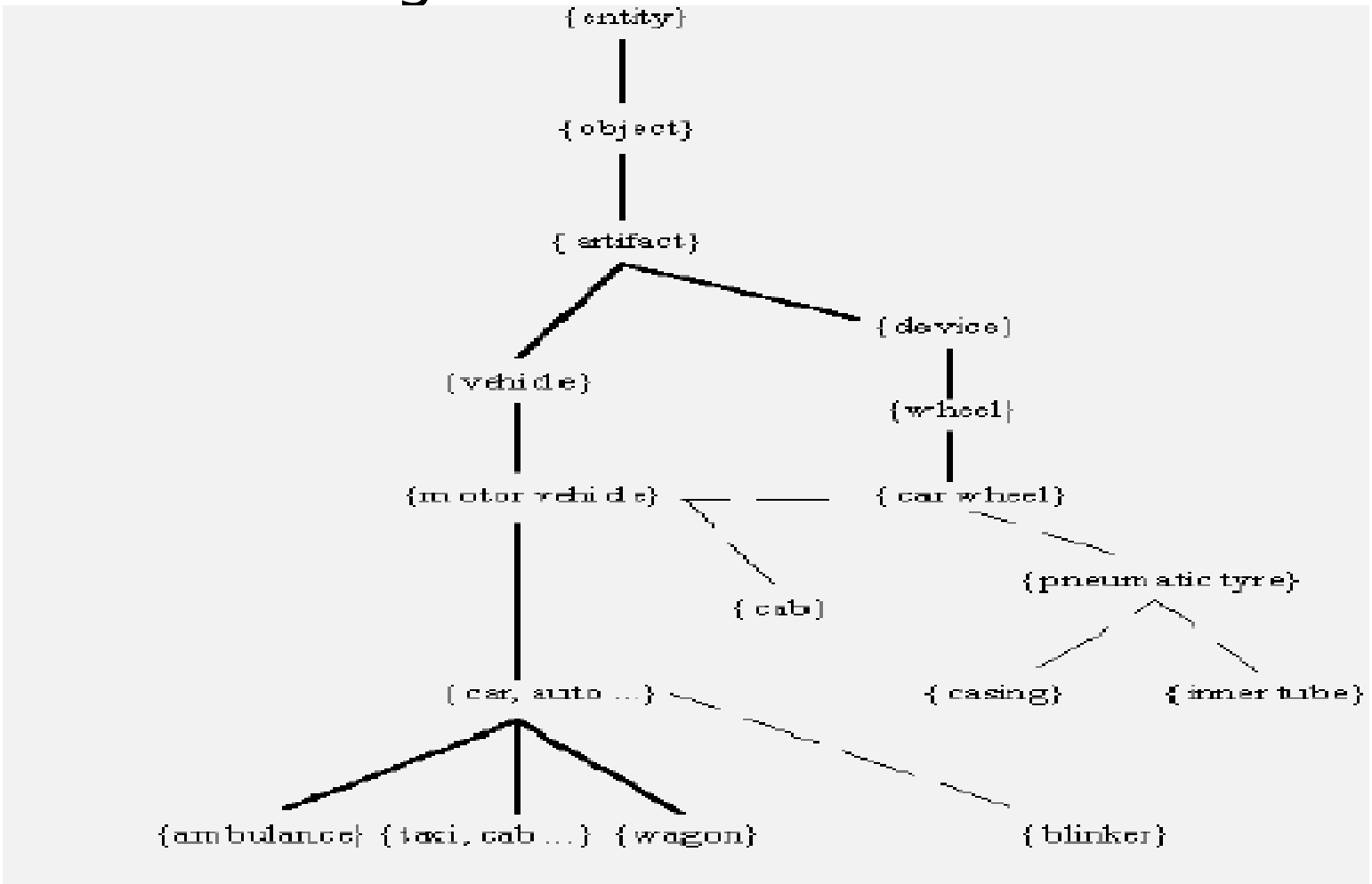
4. Instances:
   - Representation of individuals of the classes

Semantic Interoperability

Levels	Concepts	Examples
Higher	The most general	▪ Actor, process ▪ Event, Goal ▪ Business Rule
Domain/ Application	Specific to an application domain	▪ Employee ▪ Bill ▪ Order
Lower	Specific attributes and concepts used for composing the concepts in the domain level	▪ Date, duration ▪ Name, Address ▪ Amount, Cost ▪ …

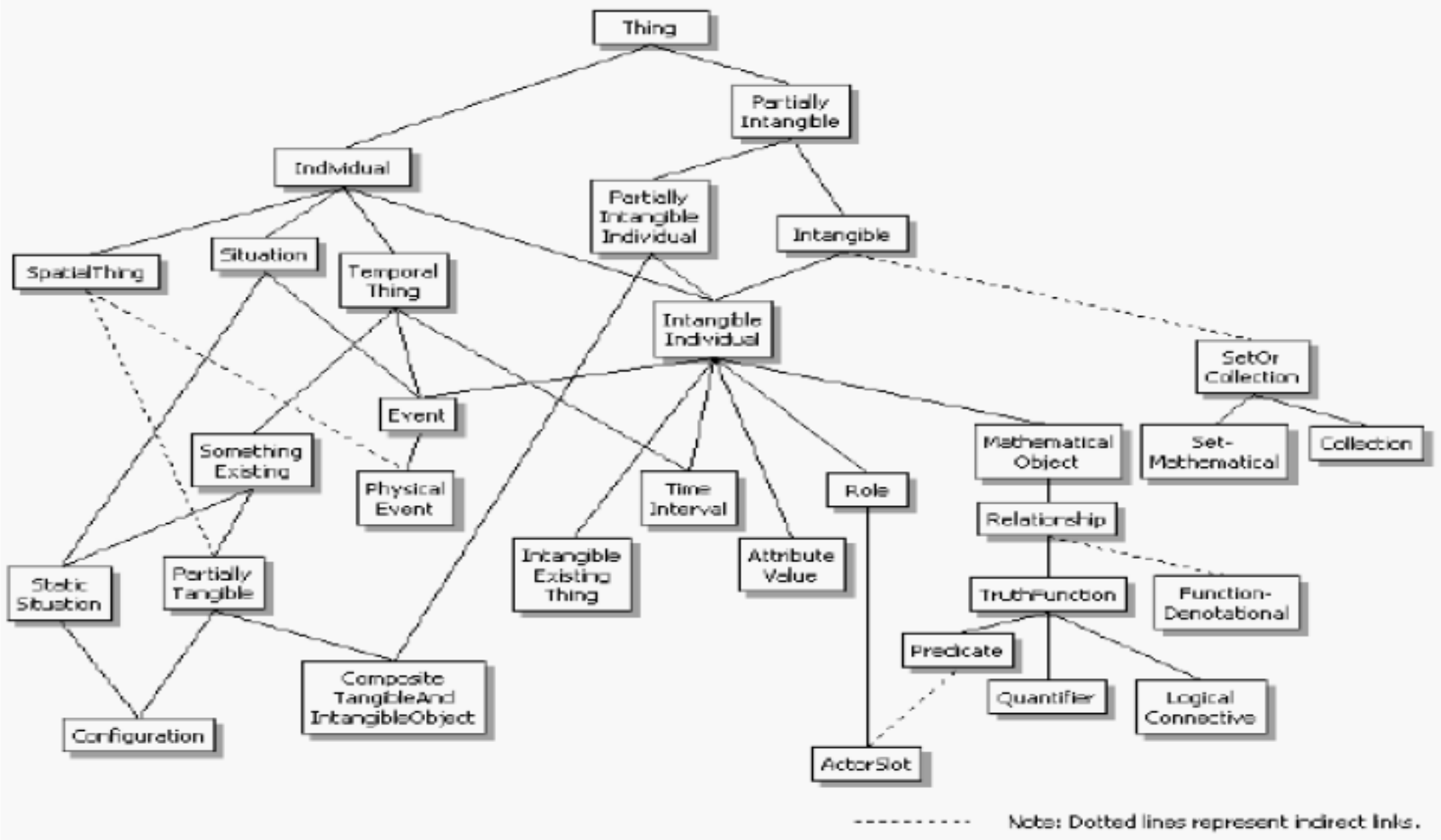
Domain Ontologies: Examples

- Wordnet: English Lexical Base



Upper Cyc

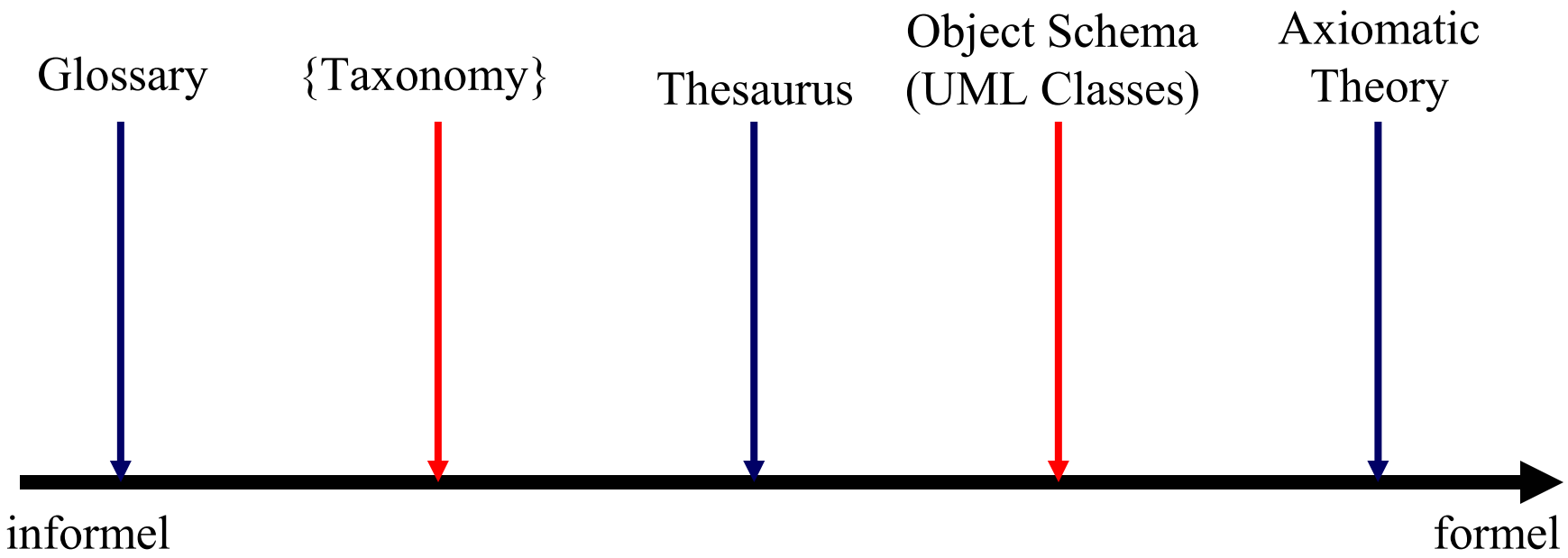
- Common terms



Ontologies : Languages

- Typology of Ontology description languages
  - *Unformal* : natural language (cf. glossaries)
  - *Unformal structured* : « restricted » natural language
  - *Semi-formal* : Artificial language usually without theoretic foundations
  - *Formal* : terms defined with
    - Formal Semantic
    - Theorems
    - Proof
    - Properties (completeness, consistency, …)

Ontologies : Languages



Ontologies : Technology

- Representation Languages:
  - UML, Express, Ontolingua, XML, etc.
- Knowledge Exchange Languages:
  - KIF: Knowledge Interchange Format
  - PIF: Process Interchange Format
  - CDIF: Common Data Interchange Format
- Translation Tools
  - Ontolingua Translators
  - Step Tools (STandard for the Exchange of Product model data, ISO-1033, often associated with EXPRESS, for the specification of information model)
  - etc.

Ontologies : Expected Benefits

- *Idea* : Improve the communication between humans and machines for:

1) Assisting the communication between persons
2) Interoperability between systems
3) Improving the process and/or the system engineering quality

Ontology and Interoperability

1) Systems Interoperability: thanks to 'translators' between
   - Different Modeling Methods
   - Different paradigms
   - Different languages and Sofwtare Tools

- In this situation: Ontology ≈ Common Exchange Format

2) Improve the process and/or the system engineering quality

1) Re-use:

- Ontology = basis for a formal encoding of important « things » (entities, attributes, processes, relations) within a domain
- Ontology = Formal Representation that might
  - Be re-used
  - Become a shared component

in Software systems

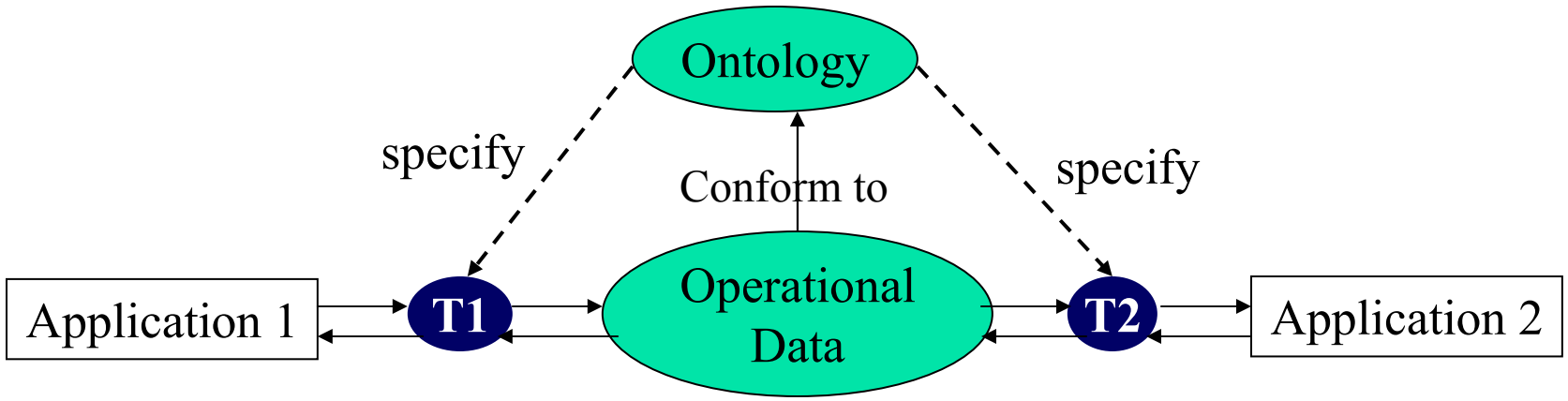
Ontology and System Engineering

2. Information Retrieval:
   - Ontology ≈ meta-data for
     - Indexing
     - Accessing

   An information repository
3. Safety/Reliability:
   - Base the verification of constraints on the formal specification of the ontology
4. etc.

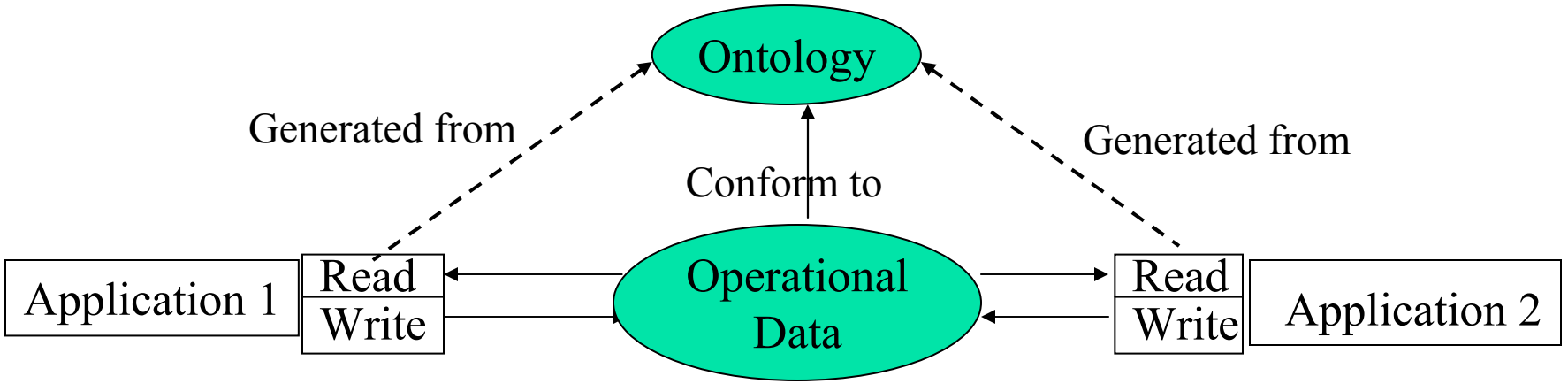
Ontology: Usage Examples

1. Data access through a shared ontology:
   - Exemple:
     - Ontology ≈ Library of process Models (PIF)
     - Applications : Different Formats (ex: IDEF3, ILOG)



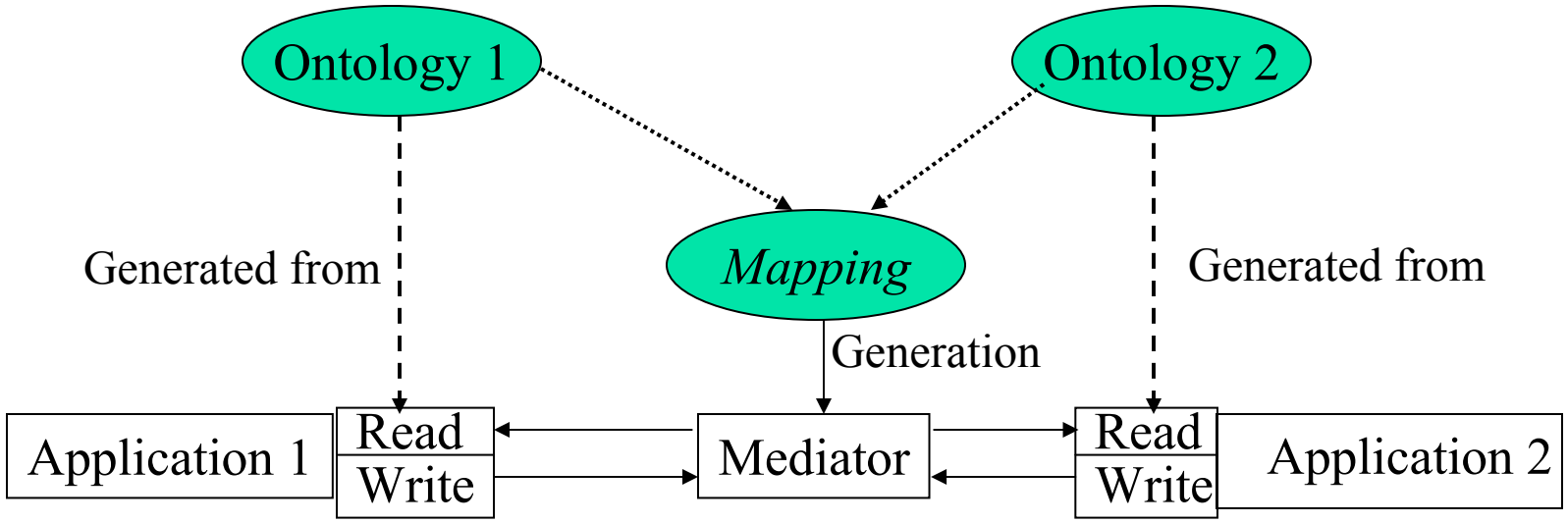
Ontology: Usage Examples

2. Data access through a shared ontology: *An alternative to translators*



Ontology: Usage Examples

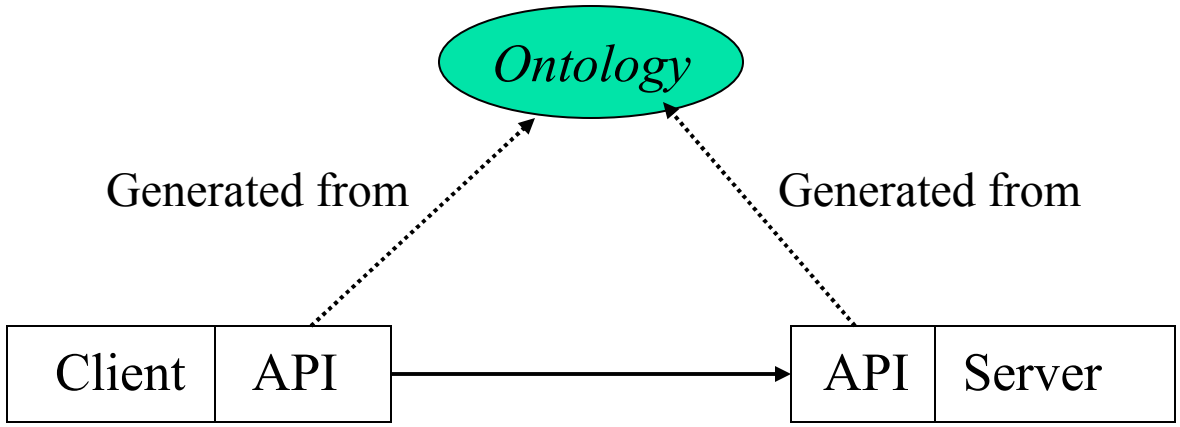
3. Data Access thanks to correspondances (*mappings*) between ontologies



Ontology: Usage Examples

4. Sharing of services

- Example:
  - Ontology in IDL or UML
  - Used to generate the code of the clients/server interfaces



Ontology: Conclusion

- Approach for Semantic Interoperability
- Mainly 'around the Web'
- Similarities to the structural integration and interoperability
  - Integration, Merging, correspondance between ontologies
  - Heterogeneous Ontologies, …
- Other extensions: Semantic Annotations

Semantic Annotations

- ***Annotate***: to add a brief explanation or opinion to a text or drawing (*Cambridge advanced learner's dictionary, http://dictionary.cambridge.org/)*

- ***Annotation***: A comment attached to a particular section of a document. Many computer applications enable you to enter annotations on text documents, spreadsheets, presentations, and other objects. This is a particularly effective way to use computers in a workgroup environment to edit and review work… (*Webopmedia, http://www.webopedia.com*).

Semantics (Types) of Annotations (examples)

1. *Decoration*: annotations = comments associated with the resource

2. *Linking*: annotations = links

3. *Instance Identification*: the annotated object (U#X) is an instance of a given class and the annotation content may be a link to that class;

Semantics (Types) of Annotations (examples)

4. *Aboutness*: no assertion is made about the existence of an instance of the concept C, but there is a loose association with the concept;

5. *Pertinence*: the target of the annotation may be of interest to the annotated object.

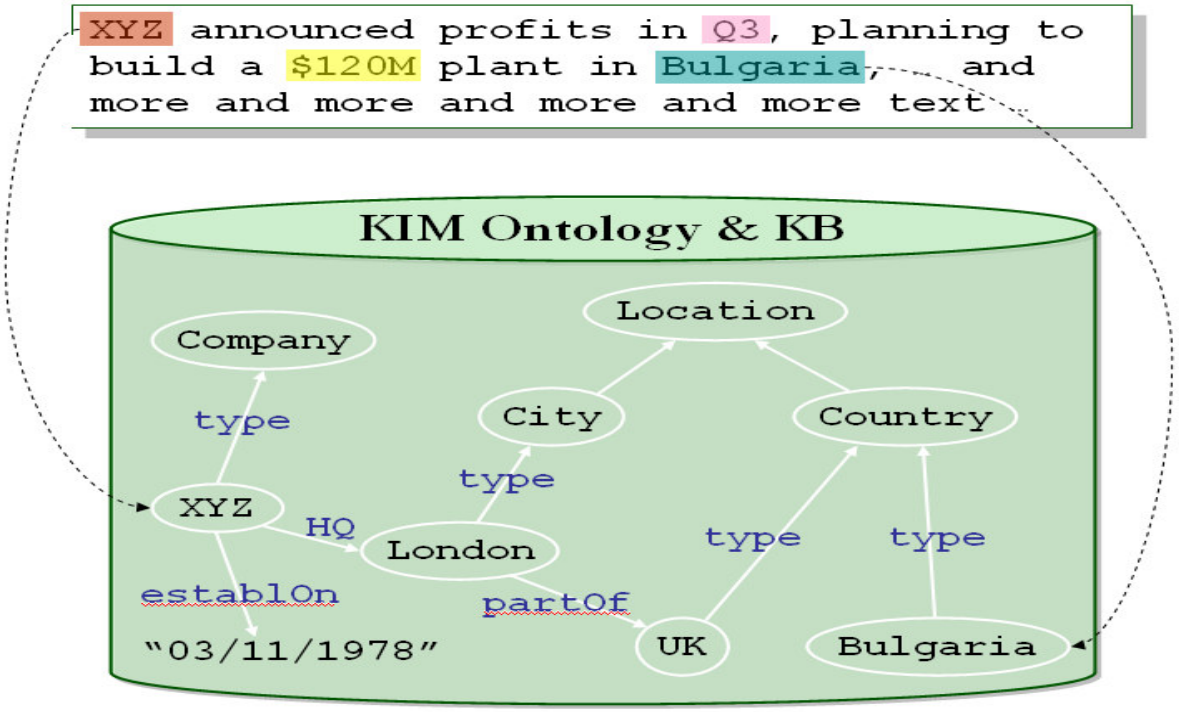
- Types of annotations to classify query results

Annotation Services

- Progressive move *from manual to automatic or semi-automatic* annotation provision
- Services to provide ("write") annotations?
- Content of some types of annotations relies on given ontologies ⇒ ontology services (like querying or browsing an ontology) have to be coupled with annotation services

Knowledge and Information Management (KIM)

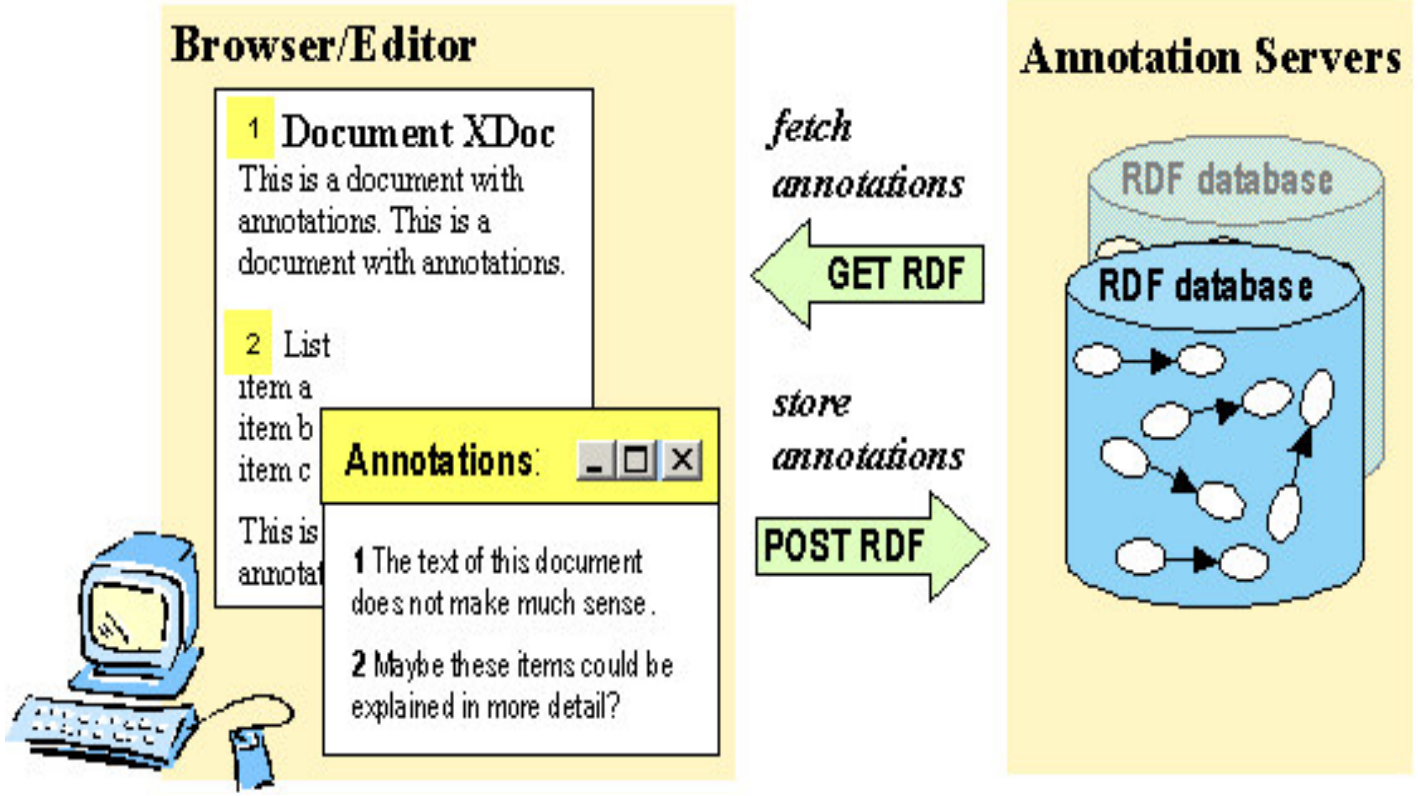
- Infrastructure and services for automatic semantic annotation, indexing, and retrieval of unstructured and semi-structured content



Annotea

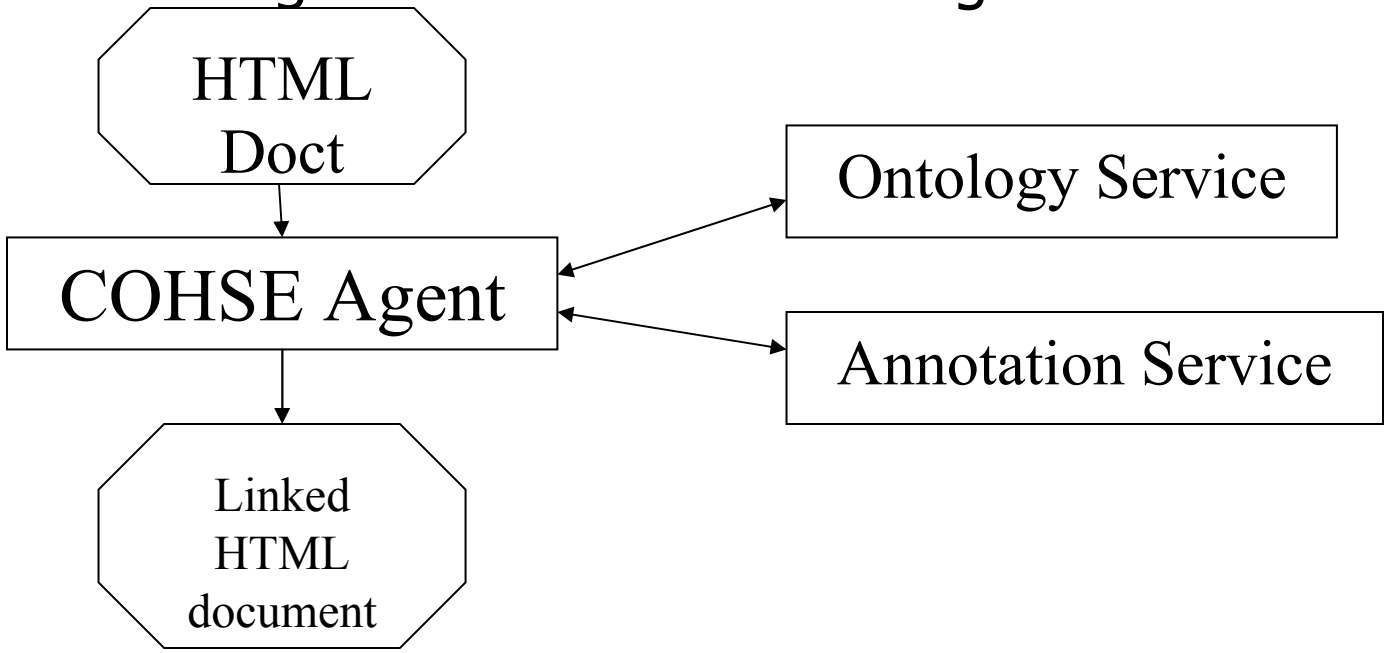
- Collaboration via shared metadata based Web annotations, bookmarks, and their combinations using
  - RDF-based annotation schema for describing annotations as metadata
  - XPointer for locating the annotations in the annotated document
- Annotea schema includes *properties* which links the annotation to the annotated resource:
  - *Annotates:* refers to the enclosing URI,
  - *context*:contains the location using Xpointer,
  - *Body:* link to the body of the annotation itself.

Annotea Architecture



COHSE

- *Conceptual Open Hypermedia SErvice*
- *Terms and words within a document* used as *entry points* to an ontology
- *Relevant concepts in the ontology* can be used to determine targets for links out of the given document



Semantic Interoperability (end)

- Semantic annotation
  - Mainly for Web Documents and Web Services
  - Concepts and Tools available
- What about Enterprise Models and Systems?
  (cf. ReX INTEROP, http://www.interop-noe.org)
- To what extent the current advances/approaches might be re-used for annotating enterprise models?

Presentation Outline

II. Ontology and Semantic Interoperability

Questions?

Presentation Outline

I. Integration and Interoperability
   1. Problem
   2. Approaches (« process » vs « objets »)

II. Ontology and Semantic Interoperability

III. Enterprise Modeling and Semantic Interoperability

INTEROP NoE

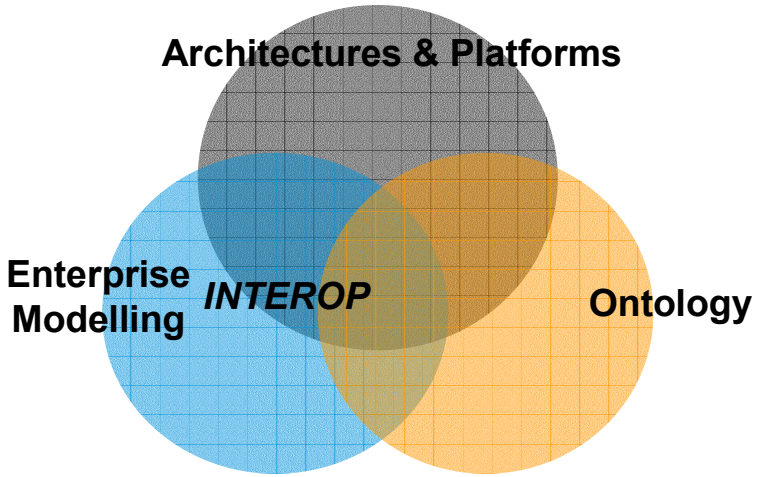
- Partners :
  - ≈ 50 partners (laboratories, enterprises and research centers)
  - ≈ 200 researchers and 120 PhD
  - 13 UE member states + Norway and Helvetic Confederation (Switzerland)

- Coordination : University Bordeaux 1 LAPS/GRAI
- Start : 1st November 2003
- Duration : 36 months (Probably extended to 42)

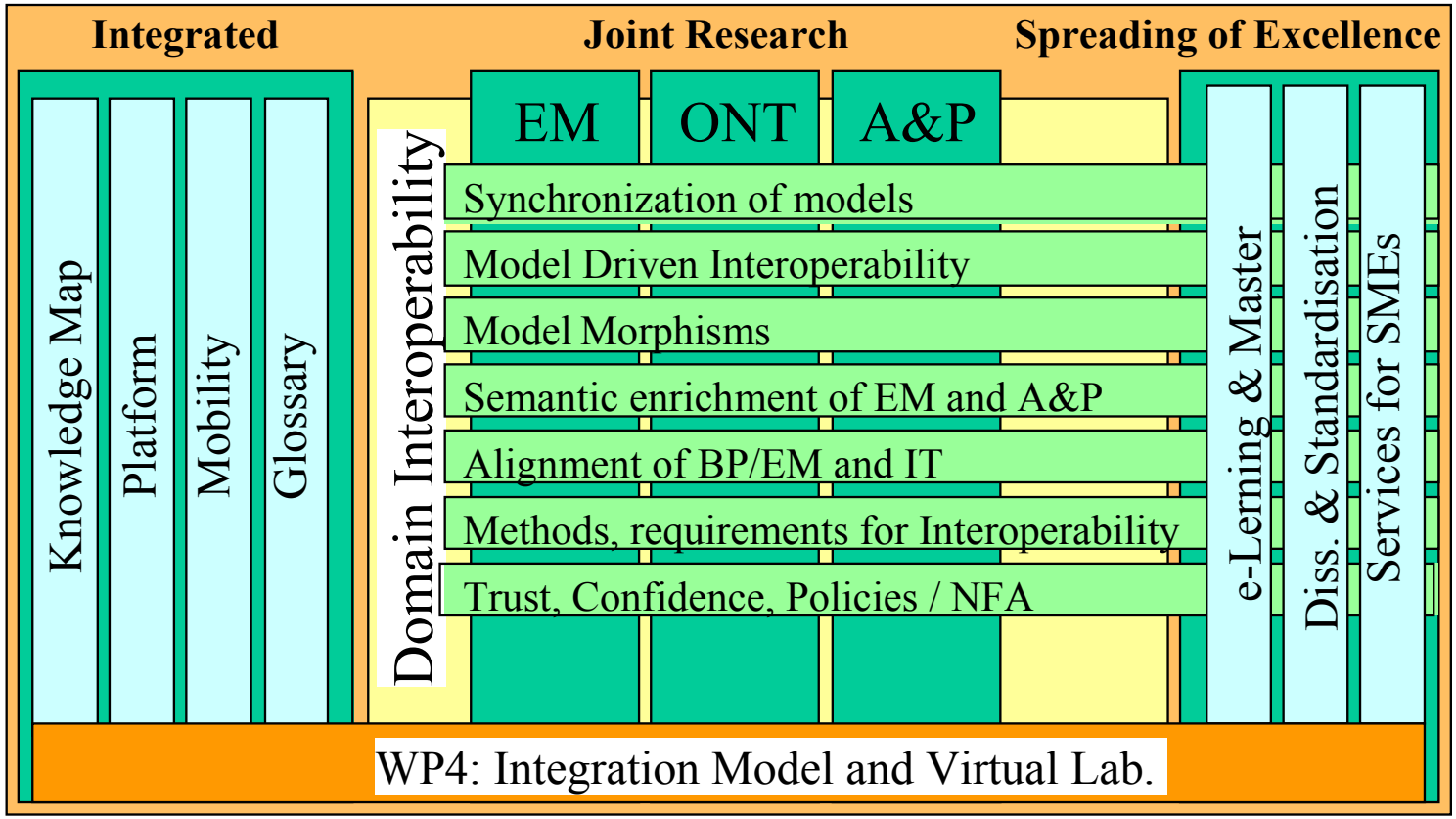
interop-noe.org

- Multidisciplinary approach for the Interoperability of Enterprise Applications and Software:
  - Architecture & Platforms: to provide implementation frameworks,
  - Enterprise Modelling: to define Interoperability requirements and to support solution implementation,
  - Ontology: to identify Interoperability semantics in the enterprise.

Knowledge integration for Interoperability research



INTEROP NoE Objectives & Work Program



What next?

- Enterprise Modelling
- Enterprise Ontologies
- First steps in Enterprise Models Annotation

Enterprise Models: Why

- Understand, Analyze, Simulate, …
- Diagnosis (disfunctioning: material, flows, organization, etc.)
- Re-structuring
- Integration/Interoperation
- Evolution
- Conformity to standard (example: Quality)
- …
- Define a common understanding

Enterprise Models: What is it?

- Describe the « things » of an enterprise
  - Functions
  - Behaviour
  - Information
  - Resources
  - Organization
- In order to
  - Understand
  - Evaluate
  - Optimize
  - Control
  - Simulate
  - etc.

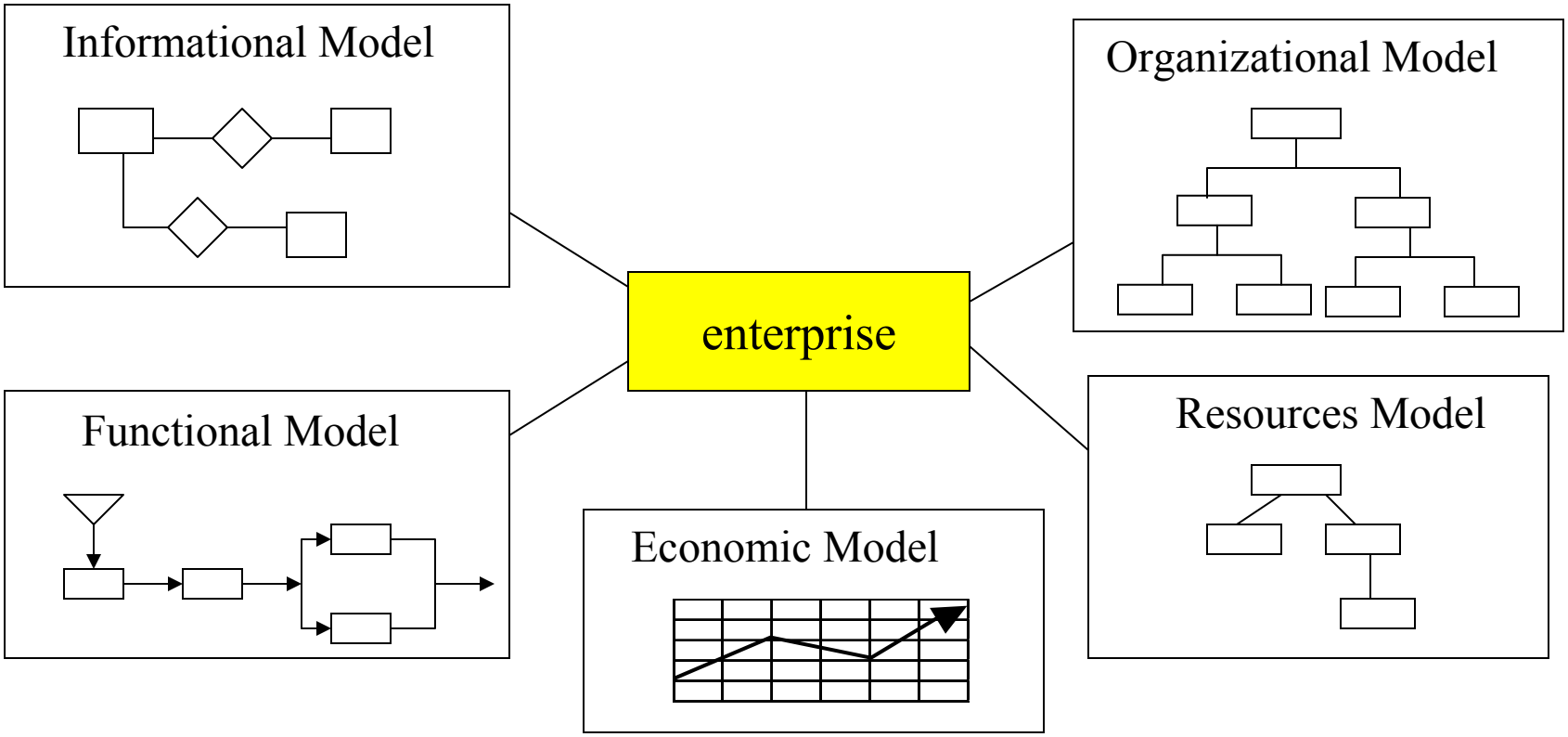
Enterprise Models: What is it?

- Representation (artefact)

- Multiple points of views
  - Operational,
  - Decisional,
  - Informational,
  - Strategic,
  - Economic
  - etc.

Enterprise Models

- Model or Models (F. Vernadat, CAiSE/EMOI, 2004)



Enterprise Models

- Diversity of the vocabulary

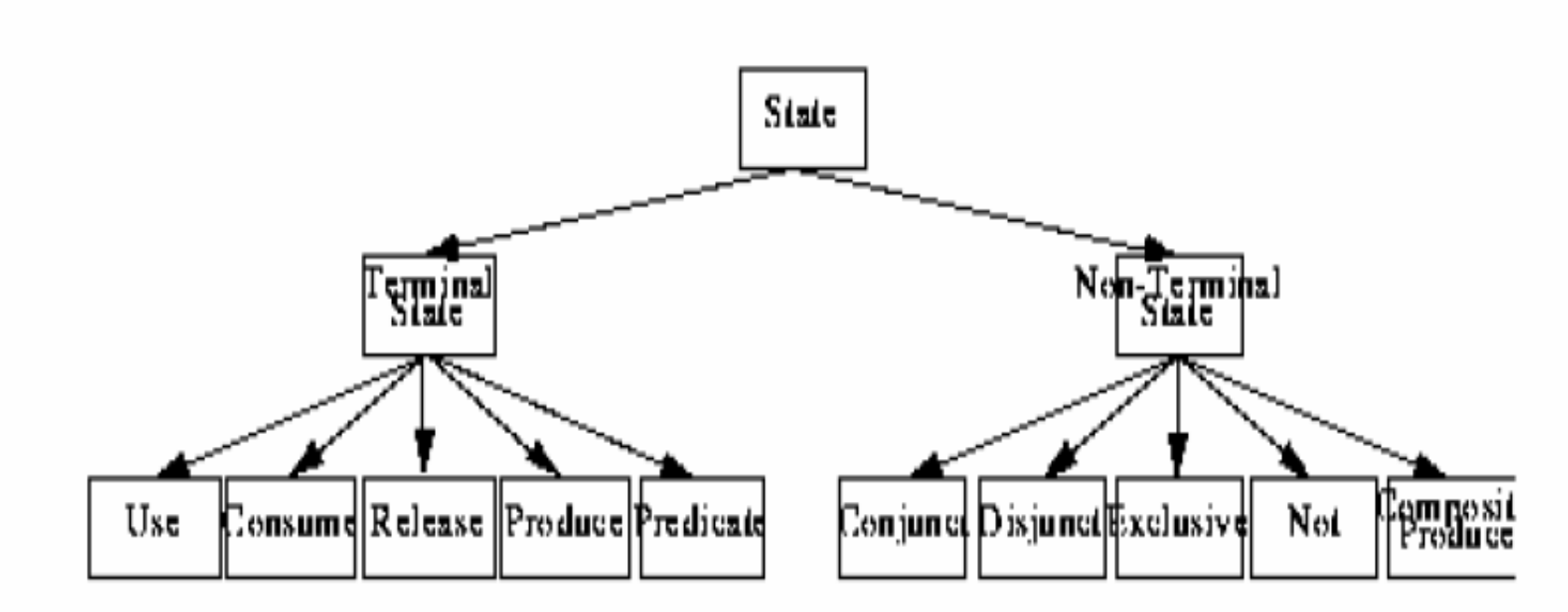
- Diversity of notations
  - Entity-Relationship
  - Objects (example: UML)
  - Petri net
  - Transition Diagrams
  - Logic(s)
  - etc.

Enterprise Models: Milestones

- 70's (notations of model) :
  - SADT, Entity-Relationship, Semantic Network
- 80's:
  - Computer Integrated Manufacturing Methods: ICAM/IDEF, GRAI, etc.
- End of 80's: CIM Open System Architecture
- 90's:
  - ERP Deployment
  - Workflow systems
  - Object Orientation (IEM, UML, etc.)
  - Ontologies (IDEF5, TOVE, Enterpise Ontology, PSL, etc.)
- 2000's?: Semantic Interoperability ?

EO: Toronto Virtual Enterprise (TOVE)

1. Time and actions
2. States
3. Resources
4. Others: product, organization, cost management



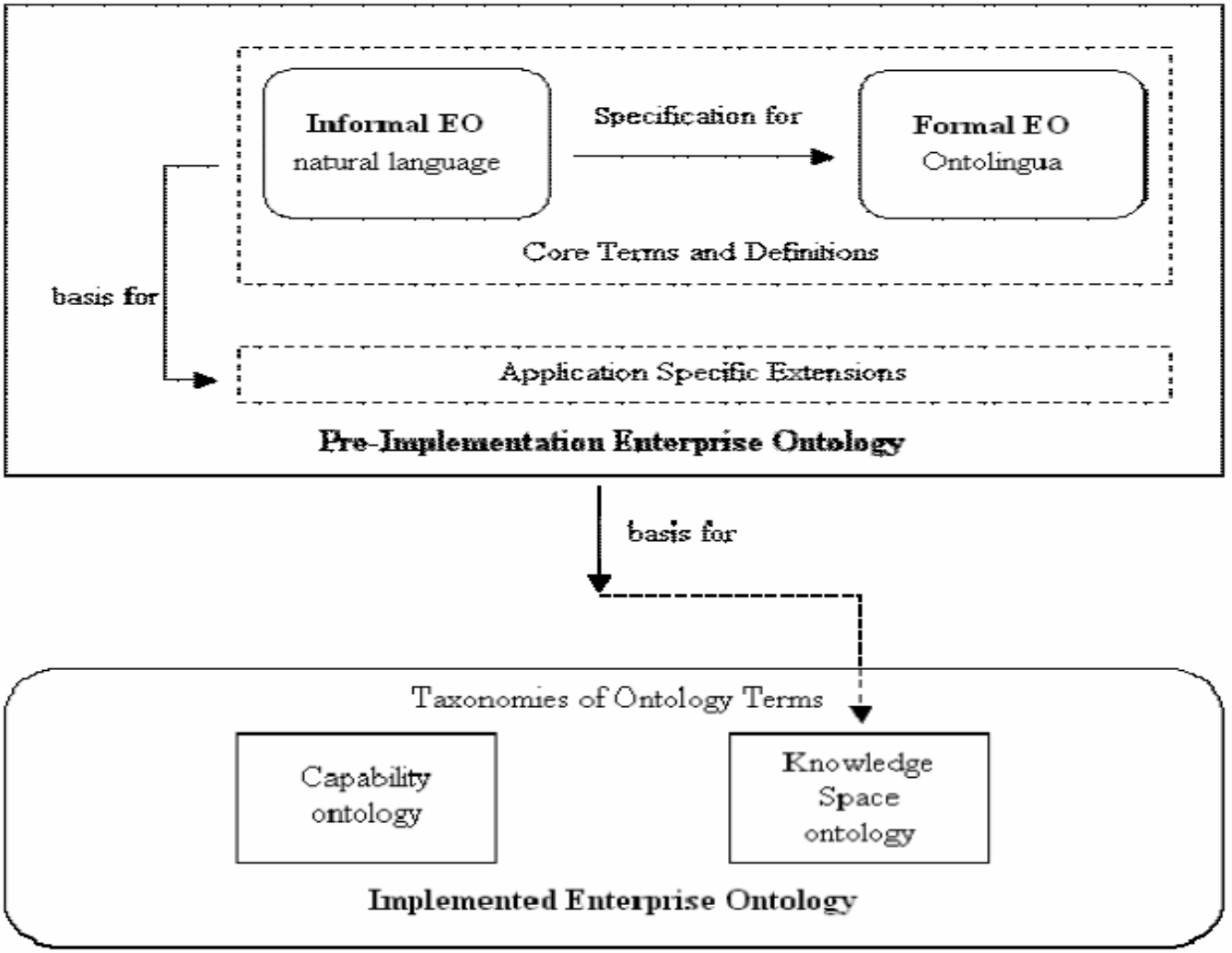
EO: Edinburgh Enterprise ontology

- 92 Classes
- 68 Relations
- 483 axioms
- 10 individuals

Enterprise ontology

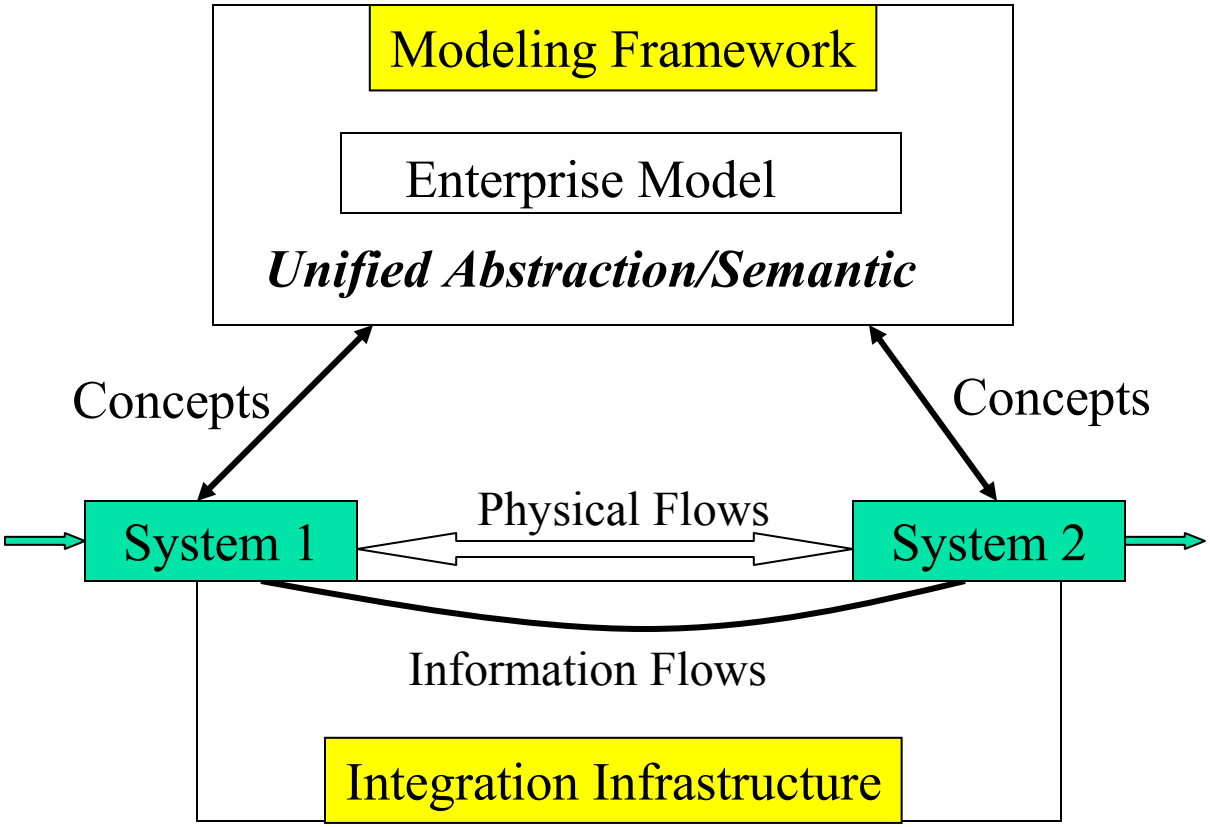
ACTIVITY, etc.	*ORGANISATION*	*STRATEGY*	*II.3.4.6 MARKETING*	*II.3.4.7 TIME*
Activity	Person	Purpose	Sale	Time Line
Activity Specification	Machine	Hold Purpose	Potential Sale	Time Interval
Execute	Corporation	Intended Purpose	For Sale	Time Point
Executed Activity Specification	Partnership	Purpose-Holder	Sale Offer	
T-Begin	Partner	Strategic Purpose	Vendor	
T-End	Legal Entity	Objective	Actual Customer	
Pre-Condition	Organizational Unit	Vision	Potential Customer	
Effect	Manage	Mission	Customer	
Doer	Delegate	Goal	Reseller	
Sub-Activity	Management Link	Help Achieve	Product	
Authority	Legal Ownership	Strategy	Asking Price	
Activity Owner	Non-Legal Ownership	Strategic Planning	Sale Price	
Event	Ownership	Strategic Action	Market	

Enterprise ontology



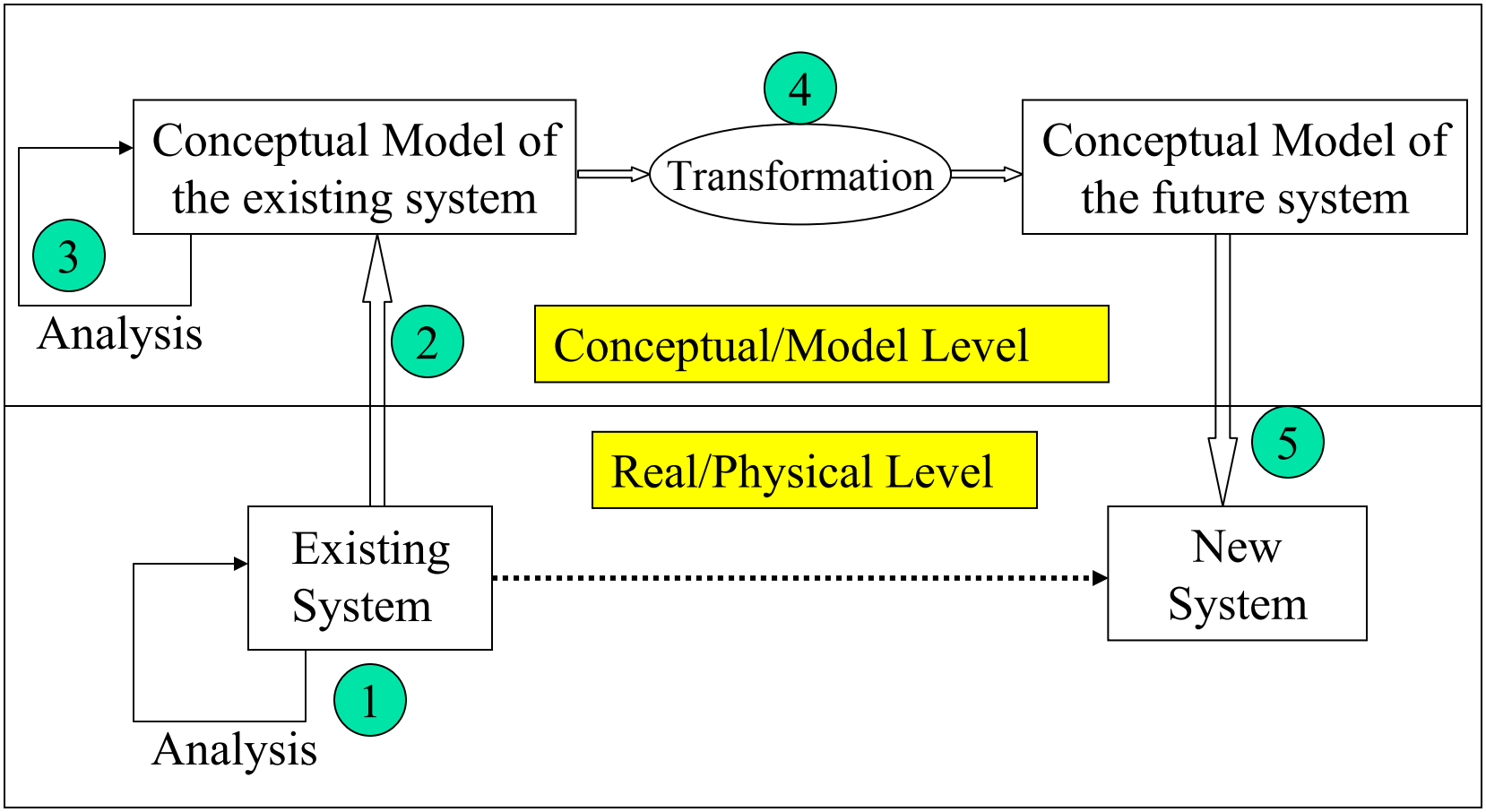
Enterprise Models

- How?

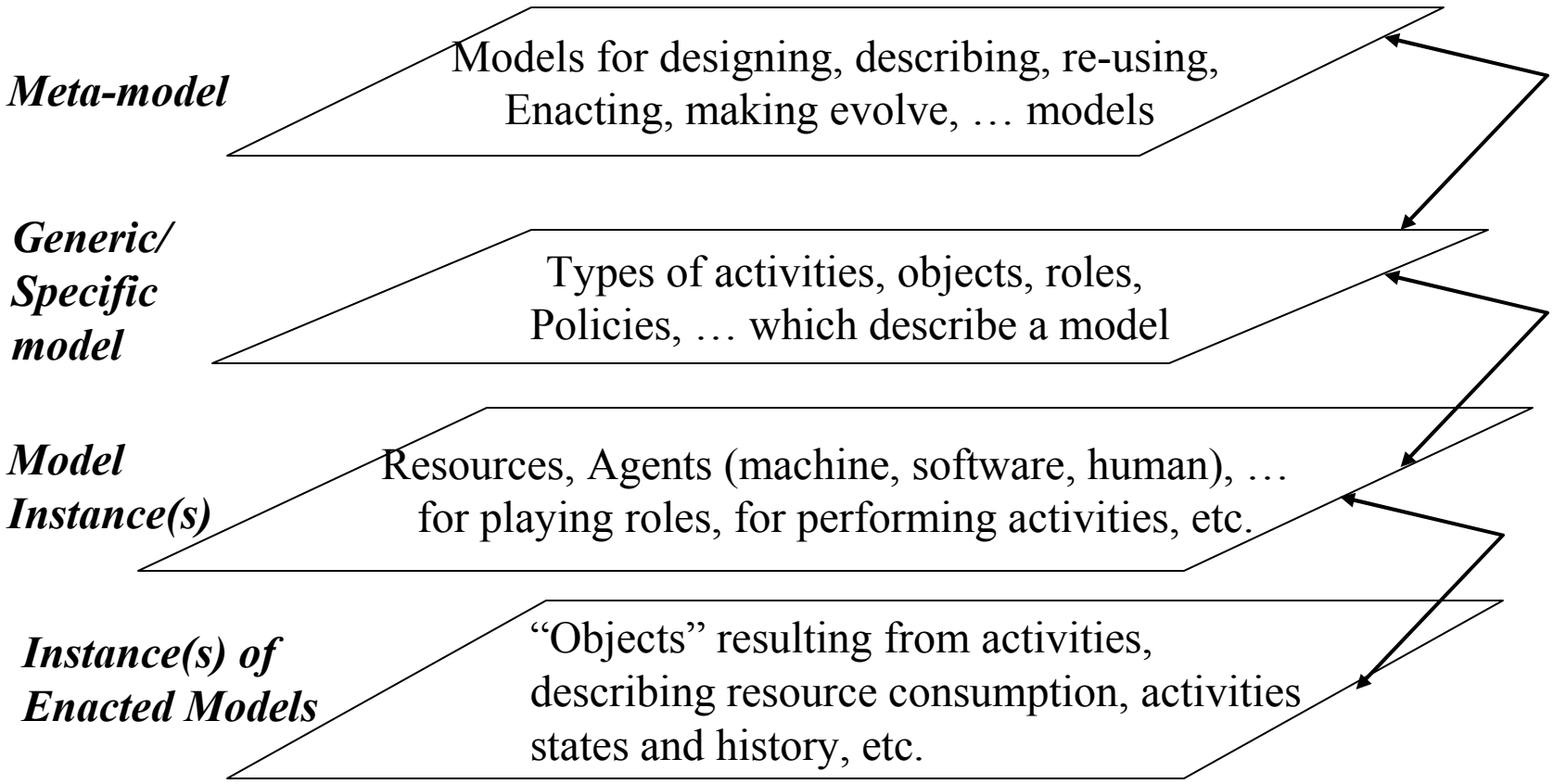


Enterprise Models

Modeling Process



Models, Instances and Annotations



⇒ Intra-levels and inter-levels semantic enrichment

- « Co-existence » of concepts and « supports »:
  - For (meta-)modeling
  - For instanciating
  - For functioning (activation/enactment)
    - Integrated software systems (example: ERP)
    - Proprietary software
    - « House made » Software
    - etc.
- Cooperation and communication Purposes ⇒ interoperability
  - Intra-level
  - Inter-levels

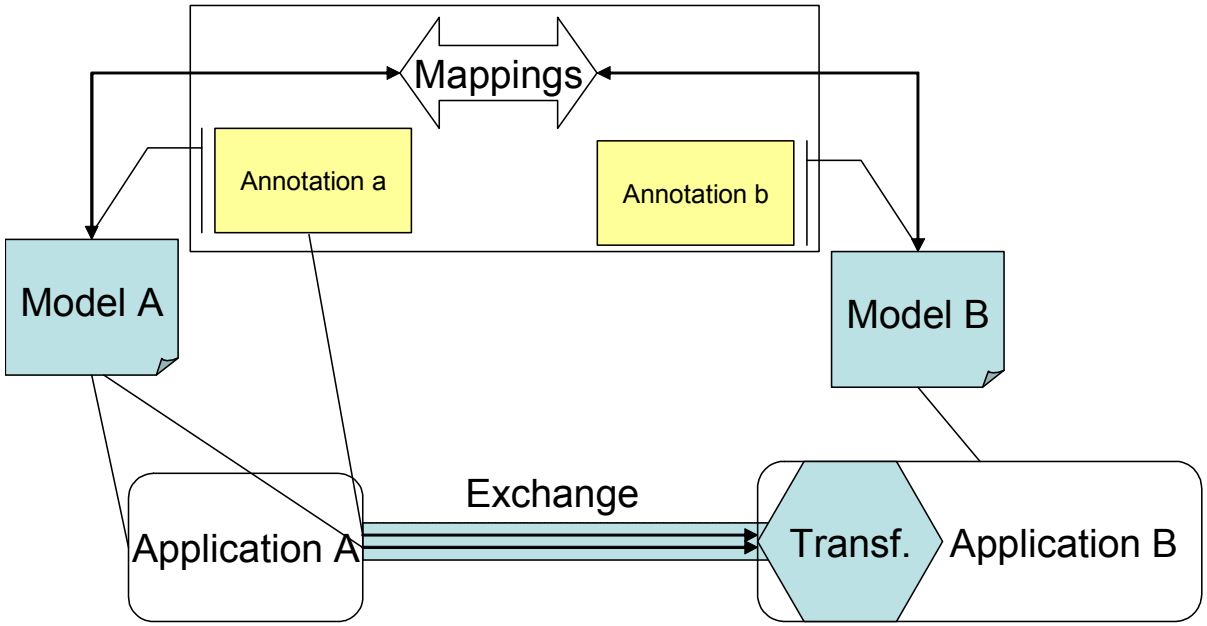
Semantic Annotations of Enterprise Models

- Types, forms, … of Semantic Annotations of Enterprise Models?

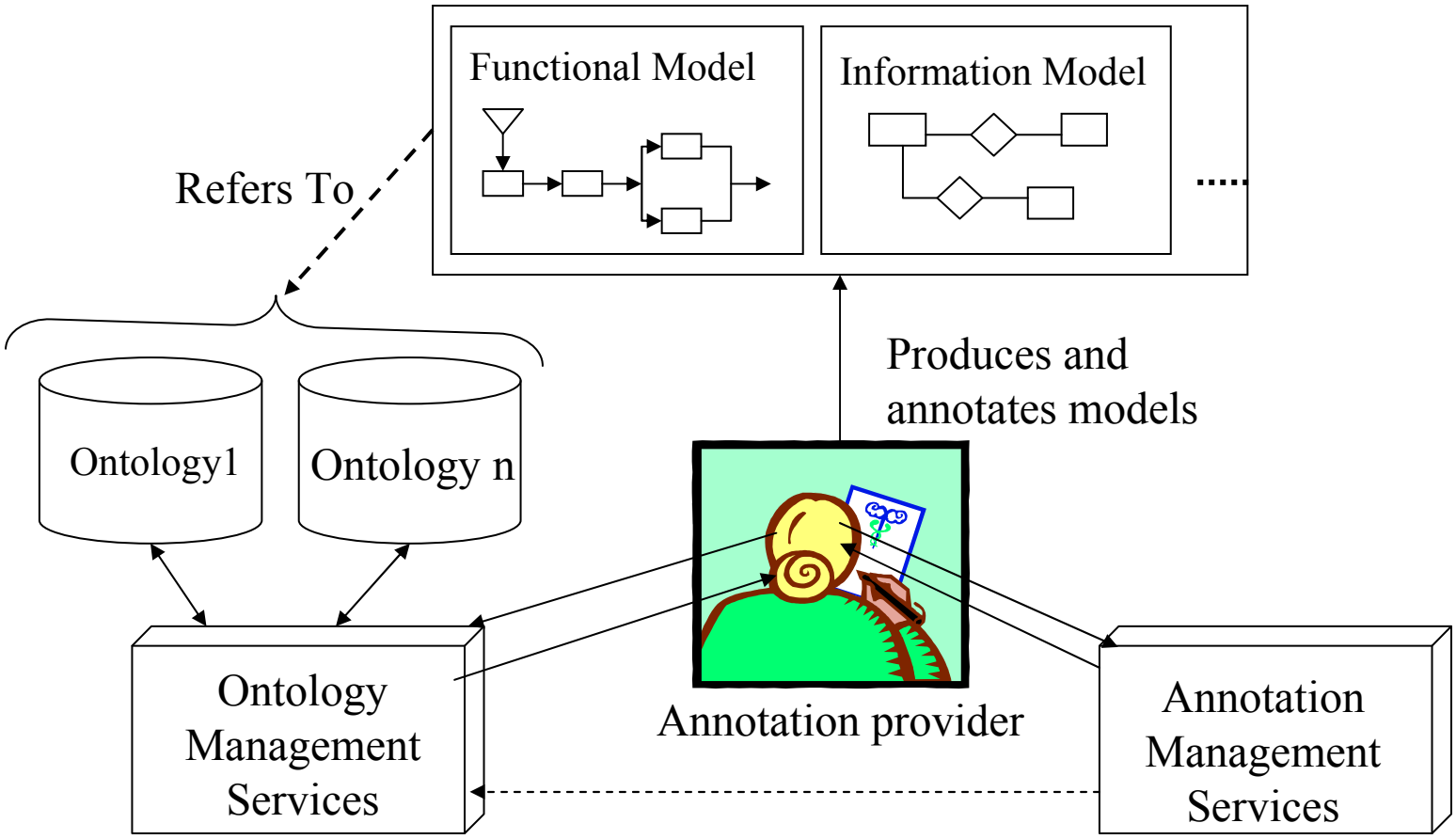
- Tentative Identification thanks to
  1. Literature (poor concerning enrichment of models)
  2. Case studies:
     - Various Perspectives of Models
  3. Common framework (Enterprise ontology, Enterprise Modeling and Ontology Management tools)

Purpose of Semantic Annotations of Models

- Model Exchange



What Types of Annotations and Services for EM?



- Types of Annotations
  - Homogeneous environment
  - Heterogeneous environment (tbd)
- Types of Services:
  - Matching to the meta-model
  - Matching to the ontology
  - Annotation Checking (inconsistencies)
  - Annotation storage and re-use
  - Model exploration, querying
  - …
  - To be cont'd

Annotation Content

<Annotation

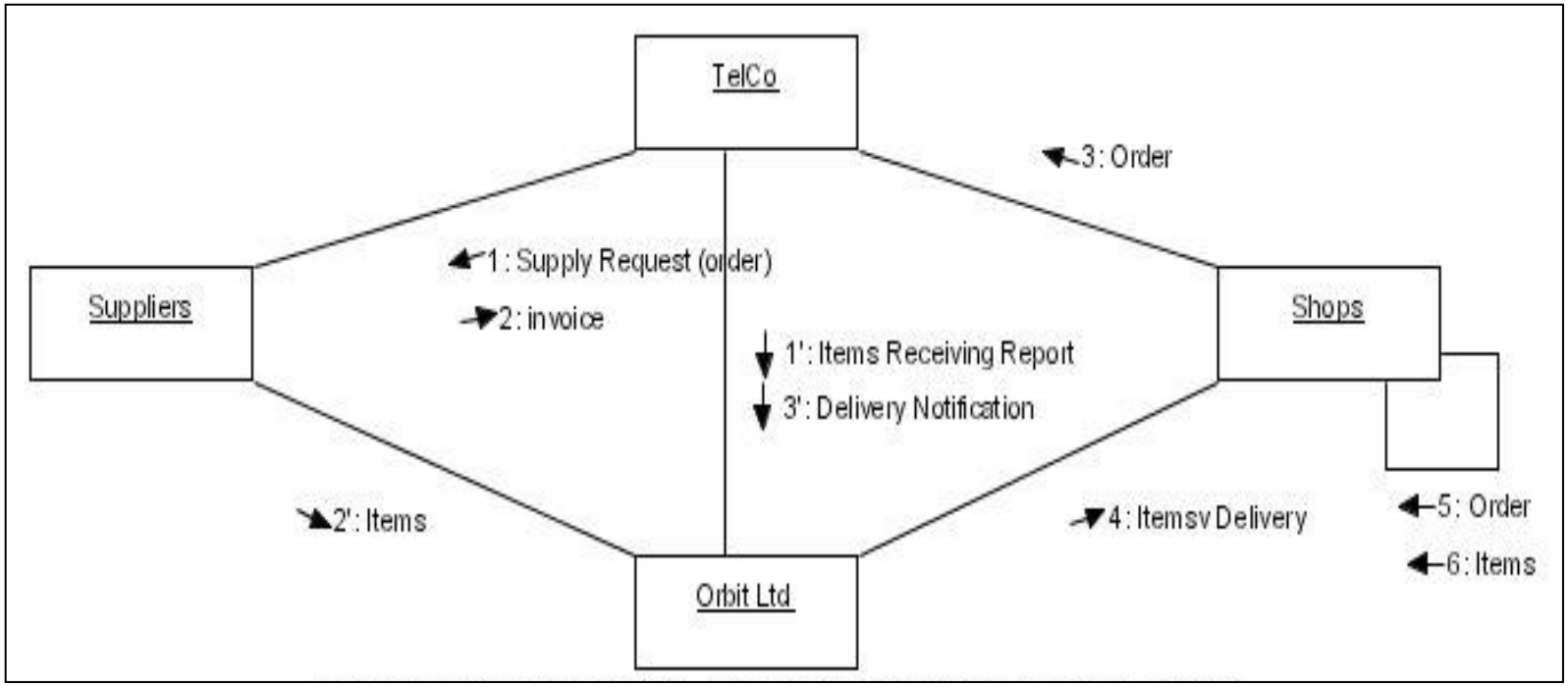
Type = (example: Decoration, Linking, Instance Identification, Aboutness, Pertinence)

Ref2Ontology = (reference (uri) to the ontology concept related to the current model concept)

Constraints = (may be written using OCL, with references to the ontology, if this one is represented by a UML class diagram)

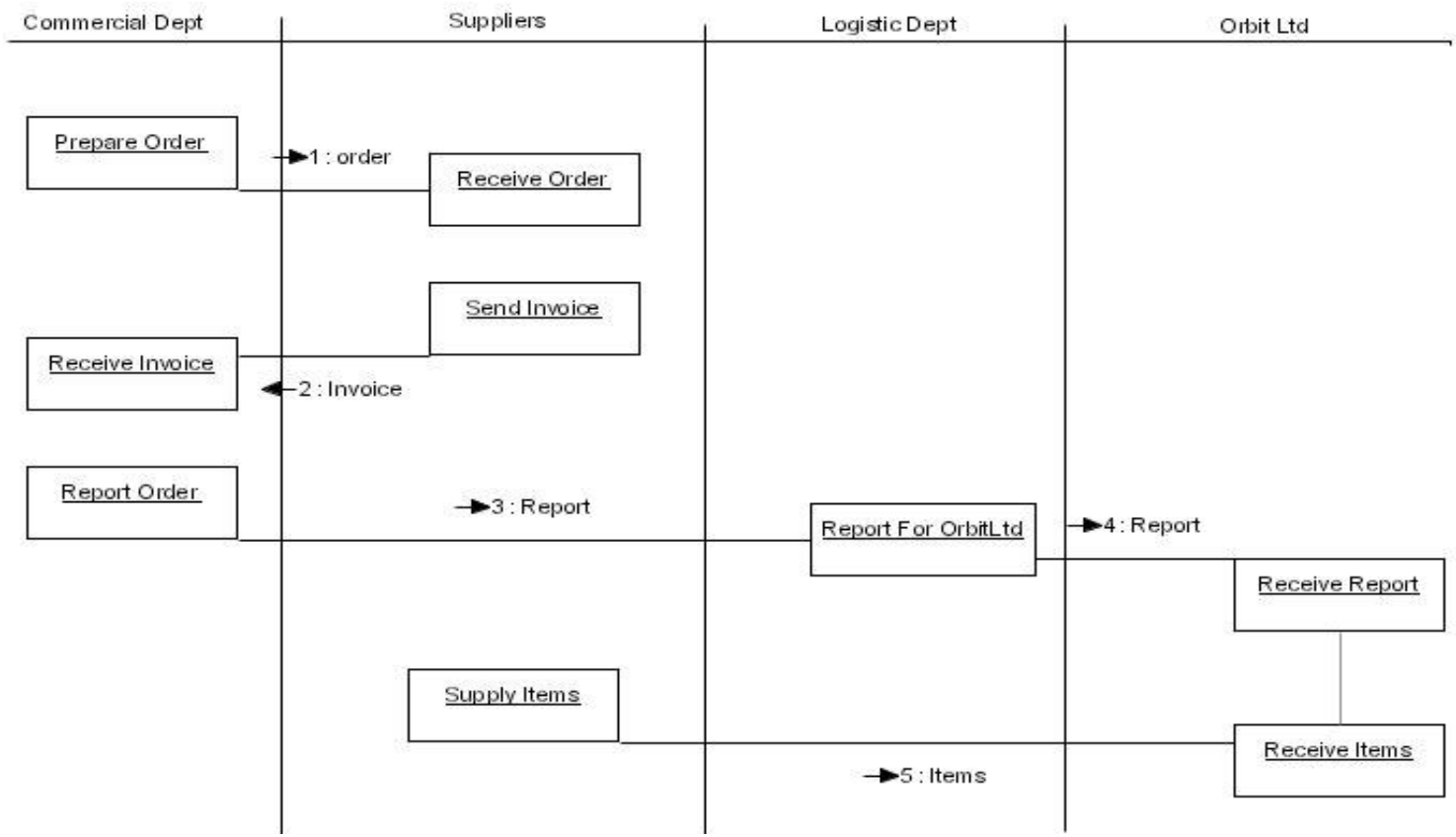
/>

Annotation Content: Example



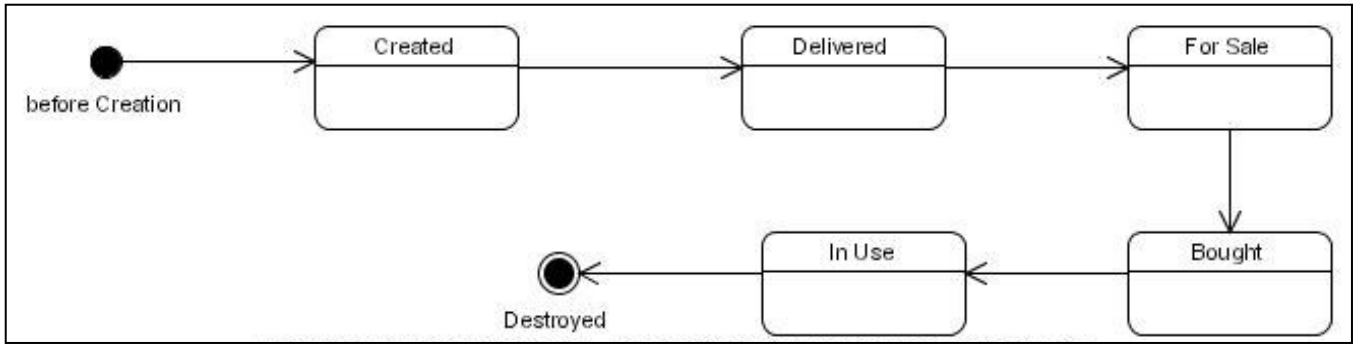
Items Receiving and Delivery Activity Diagram (in the studied enterprise)

Annotation Content: Example



Activity diagram of the process "Ordering & Receiving Items"

Annotation Content: Example



Items Receiving and Delivery Activity Diagram (in an Ontology)

<Annotation

Type = 'Instance identification'

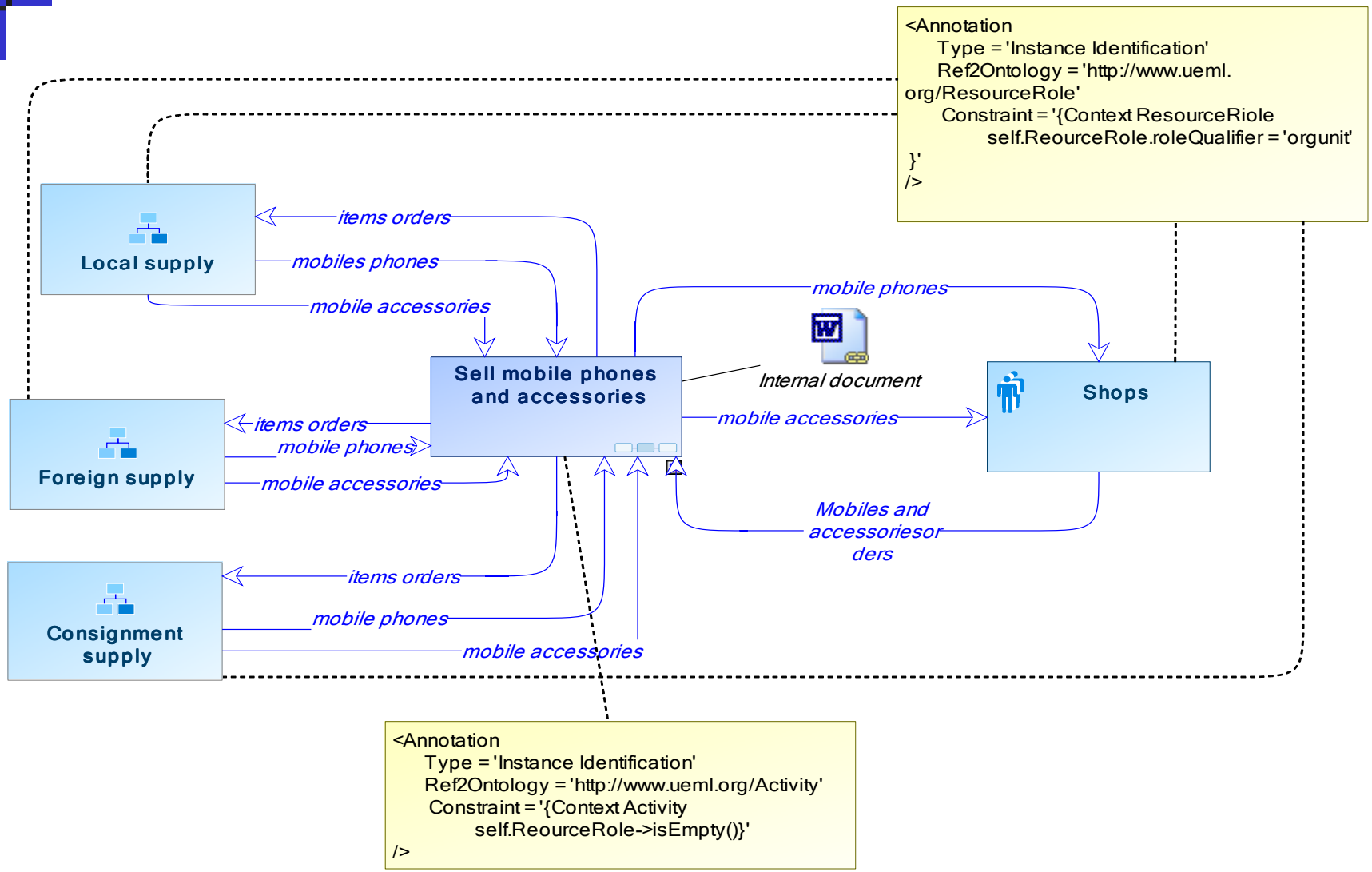
Ref2Ontology = 'http://www.ueml.org/Activity'

Constraint = '{Context Activity

Self. ResourceRole->isEmpty()}'

/>

Annotation Content: Example



Annotation Content: Constraints

- Equivalence
- Part-of
- Ad hoc

Concluding Remarks

- EM annotation:
  - "Analytical" identification of annotation types
  - Expressed with reference to:
    - The EM tool
    - The provided ontology
  - No theory behind
- Open questions:
  1. Language(s) for semantic annotation of models?
  2. Annotation "content" (purpose, link to ontology, type of matching, link from the artefact to the modelled real world, …)?

Concluding Remarks

3. Adequacy/Completeness of existing services (Annotation provision and management)?
4. Impact on existing tools/environments?
   - EM Exported in XML
   - Explored and Annotated
   - Later on:
     - offer "direct" annotation of models
     - Coupling/Integration of EM tools and
       - Ontology management services
       - Annotation management services

- *= Focus* of the *2nd ENEI Intern'l workshop*, Sept. 2006, Vienna (@BPM Conference)

Concluding Remarks

Thank you for paying attention

Questions?

Elements of bibliography

- M. Uschold, R. Jasper. A Framework for Understanding and Classifying Ontoloy Applications. *Procedings of the IJCAI'99 Workshop on Ontologies and Problem Solving Methods,* Stockholm, Sweden, August 2, 1999.
- ISO. The EXPRESS Language Reference Manual. 1994. ISO 1033-11:1994(E).
- M. Uschold and M. Gruninger. Ontologies: Principles Methods and Applications. *Knowledge Engineering Review*, 11(2), 1996.
- K. Sivashanmugam, K. Verma A. Sheth and J. Miller, Adding Semantic to Web Services Standards, *1rst Intern'l Conference on Web Services (ICWS'03),* Las Vegas, Nevada, June 2003, pp. 395-401
- A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic Annotation, Indexing, and Retrieval. *Elsevier's Journal of Web Semantics*, Vol. 2, Issue (1), 2005.
- N. Boudjlida and D. Cheng, Complement Concept and Capability Discovery, *in proceedings of the CaiSE (Conference on Advanced Information Systems Engineering) EMOI'O4 (Enterprise Modelling and Ontologies for Interoperability) Open Workshop*, Vol.3, p.337-342, J. Grundspenkis and M. Kirikova Editors, Riga, Latvia, 7-11 June 2004.

Elements of bibliography

- S. Abiteboul, P. Buneman, and D. Suciu. Data on the Web; From Relations to Semi-Structured Data and XML. *Morgan Kaufmann Publishers*, San Francisco, CA, 2000.
- N. Boudjlida and O. Perrin. A Formal Framework and a Procedural Approach for Data Integration. *In Proceedings of the International Conference on Systems Integration, ICSI'94,* pages 476–485, Sao Paulo, Brazil, August 1994. IEEE Comp. Society Press.
- L. Cardelli and P. Wegner. On Understanding Types, Data Abstraction and Polymorphism. *ACM Computing Surveys*, 17(4):471–522, December 1985.
- M.S. Fox, M. Barbuceanu, and M. Gruninger. An organization ontology for enterprise modeling: Preliminary concepts for linking structure and behaviour. *Computers in Industry*, 29(1-2):123–134, July 1996
- INTEROP. Interoperability Research for Networked Enterprises Applications and Software. NoE IST-508 011, Technical Annex, November 2003. http://www.interop-noe.org
- D. Konstantas. Object Oriented Interoperability. *In ECOOP'93, 7th European Conference on Object Oriented Programming,* Lecture Notes in Computer Science 707, pages 80–102, Kaiserslautern, Germany, July 1993.

Elements of bibliography

- S.J. Mellor, S. Kendall, A. Uhl, and D. Weise, editors. Model Driven Architecture. *Addison Wesley Publishing Company*, March 2004. ISBN 0201788918.
- D.E. Perry and G.E. Kaiser. Models of Software Development Environments. *IEEE Trans. on Software Engineering,* 17(3):283–295, March 1991.
- F. Vernadat. Enterprise Modeling: Objectives, Constructs & Ontologies, Tutorial at the 1st EMOI workshop co-located with CAiSE 2004, Riga, Latvia, June 2004.
- A.I. Wasserman. Tool Integration in Software Engineering Environments. *In Software Engineering Environments: International Workshop on Environments,* Berlin, 1990. Springer-Verlag.
- G. Wiederhold. Mediators in the Architecture of Future Information Systems. *IEEE Computer*, pages 38–49, 1992.
- J.A. Zachman. Framework for Information Systems Architecture. *IBM Systems Journal*, 26(3):276–292, 1987.
- W. Inmon J.A. Zachman and J.G. Geiger. *Data stores data warehousing and the Zachman framework, Managing enterprise knowledge*. McGraw-Hill, 1997.
- W3C. http://www.w3c.org
- OMG. http://www.omg.org

Elements of bibliography

- METEOR-S: Semantic Web Services and Processes, http://swp.semanticweb.org
- M. Vargas-Vera, E. Motta, J. Domingue, M. Lanzoni, A. Stutt and F. Ciravegna. *"MnM: Ontology Driven Semi-Automatic and Automatic Support for Semantic Markup",* The 13th International Conference on Knowledge Engineering and Management (EKAW 2002), ed. Gomez-Perez, A., Springer Verlag, 2002.
- M-R. Koivunen, R. Swick, E. Prud'hommeaux, Annotea Shared Bookmarks. *In Proc. of the KCAP 2003 workshop on knowledge markup & semantic annotation.*
- S. Bechhofer, L. Carr, C.A. Goble, S. Kampa and T. Miles-Board, *The semantics of Semantic Annotation.* In CoopIS/DOA/ODBASE'2002, First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems, Irvine, California. Springer-Verlag LNCS Vol. 2519, pp. 1151--1167. 2002.
- PSL, ISO 18629, 2004, http://www.iso.ch
- M. Missikoff & al. Ontology-Based Integration and Interoperability of Enterprise Modelling, Architectures and Platforms: State of the Art and State of the Practice. *INTEROP Network of Excellence, Deliverable D8.1, November 2004.* http://www.interop-noe/ (Public Deliverables).