Verso il Semantic Web

La tappa italiana del W3C Semantic Tour

Il Web Semantico nelle applicazioni d'impresa

Interoperabilità e ontologie d'impresa

M. Missikoff, F. Schiappelli, F. Taglino
LEKS, IASI-CNR

10 Giugno 2003

Summary

- The Semantic Web and **Interoperability**
- Semantic Interoperability: an ontology approach for **enterprise cooperation**
- A Case Study: the Harmonise solution
- Methodology for enterprise ontology representation: **OPAL**
- **Semantic annotation** for information elements
- Hindrance to Interoperability: identification and Resolution of **clashes**

What is Semantic Web?

A vision of possibilities

"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling **computers** and people to work in **cooperation**."

(Tim Berners-Lee, James Hendler and Ora Lassila, The Semantic Web, Scientific American, May 2001)

The Semantic web for interoperability

- **Internet** guarantees reliable data packets exchange;
- Applications interoperability mainly achieved by using standards (official and “de facto”)

- The **Semantic web** allows application software interoperability, with a minimal impact
- Application software keeps its **own** organisation of data and processes
- The Semantic Web **bridges the differences** that unanticipated cooperation will present

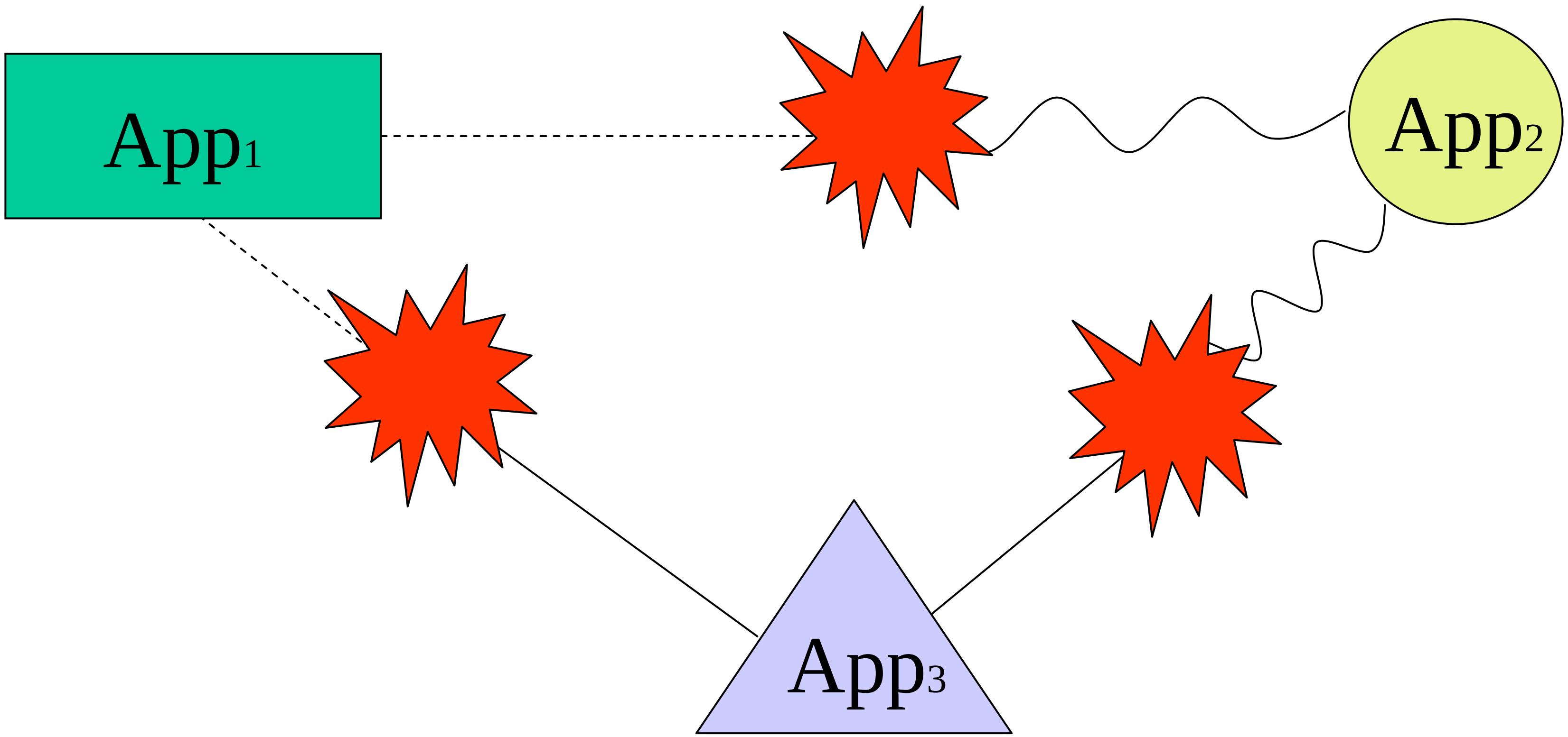
Different levels of interoperability problems

- **Data level**
  - Different representation formats (e.g., RDB, XML, …)
  - Different data organization (e.g., schema structures)
  - Different naming policy
- **Functional level**
  - Data interoperability is required
  - Different functions
  - Different Business Process models

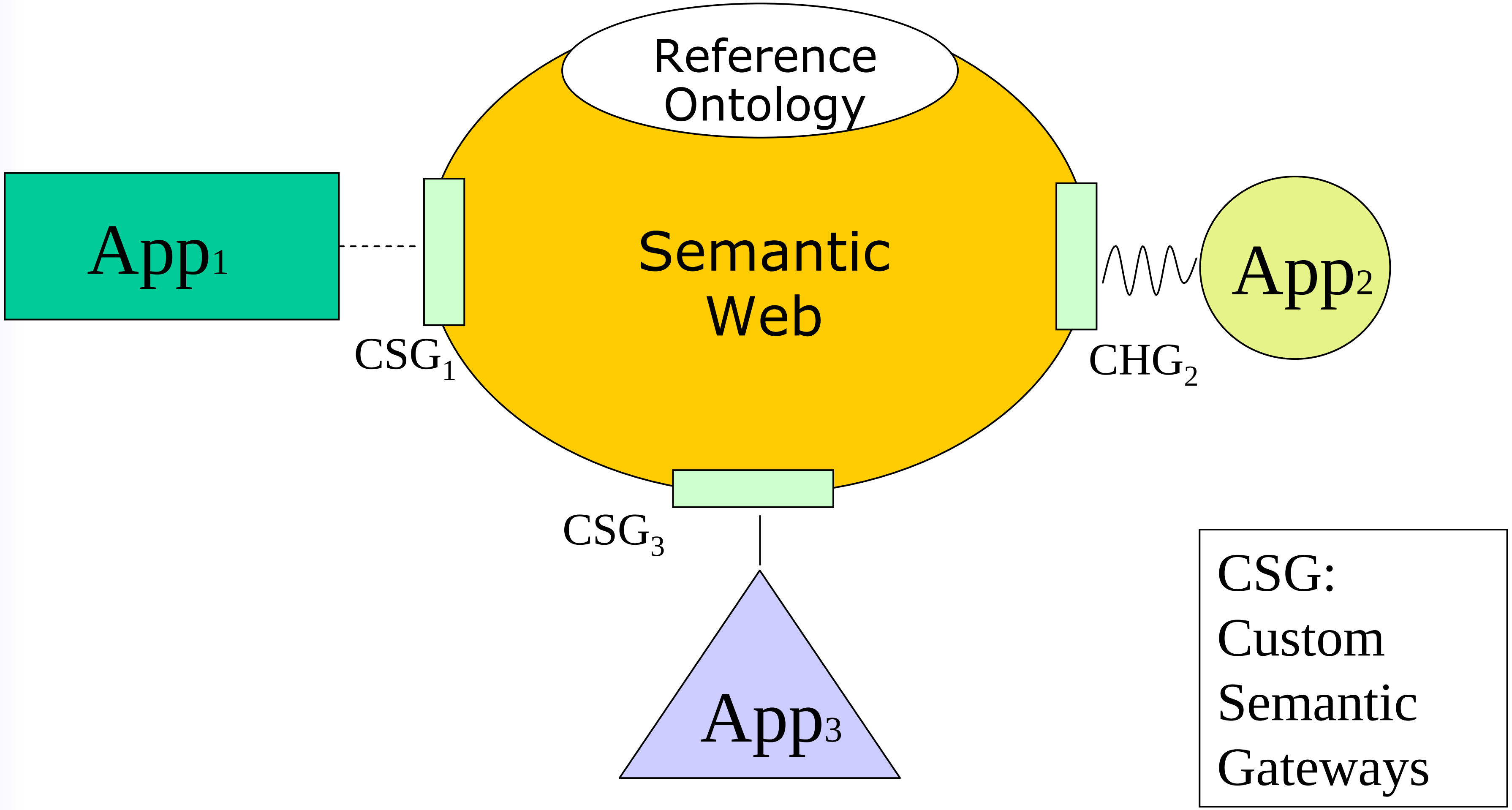
- We need a solution to achieve interoperability despite the above differences
- The ideal solution: keeping the differences (that represent a value), the role of the SW is to **reconcile** such differences

Ontology as a solution for semantic interoperability

In absence of a common interpretation



The SW for Interoperability



The Harmonise Approach to Semantic Interoperability

The Harmonise Project

An FP5 European project on tourism interoperability

- Definition of a **Reference Ontology**, in the Tourism Domain (spec. Events, Accomodation)
- **Semantic annotation** of Local Conceptual Schemas (Information Elements), by using the Reference Ontology
- **Semantic reconciliation** rules generation, starting from semantic annotation
- Harmonise Interchange Representation (**HIR**): an ontology-based representation
- **Transformation** engine for data exchange

Ontology: a definition

*An ontology is a formal, explicit specification of a shared conceptualisatio*n.

'Formal' refers to the fact that the ontology should be machine understandable.

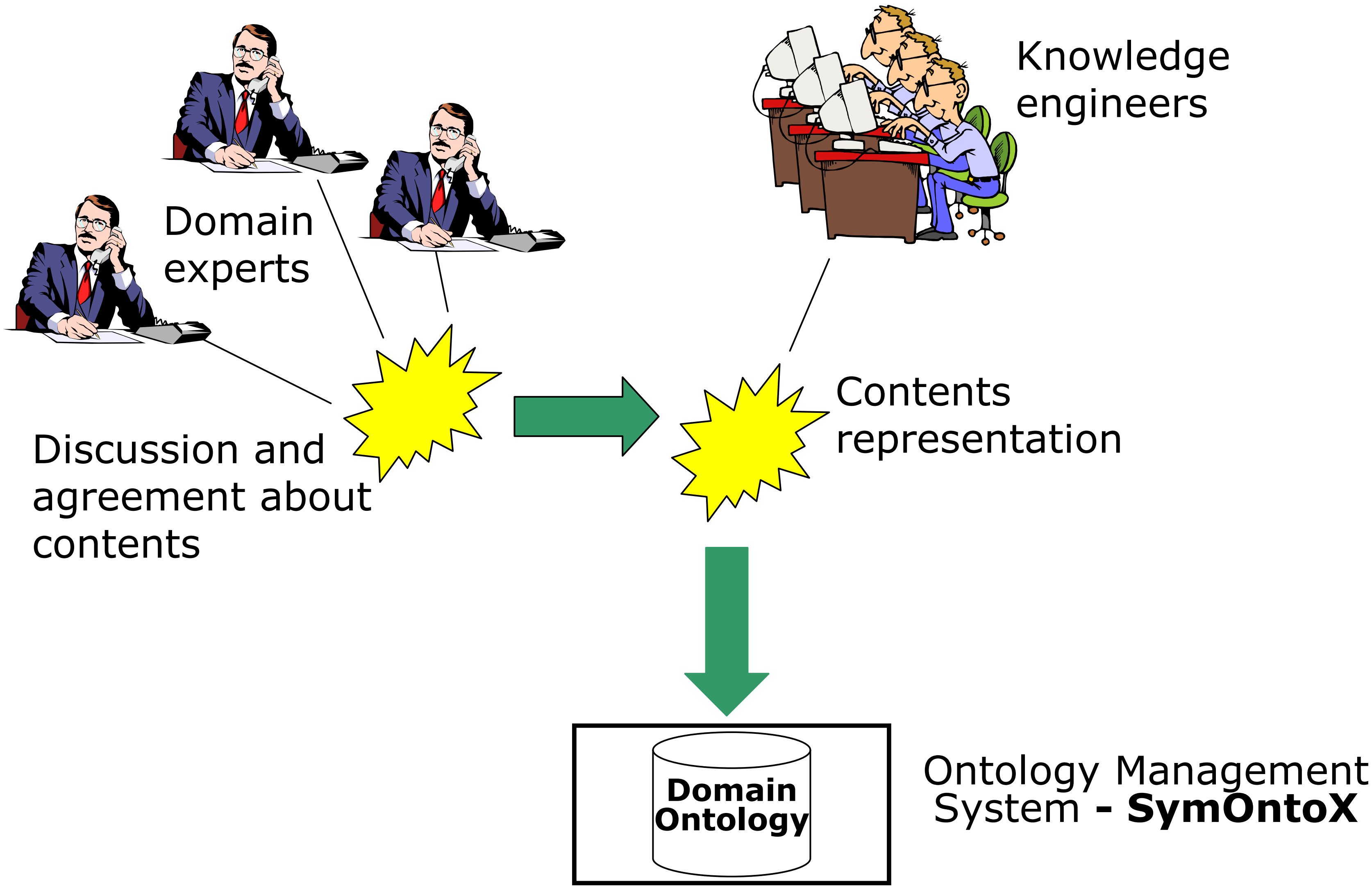
'Explicit' means that the type of concepts used and the constraints on their use are explicitly defined.

'Shared' reflects the notion that ontology captures consensual knowledge, that is, it is not restricted to some individual, but accepted by a group.

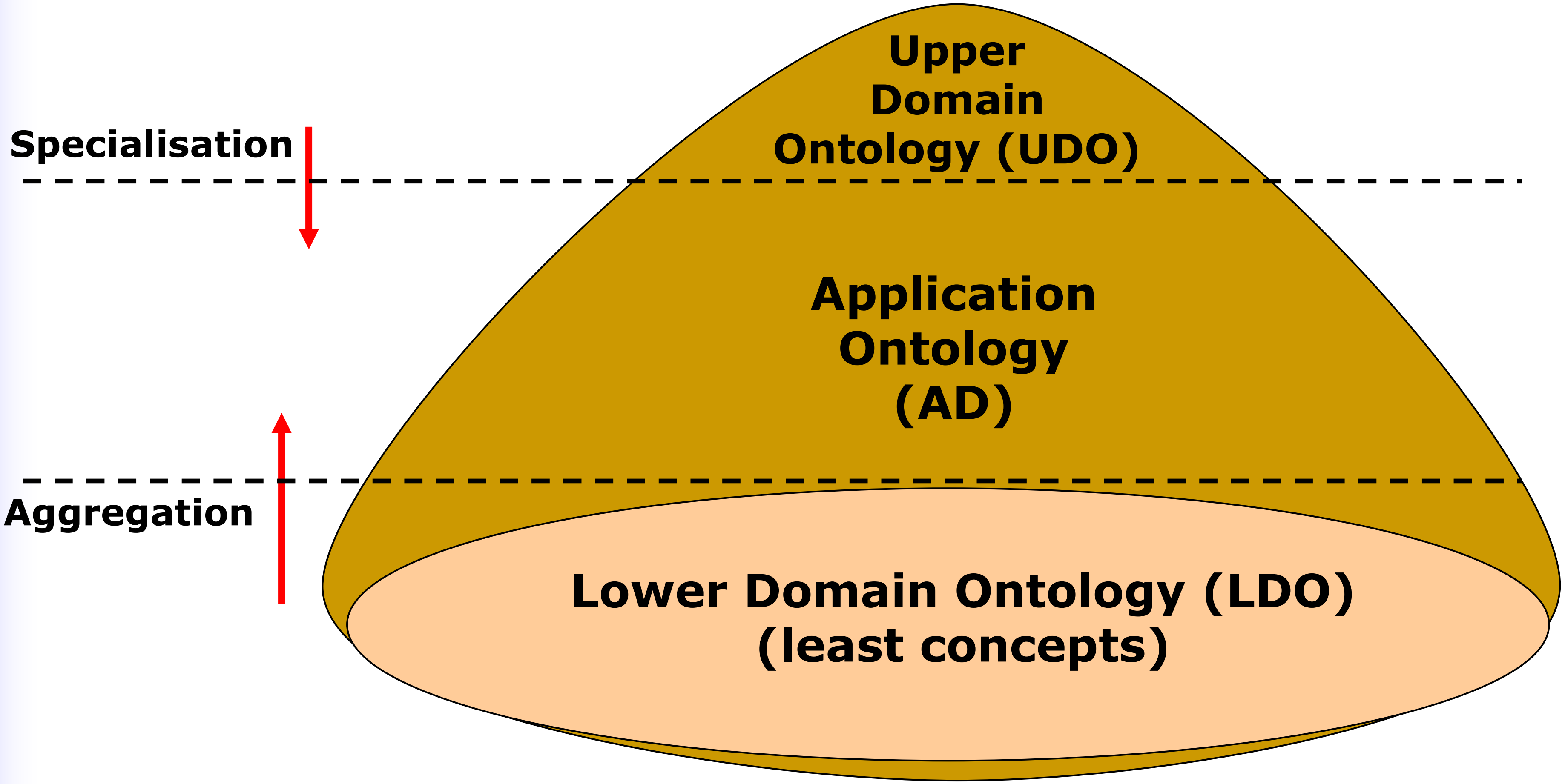
A ***'conceptualisation'*** refers to an abstract model of some phenomena in the world, which identifies the relevant concepts of that phenomena.

(Tom Gruber, Stanford U.)

The Ontology construction



The Ontology *Chestnut*



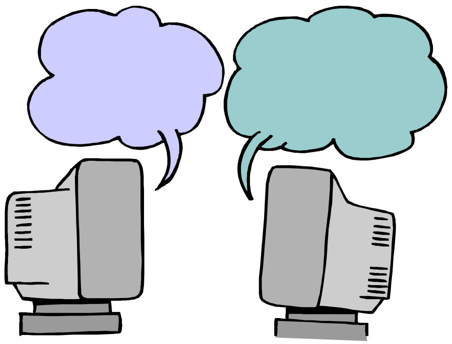
The three Ontological Layers

Where are Interoperability problems?

UDO (Top Concepts) - universal, vague, fully agreeable (accommodation, travel, customer, partner, meal)
AO (Application Specific Concepts) - very concrete, contextual, conflicting (address, confirmation, lunch, reception)
LDO (Least Concepts) - universal, too detailed, scarcely usable (Street, ArrTime, HotelName, Spaghetti)

Cooperation

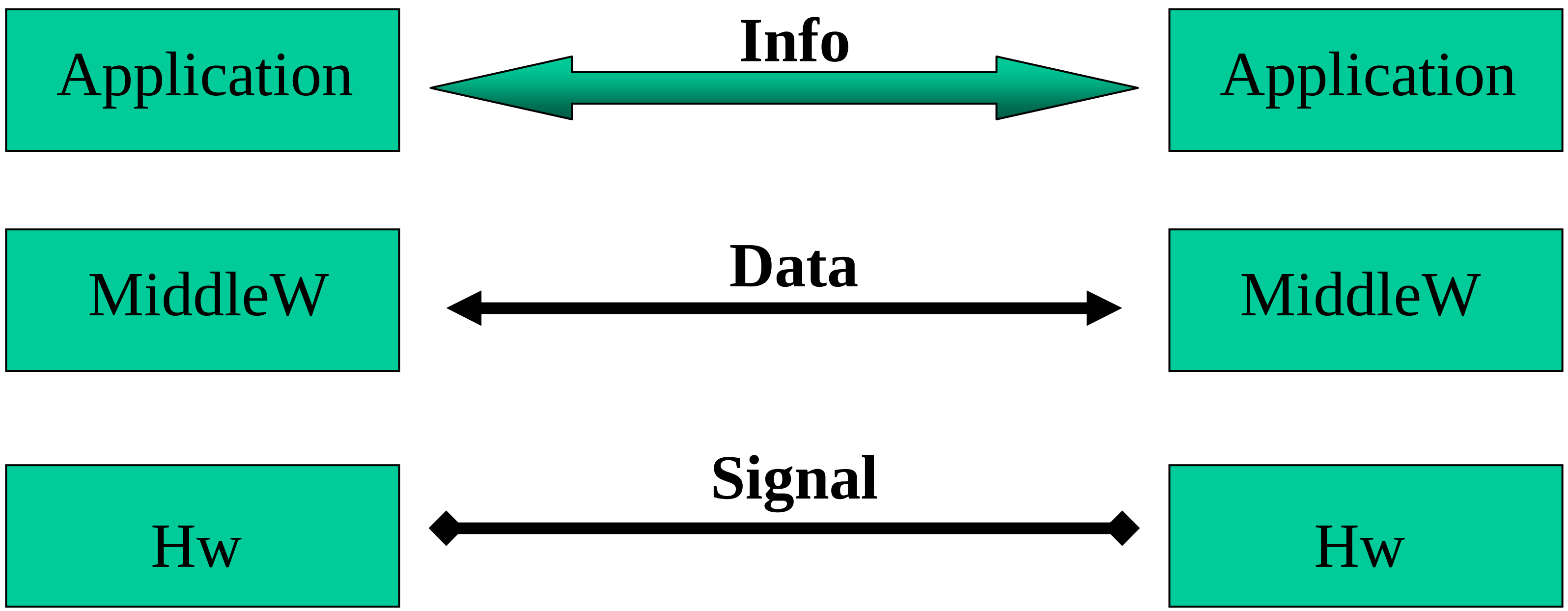
- Take place within a shared conceptual space, that includes a model of relevant entities of the Tourism Domain
- Requires to agree on a common subset of their view of the world
- Privacy: Entities (and information) outside the shared space will remain proprietary and unreachable



The Harmonise partner by means of the *Harmonise Space* (a middleware platform), communicates with the others partners as if they had his own Information view (LAV approach)

The Goal of Harmonise

- Allow tourism organisation to cooperate, without changing their information systems;
- This cooperation impacts on the 3rd level, assuming that the first two levels are available.



Info = Data+interpretation

The OPAL methodology for Ontology Modeling

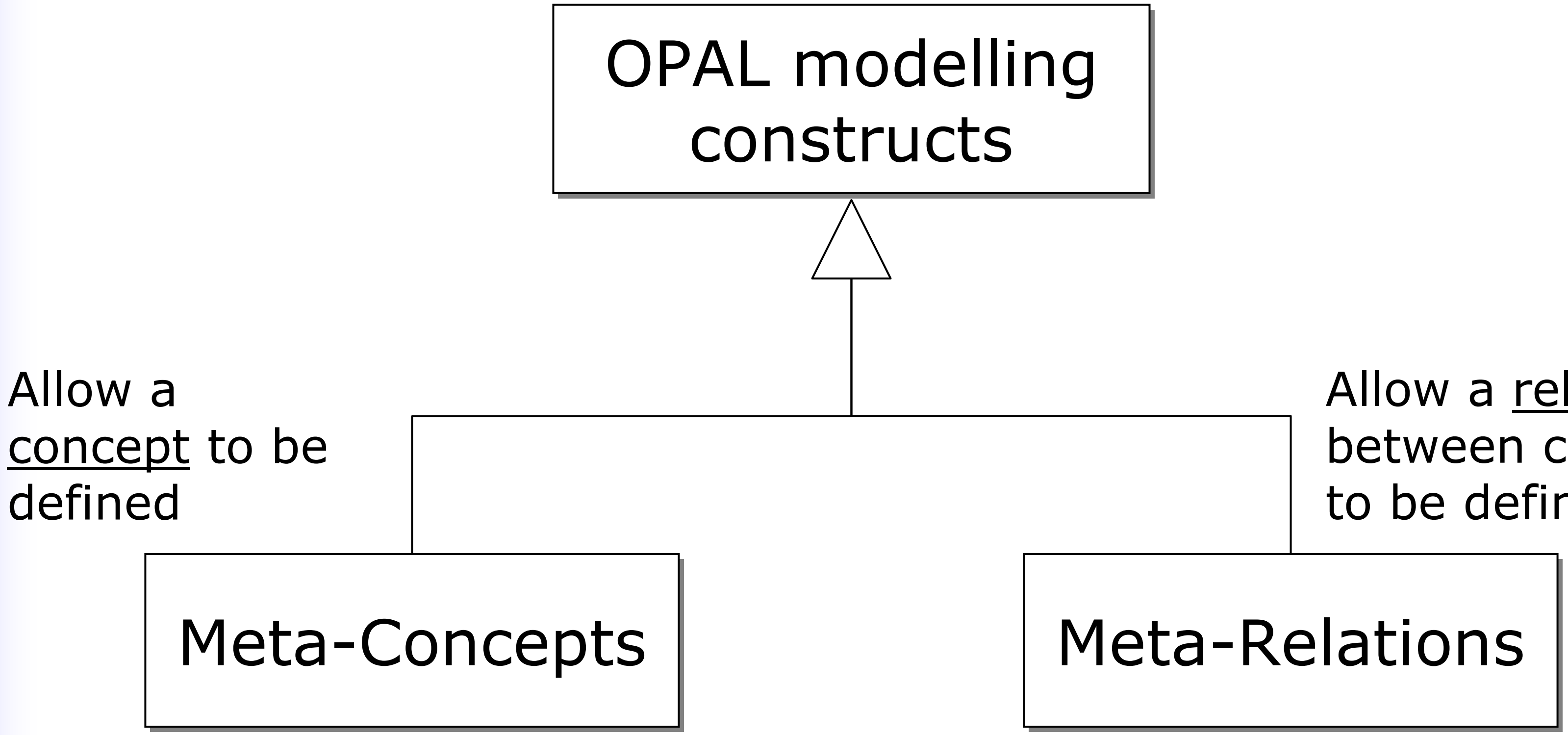
OPAL

- **OPAL** (Object, Process, Actor Language) is an Object-Oriented methodology for knowledge representation
- Developed by **LEKS** (Laboratory of Enterprise Knowledge and Systems), at IASI-CNR (Italy)

OPAL objectives

- Aims at enhancing domain adequacy in enterprise modelling
- Domain adequacy is related to the orientation to a problem area (e.g., *business management*, *medicine*, *high energy*, *physics*)
- In this perspective, OPAL introduces a number of modelling constructs, that enrich typical conceptual modelling notions (i.e., *class/frame*, *attribute/slot*, …)

Modelling constructs



Difference between meta- **and** super-**concepts**

- Super-concept
  - Is a generalization of a number of concepts;
  - Holds the key features, acquired by subconcept (*inheritance)*;
  - Specialisation (Generalisation) is the meta-relation between super and sub concept.
- Meta-concept
  - Is a sort of template for building domain concepts;
  - Holds meta-features, not inherited by domain concepts;
  - Instance-of (Classification) is the meta-relation between meta and domain concepts.

(Motschnig-Pitrik and Mylopoulos 92)

Different kinds of concepts

Primary meta-concepts

- ***Actor_kind***: allows for a relevant entity of the domain that actively participates in, typically performs, a processes (e.g, *Customer* or *Travel_Agency*)

- ***Object_kind***: allows for a passive entity on which a process operates (e.g, *Hotel, Flight*)

- ***Process_kind***: allows for an activity aimed at the satisfaction of an enterprise goal (e.g, *Hotel_Room_Reserving).* At different levels of granularity can be decomposed into: A*ction, Elementary Action*

Different kinds of concepts

Complementary meta-concepts, used in defining primary concepts

- ***Information Component***: structured property of an Actor or an Object (e.g. *Address*).
- ***Information Element***: simple properties (basic typed) or not further decomposable components of an Actor or an Object (e.g. Street_Name).

Additional meta-concepts

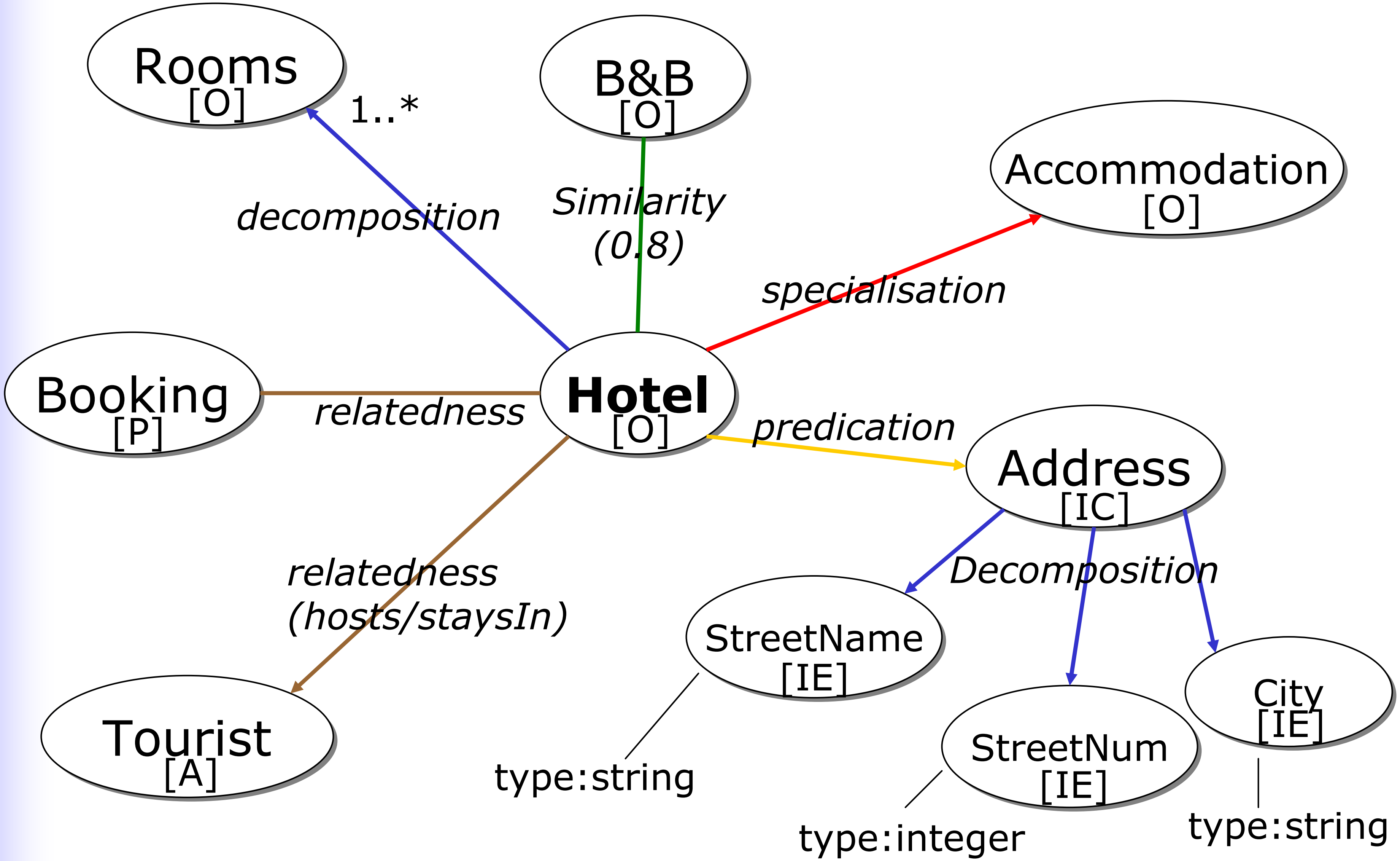
- **Message**
- **Decision**
- **Goal**
- **Exception**
- **Event**
- **Rule**
- **State**

The definition and axiomatization of meta-concepts is still object of (debatable) research

Meta-relations

- **Generalization**
- **Specialization**
- **PartOf**
- **HasPart**
- **Similar**
- **Predication** (**HasAttribute**)
- **IsAttributeOf**
- **Related** (adorned with domain relation name)

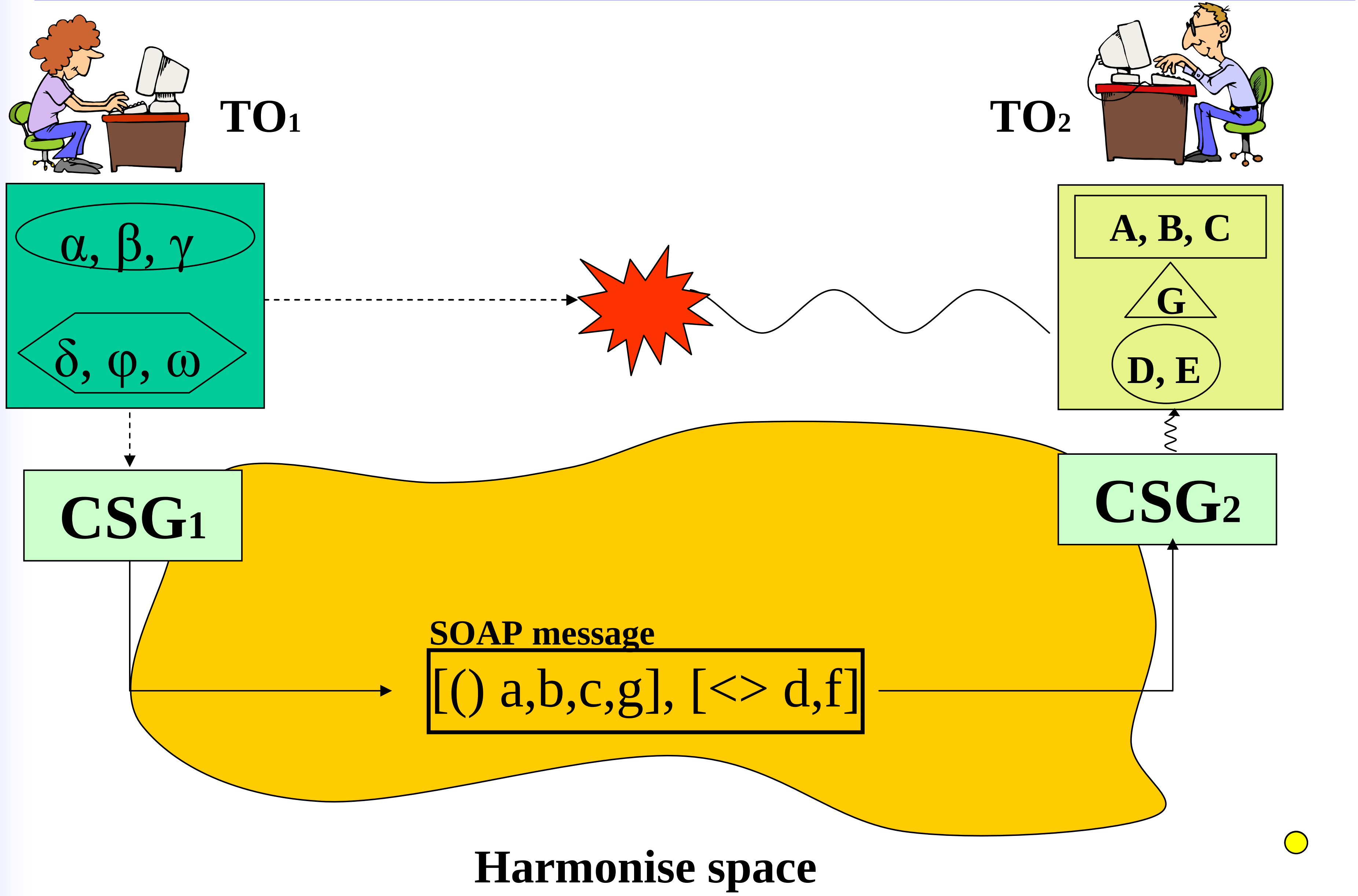
Example of relations between concepts



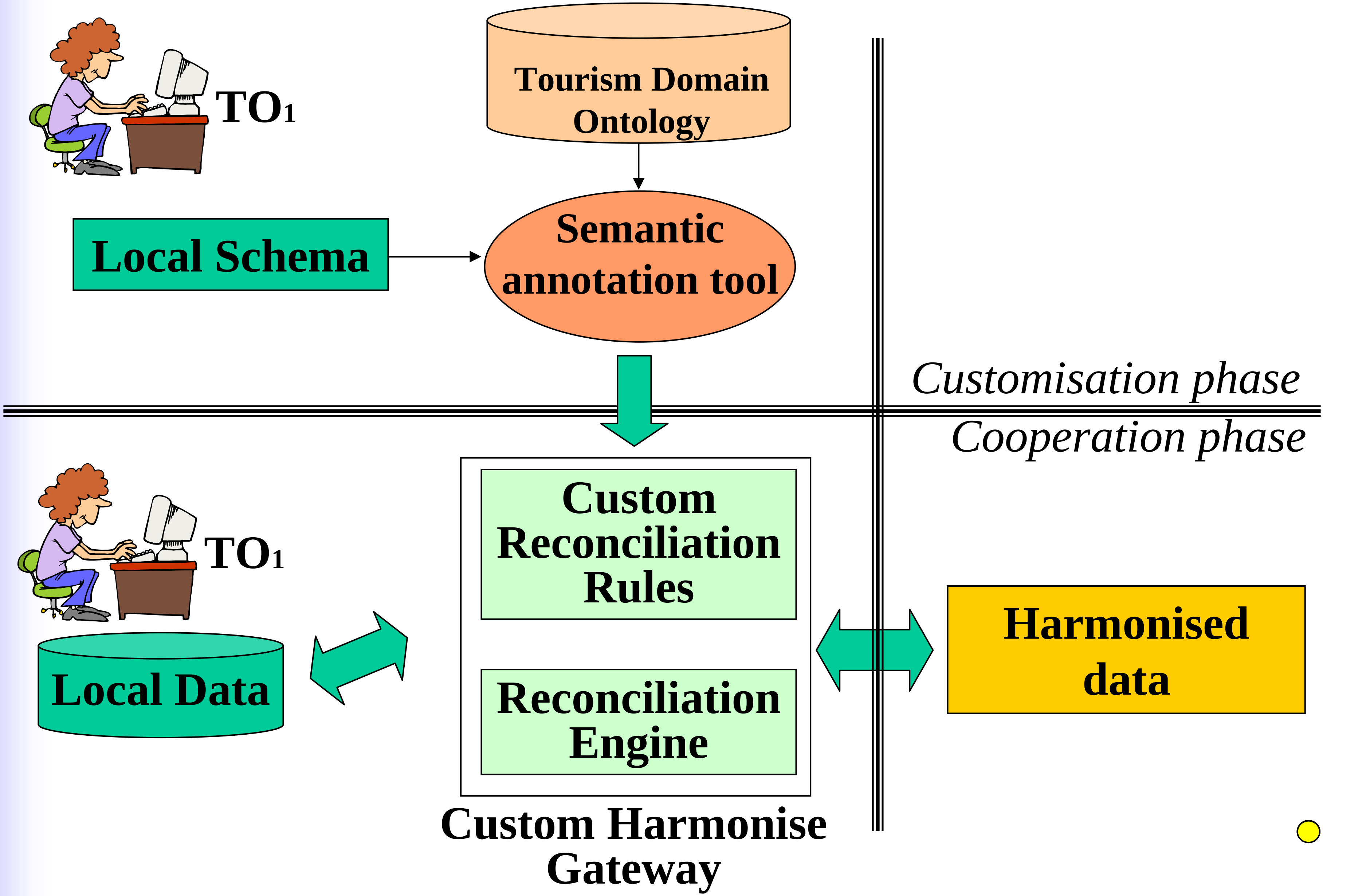
The Harmonise project

Architectural Issues

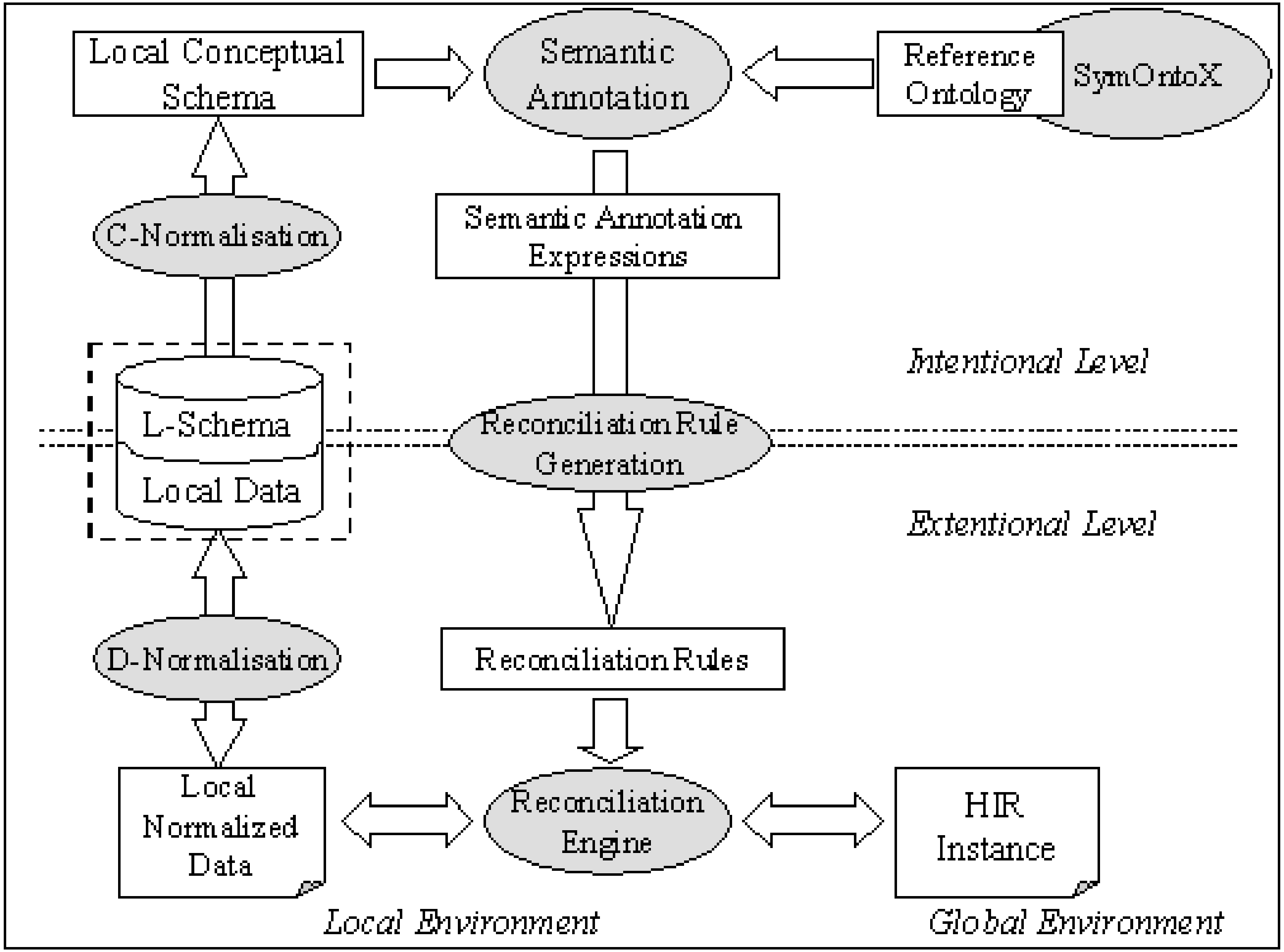
Cooperation at work



The Harmonise phases



The Harmonise platform Architecture



Focussing on resolving Interoperability Hindrance

Reconciliation and clashes

- The reconciliation rules are based on the analysis of the typical clashes that can be found comparing the Local Schema and the Reference Ontology.

- In the real cases, more than one clashes can be found at the same time.

- The clashes can be divided in two main categories:
  - **Lossless**: when the clash can be solved with a transformation that preserves the info content
  - **Lossy**: when the clash cannot be solved without loss of information

Semantic Annotation

- It gives semantics to the elements of local conceptual schema in terms of the ontology (Reference Ontology)
- It aims at solving the clashes between the Local Conceptual Schema and the Reference Ontology
- Semantic annotation is expressed by means of semantic operators

Kinds of clashes

Lossless transformation

- ***Encoding*** (different format of data or unit of measure)
- ***Typing*** (different types to represent information)
- ***Naming*** (different names for the same content)
- ***Structuring*** (different structures about the same content)

Lossy transformation

- ***Content*** (different content about the same concept)
- ***Coverage*** (the presence/absence of information)
- ***Precision*** (the accuracy of information)
- ***Abstraction*** (the granularity of information)

Semantic Transformation Operators

assoc (associate), maps an lcs_elem with a concept in the Ontology (1 to 1) (*naming clash*)

pack, maps an lcs_elem with several concepts in the Ontology (1 to m) (*structuring clash*)

extract, maps an lcs_elem with a fragment of a concept instance in the Ontology (*structuring clash*)

a_val (associate value), maps a possible value of a lcs_elem with a possible value of an Ontology concept (especially for enumerated) (*content clash*)

c_val, (compute value) computes the value of the lcs_elem by using operators (i.e. +, -, …) (*encoding clash*)

cast, converts from a basic data type of the lcs_elem into that of the Ontology concept (*typing and precision clashes*)

Examples of clashes

Hotel (LCS)

Denomination: String
Address: String
Telephone: String
Category: enum('*', '**', '***')
Services: enum('room Service', 'security box', 'parking', 'sauna')
Number_of_rooms: int
Number_of_beds: int

Hotel (Ontology)

Name: String
Address:
  Street_name: String
  Street_num: String
  City: String
Telephone: String
Category: enum('1st', '2nd', '3rd')
Services: enum('room Service', 'safe', 'shuttle to city', 'ironing center')
Number_of_rooms: int

naming (Denomination ↔ Name)
structuring (Address ↔ Address)
content (Services ↔ Services)
coverage (Number_of_beds)

Naming clash (example)

Local Conceptual Schema	**Ontology (IMHO)**
Hotel Denomination: *String* ...	Hotel Name: *String* ...

Naming

Lcs.Denomination =: **assoc**(IMHO. Name)

Structuring clash (example)

Local Conceptual Schema	**Ontology (IMHO)**
Address: String	Address Street_name: *String* Street_num: *String* City: *String*

Structuring

Lcs.Address=:

pack(IMHO.Address.Street_name, “,” ,

IMHO.Address.Street_num, “,”,

IMHO.Address.City)

Content clash (example)

Local Conceptual Schema	**Ontology (IMHO)**
Services: enum ("room Service" "security box" "parking" "sauna")	Services: enum ("room Service" "safe" "shuttle to city" "ironing center")

Content

Lcs.Services =:

assoc(IMHO.Services),

a_val(IMHO.Services("room Service"), Lcs.Services("room Service")),

a_val(IMHO.Services("safe"), Lcs.Services("security box"))

Coverage clash (example)

Local Conceptual Schema	**Ontology (IMHO)**
Hotel:	Hotel:
…	…
Number of rooms: *String*	Number of rooms: *String*
Number of beds: *String*	??

Coverage

No rule to solve this clash

Conclusions

- Semantic Web and Interoperability
  - Ontology-based infrastructure to allow co-operation
  - Harmonise solution allow Legacy Systems to keep their data and process organisation
  - Semantic annotation of legacy data
  - Reconciliation of differences
- Data Reconciliation
  - Lossless/lossy clashes
- Future activities: Semantic Reconciliation of Processes and … Enterprises (Athena IP)