Ontology Engineering

Dr. Mustafa Jarrar

University of Birzeit

mjarrar@birzeit.edu
www.jarrar.info

Reading Material

1) Everything in these slides   +   everything I say

2) Mustafa Jarrar: Towards methodological principles for ontology engineering. PhD Thesis. Vrije Universiteit Brussel. (May 2005) (Only chapter 2 and chapter 3)

3) Mustafa Jarrar: **Towards The Notion Of Gloss, And The Adoption Of Linguistic Resources In Formal Ontology Engineering**. In proceedings of the 15th International World Wide Web Conference (WWW2006). Edinburgh, Scotland. Pages 497-503. ACM Press. ISBN: 1595933239. May 2006.http://www.jarrar.info/publications/J06.pdf.htm

Outline

➡️ Ontology Engineering Challenges

- Ontology Double-Articulation

Ontology Engineering Challenges

❖ Ontology Usability verses Ontology Reusability

❖ Ontology Application Dependence

- Only these challenges will be discussed, but there are many other challenges that may face an ontology engineer.

- Discussing such challenges will help improve the modeling skills of an ontology engineer.

Ontology Reusability vs Usability
Given 4 different LegalPerson ontologies (which is more usable/reusable?)

O1 — Used by App1, 9000 times/day.

O2 — Used by App1, 1000 times/day.
Used by App2, 1000 times/day.

O3 — Used by App1, 100 times/day.
Used by App2, 100 times/day.
Used by App3, 100 times/day.

O4 — Used by App1, 10 times/day.
Used by App2, 10 times/day.
Used by App3, 10 times/day.
Used by App4, 10000 times/day.

App1: Ministries' Web Service to exchange companies' profiles is based on this ontology.
App2: Champers of commerce's Web Service to exchange companies' profiles, based on this ontology.
App3: Banks designed their "new account" form, based on the company properties in this ontology (off time use).
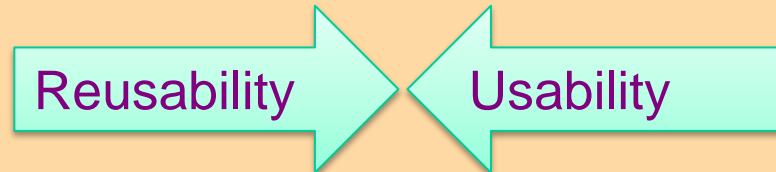**App4:** Lawyers refer to the definition of "company", as stated in this ontology (off time use).

Ontology Reusability vs Usability
Given 4 different LegalPerson ontologies (which is more usable/reusable?)

Usability: maximizing the number of different applications using an ontology *for the same kind of task*.

Reusability: maximizing the number of different applications using an ontology *over different kind of tasks*.

Why Reusability:
1) Saving time, cost, and efforts…
2) Increasing reliability: the more reused the more tested.
3) An important quality factor: a highly reusable ontology is an indication that it is a good ontology.

How to increase Usability?
by being closer to the application specifications and requirements at hand.

How to increase Reusability?
by taking into account different usages/applications, i.e. be more general.

App1: Mi
App2: Ch
App3: Ba   se).
App4: Lawyers <u>refer</u> to the definition of "company", as stated in this ontology (off time use).

Ontology Reusability vs Usability

Given 4 different LegalPerson ontologies (which is more usable/reusable?)

Usability: maximizing the number of different applications using an ontology *for the same kind of task*.

Reusability → ← Usability

Tradeoff between usability and reusability

➤ The more an ontology is usable the less reusable it will be, and vice versa.

➤ A good ontology engineer knows how/where to compromise this tradeoff.

by being closes to the application specifics and requirements at hand.

How to increase Reusability?
 by taking into account different usages/applications, i.e. be more general.

App1: Mir
App2: Ch
App3: Ba   se).
App4: Lawyers <u>refer</u> to the definition of "company", as stated in this ontology (off time use).

Ontology Application Dependence

Ontologies are supposed to capture knowledge at the domain level independently of application requirements [G97] [GB99] [CJB99].

The problem is that when building an ontology, there will always be intended or expected usability requirements -"at hand"- which influence the independency level of ontology axioms.

This problem is as the Interaction Problem:

> *"Representing knowledge for the purpose of solving some problem is strongly affected by the nature of the problem and the inference strategy to be applied to the problem."*

Bylander and Chandrasekaran in [BC88]

Ontology Application Dependence
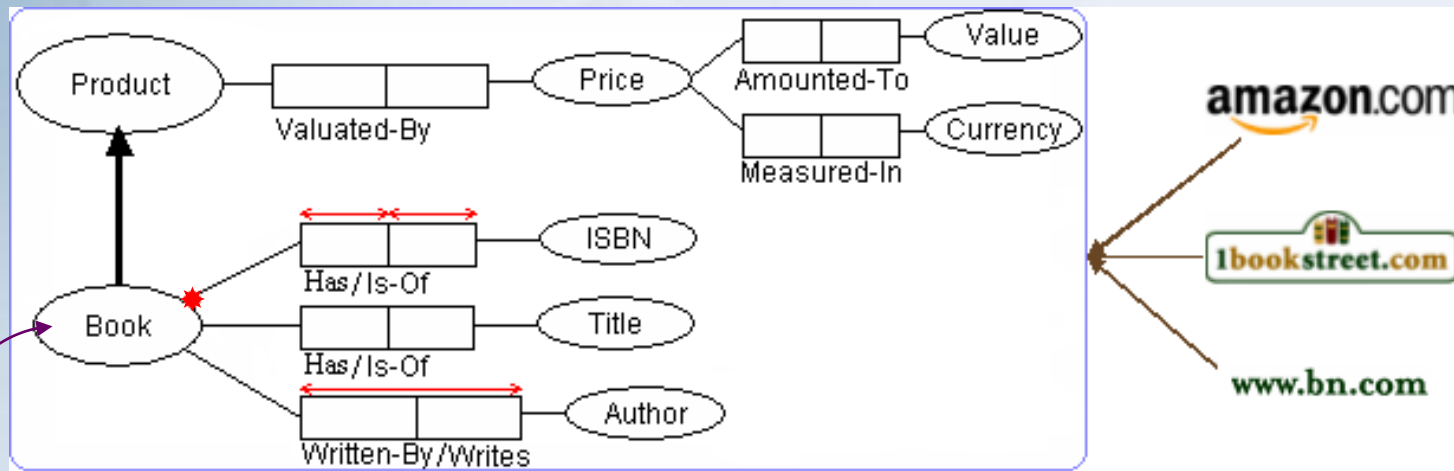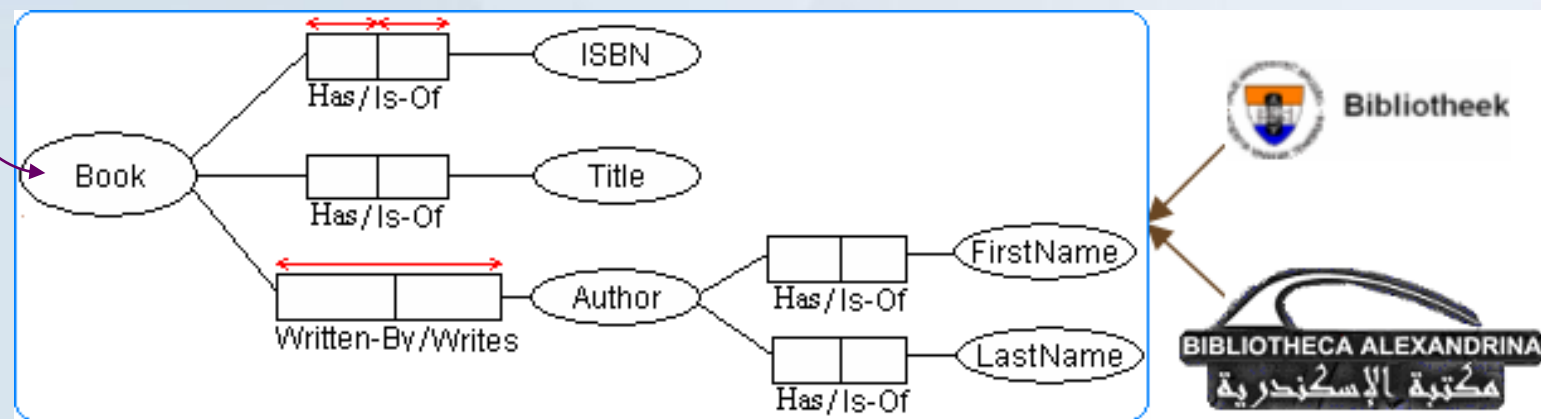
What is the meaning of a "book" here?

> Usability perspectives lead to different (and sometimes conflicting) axiomatizations although these axiomatizations might agree at the domain level.

Ontology Application Dependence

What is the meaning of a "book" here?
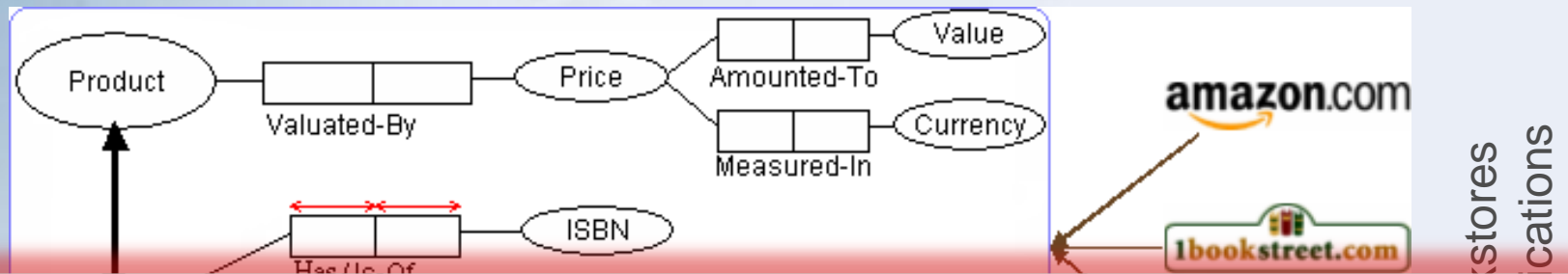
Both are not ontologies, they are data schemes.

➤Can you build a useful and an application-independent ontology?
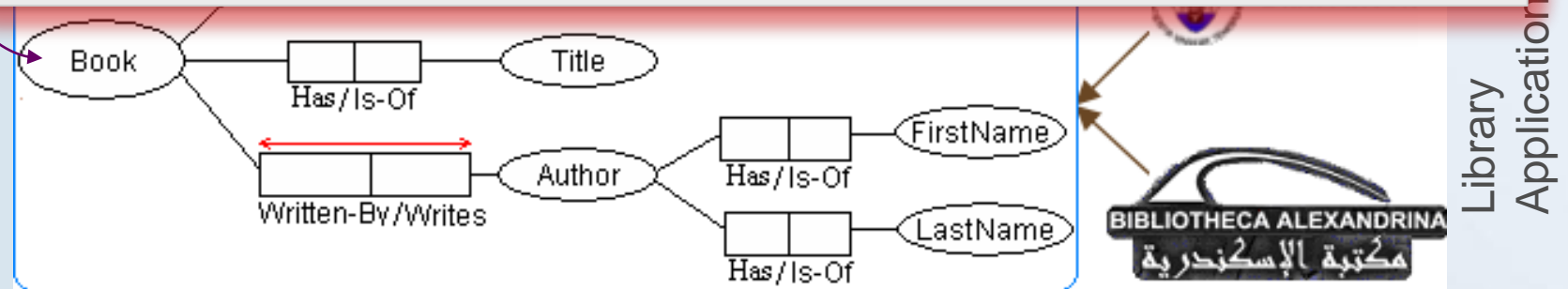
➤ Usability perspectives lead to different (and sometimes conflicting) axiomatizations although these axiomatizations might agree at the domain level.

Outline

- Ontology Engineering Challenges
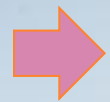
➡ Ontology Double-Articulation

Knowledge Double-Articulation
A methodology to engineer ontologies
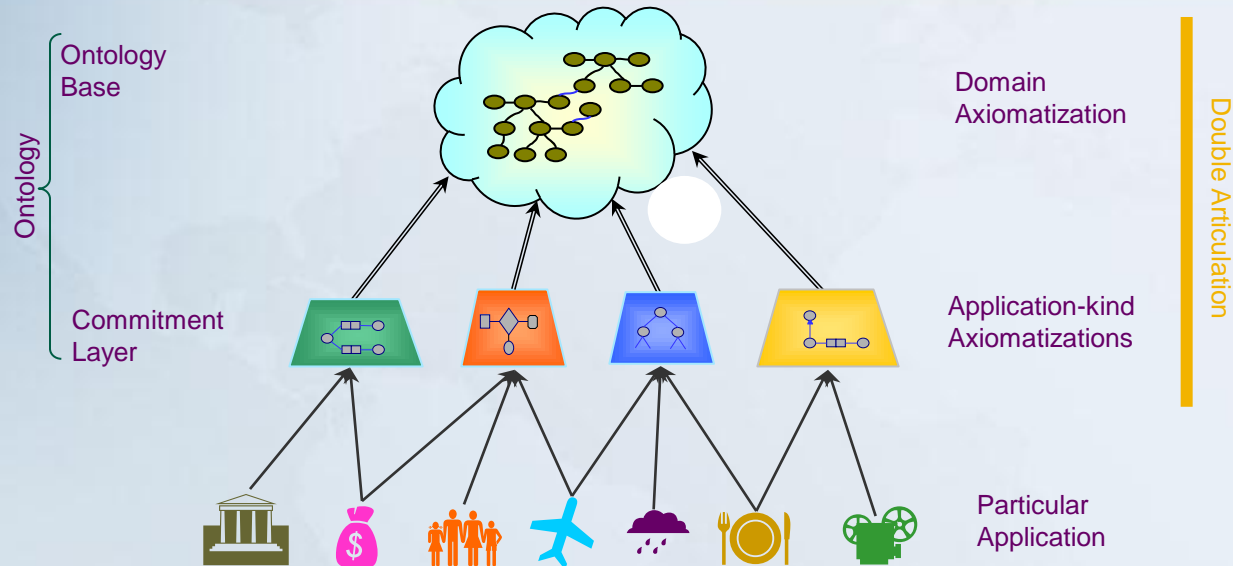
The meaning of a vocabulary should be doubly-articulated into domain axiomatization and application axiomatization(s).

- Domain axiomatization (or a linguistic resource) is mainly concerned with characterizing the "intended meaning/models" of a vocabulary at the community/domain level.
- Application axiomatization is more concerned with the utility of these vocabularies according to certain application/usability perspectives.
- Ontologies built in this way are easier to build, highly reusable and usable, easier to integrate with other ontologies, and smoother to maintain.

Knowledge Double-Articulation

Highly reusable (domain/community level)

Domain axiomatization

Highly usable (application level)

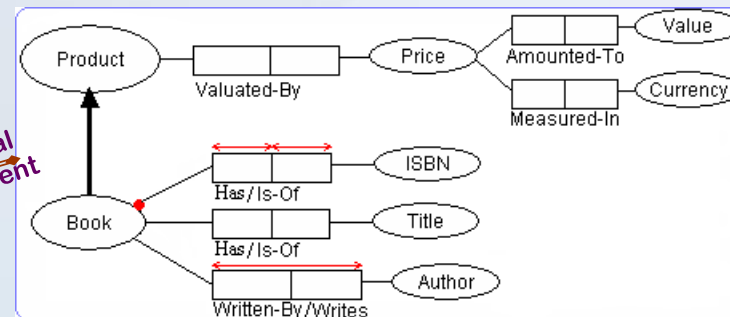Application-kind axiomatizations

Particular Applications

Knowledge Double-Articulation

Highly reusable (domain/community level)

Domain axiomatization

Highly usable (application level)

Application-kind axiomatizations

Particular Applications

OntologyBase, holding linguistic knowledge, such as

WordNet

Context	Term1	Role	InvRole	Term2
Bibligray	Written Material	Is-A	Has-Type	Product
Bibligray	Book	Is-A	Has-Type	Written Material
Bibligray	Book	Has	Is-Of	ISBN
Bibligray	Book	Has	Is-Of	Title
Bibligray	Book	Issued-By	Issues	Publisher

Bib-Topics	Topic	SuperTopicOf	SubTopicOf	Sports
Bib-Topics	Topic	SuperTopicOf	SubTopicOf	Arts
Bib-Topics	Computers	SuperTopicOf	SubTopicOf	Computers Sci
Bib-Topics	Computers	SuperTopicOf	SubTopicOf	Programming
Bib-Topics	Computers	SuperTopicOf	SubTopicOf	Product
Bib-Topics	Product	SuperTopicOf	SubTopicOf	CASE Tools
Bib-Topics	Product	SuperTopicOf	SubTopicOf	Word Processo
Bib-Topics	Product	SuperTopicOf	SubTopicOf	DBMS

Biblgray Ontology Base

Ontological Commitment

Ontological Commitment
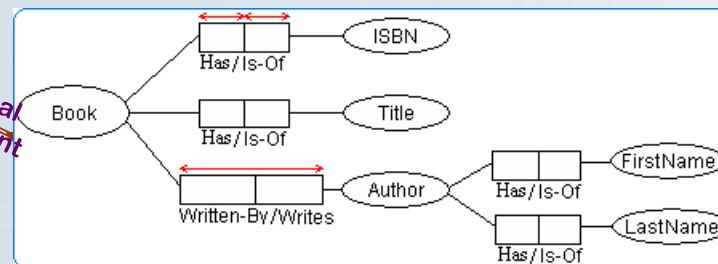
amazon.com

1bookstreet.com

www.bn.com

Bibliotheek

BIBLIOTHECA ALEXANDRINA
مكتبة الإسكندرية

Knowledge Double-Articulation

Domain axiomatization

Highly usable (application level)
Application-kind axiomatizations

Particular Applications

- ❖ **accounts for the intended meaning of domain vocabularies;**

- ❖ **rooted at a human language/community conceptualization.**

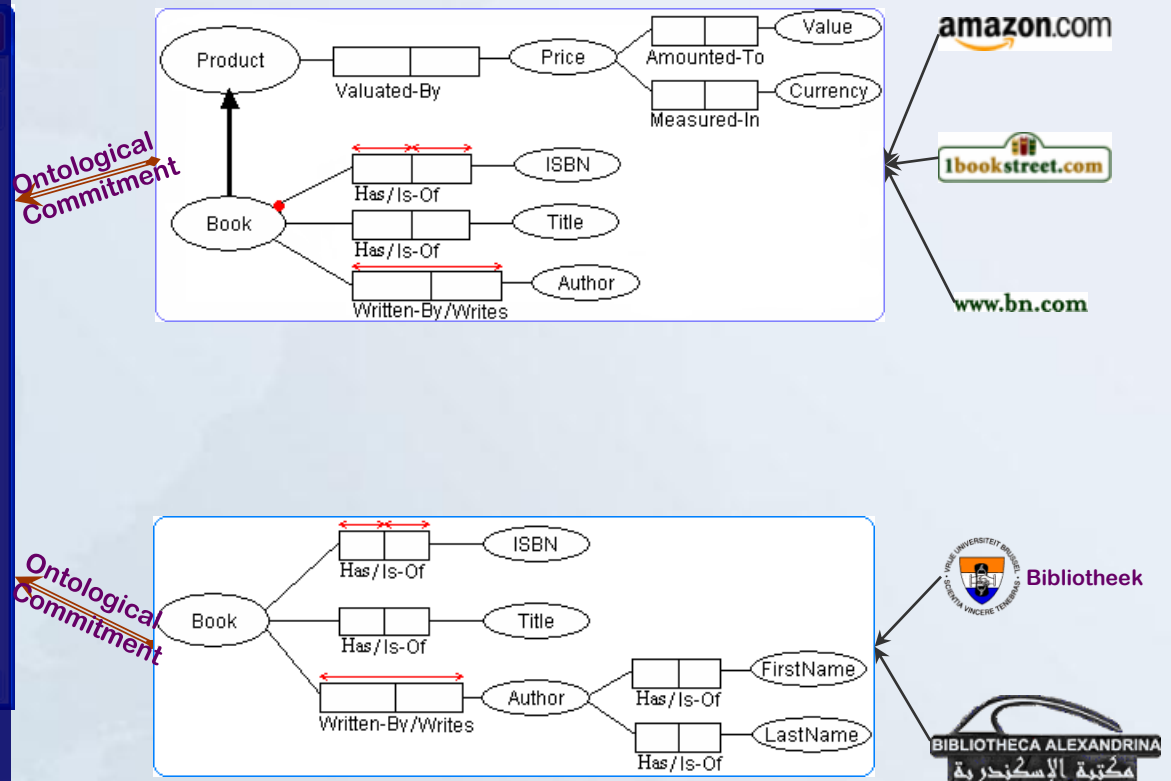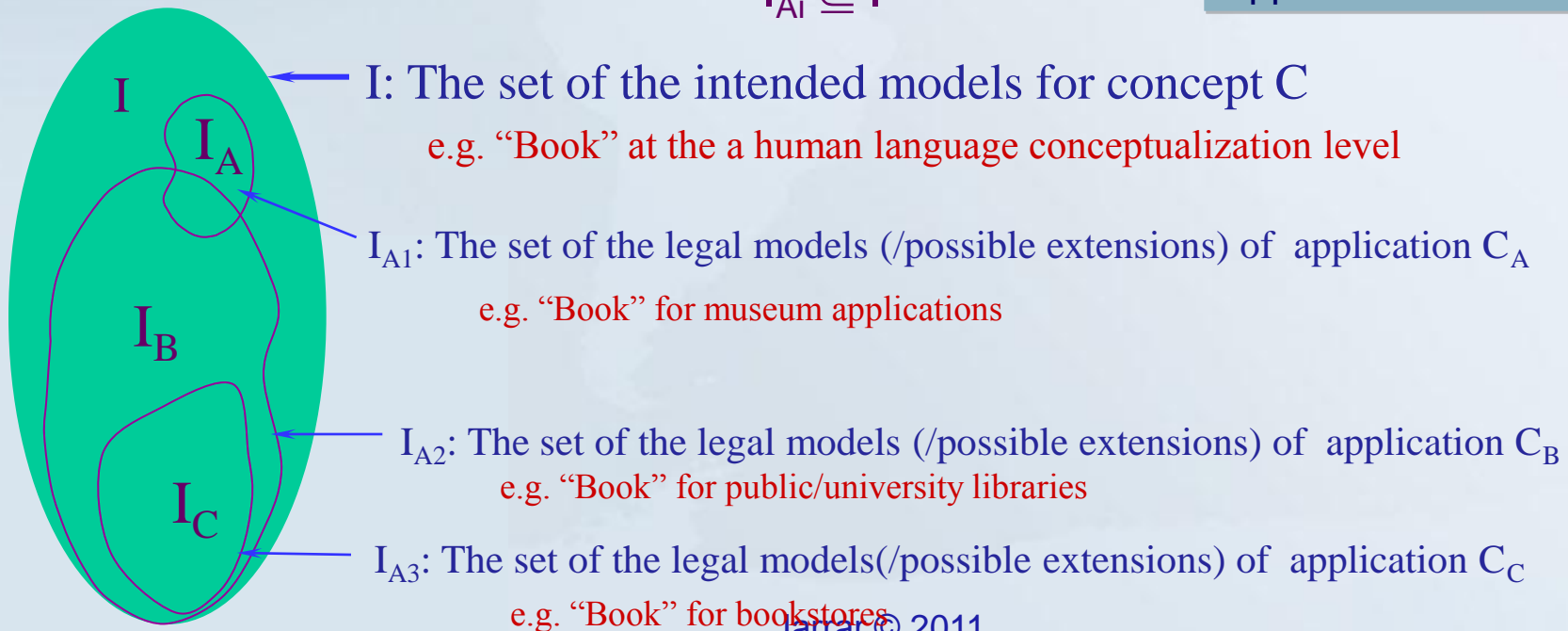- ❖ **interpreted intensionally;**

- ❖ **a shared vocabulary space for application axiomatizations;**

Knowledge Double-Articulation Theory

- **A concept** is *a set of rules in our mind about a certain thing in reality*.

- For concept C, the set I of "all possible" instances that comply with these rules are called the ***intended models*** of the concept C.

  Domain/Language Level

- An application A that is interested -according to its usability perspectives- in a subset I_{Ai} of the set I, is supposed to provide some rules to specialize I, I_{Ai} is called ***legal models.***

  Application Level

  $$I_{Ai} \subseteq I$$

I: The set of the intended models for concept C

e.g. "Book" at the a human language conceptualization level

I_{A1}: The set of the legal models (/possible extensions) of application C_A

e.g. "Book" for museum applications

I_{A2}: The set of the legal models (/possible extensions) of application C_B

e.g. "Book" for public/university libraries

I_{A3}: The set of the legal models(/possible extensions) of application C_C

e.g. "Book" for bookstores

Applying the Double-Articulation Theory

To apply the Double-Articulation Theory in practice you may assure that your ontology is engineering in this way:

❶ **Rooting vocabulary:** all vocabulary used in an application axiomatization is linked with a vocabulary in the domain axiomatization (which can be linguistic resources, e.g., WordNet). e.g., each concept in an ORM model/OWL file is linked with a concept WordNet/ArabicOntology.

❷ **Glosses:** If a certain vocabulary does not exist in the domain axiomatization, then it must define entroduced with gloss.

❸ **Context:** Each application axiomatization must have a context, as its scope of interpretation.

❹ **Modularize** application axiomatization into several modules.

❶ **Rooting vocabulary**

Each vocabulary in your ontology can be linked (e.g. though a namespace) with a concept in a linguistic resource (e.g. a synset in WordNet).
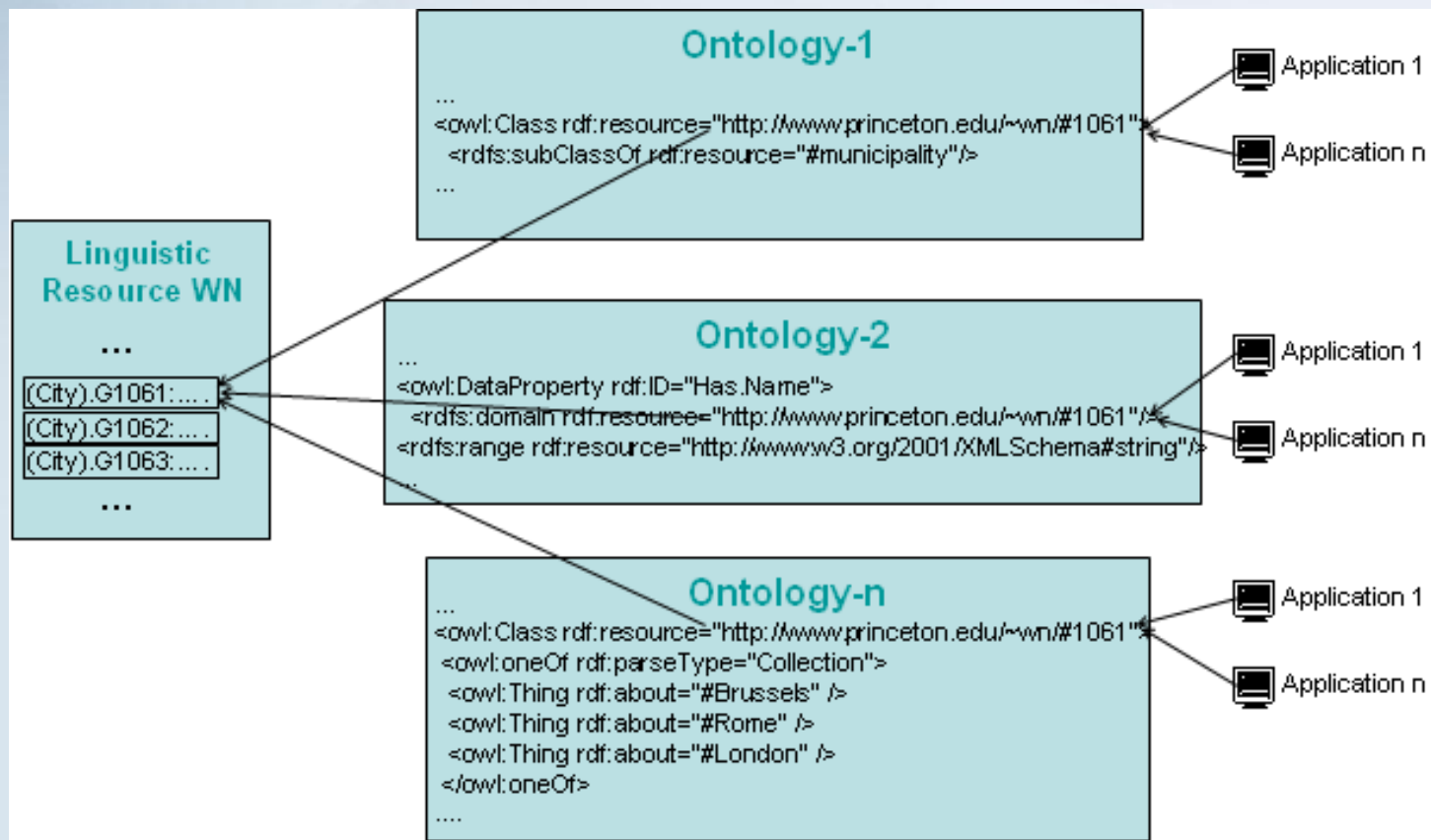
Example (Customer Complaint Ontology)

Central complaining portal

See http://www.jarrar.info/publications/mjarrar-CCFORM-chapter.pdf.htm

Example (Customer Complaint Ontology)

See http://www.jarrar.info/publications/mjarrar-CCFORM-chapter.pdf.htm

20

CC Ontology (Example)

CC Ontology base: 300 lexons

CCcontext

CC Glossary: 220 glosses

Domain Axiomatization

7 axiomatization Modules

Contract   Recipient   Complaint   Problem   Complianant   Resolution   Address

$CCApplication_1$ $CCApplication_2$ $CCApplication_n$

❷ Defining **Glosses**

An auxiliary *informal (but controlled) account* of the intended meaning of a linguistic term, for the commonsense perception of humans.

A gloss is supposed to render factual knowledge that is critical to understand a concept, but that e.g. is implausible, unreasonable, or very difficult to formalize and/or articulate explicitly

(NOT) to catalogue general information and comments, as e.g. conventional dictionaries and encyclopedias usually do, or as <rdfs:comment>.

The ontological notion of **Gloss**

What should and what should not be provided in a gloss:

1. Start with the *principal/super type* of the concept being defined.

   E.g. 'Search engine': "A computer program that …", 'Invoice': "A business document that…", 'University': "An institution of …".

2. Written in a form of propositions, offering the reader *inferential knowledge* that help him to construct the image of the concept.

   E.g. Compare 'Search engine':
   "A computer program for searching the internet, it can be defined as one of the most useful aspects of the World Wide Web. Some of the major ones are Google, ….";
   A computer program that enables users to search and retrieves documents or data from a database or from a computer network…".

3. Focus on distinguishing characteristics and intrinsic prosperities that differentiate the concept out of other concepts.

   E.g. Compare, 'Laptop computer':
   "A computer that is designed to do pretty much anything a desktop computer can do, it runs for a short time (usually two to five hours) on batteries".
   "A portable computer small enough to use in your lap…".

The ontological notion of **Gloss**

4. Use supportive examples :

- To clarify cases that are commonly known to be false but they are true, or that are known to be true but they are false;

- To strengthen and illustrate distinguishing characteristics (e.g. define by examples, counter-examples).

  Examples can be types and/or instances of the concept being defined.

5. Be consistent with formal definitions/axioms.

6. Be sufficient, clear, and easy to understand.

❸ Specifying a **Context**

- *Context: A scope of Interpretation*
- *That is: An abstract identifier that refers to implicit (or maybe tacit) assumptions, in which the interpretation of a term is bounded to a concept*

In In practice, we define context by referring to a source (e.g. a set of documents, laws and regulations, informal description of "best practice", etc.), which, by *human understanding*, is assumed to "contain" those assumptions. Concepts, relations and rules are assumed (by human understanding) to be "true within their context's source".

Context (Example)

Customer complaining Context

Background knowledge (i.e. explicit, implicit, or tacit assumptions) about all (activities, communications, institutions, people, places, objects, etc.) that are involved in consumer-provider relationships, regarding contractual and non-contractual complaining issues.

These assumptions can be understood (i.e. can be found explicitly or intuitively) in the following sources:

- European Distance Selling Directive (97/7/EC), on the promotion of consumers in respect of distance contracts.

- European e-Commerce Directive (2000/31/EC) on certain legal aspects of information society services, in particular, electronic commerce, in the Internal Market.

- European Data Protection Directives (95/46/EC and 97/66/EC) on the protection of individuals with regards to the processing of personal data and on the free movement of such data.

- European Directive (99/44/EC) on aspects of the sale of consumer goods and associated guarantees.

- European Directive (98/27EC) on Injunctions for the Protection of Consumers' Interests.

- CEN/TC331 Postal Services EN 14012:2002 Quality of Service – Measurement of complaints and redress procedures.

- "Best practice" guidelines, The Nordic Consumer Ombudsmen's position paper on trading and marketing on the Internet and other similar communication systems(http://econfidence.jrc.it, June 2002)
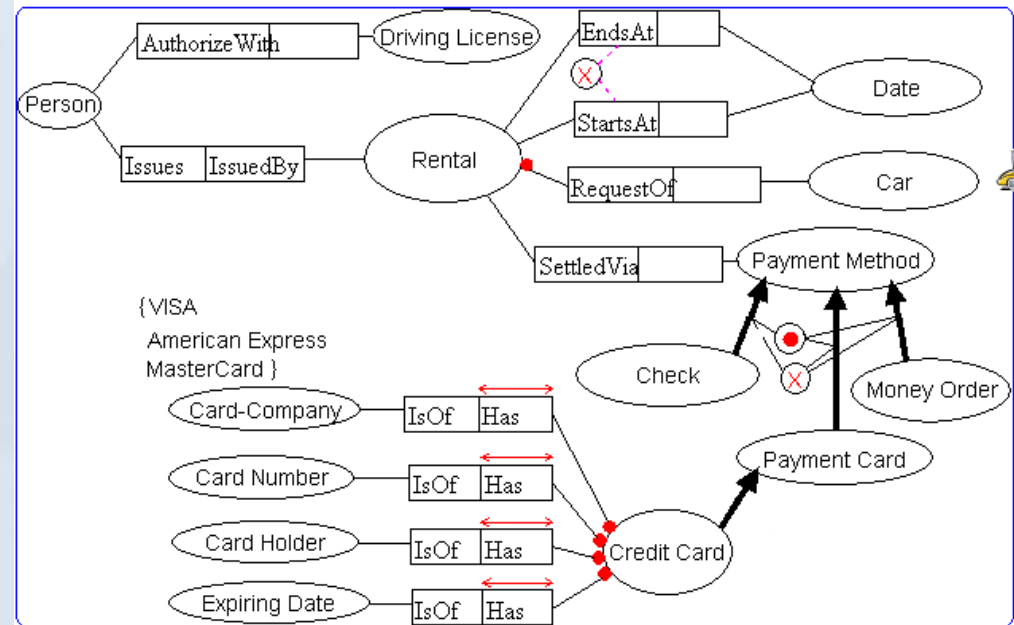
- CCFORM Annex 1, (IST-2001-34908, 5th framework).
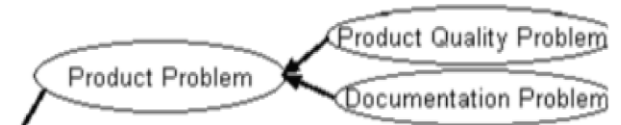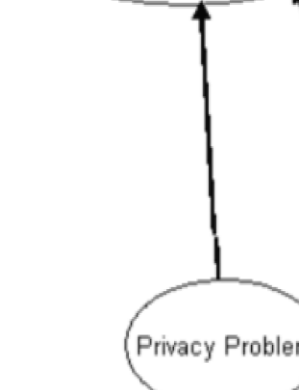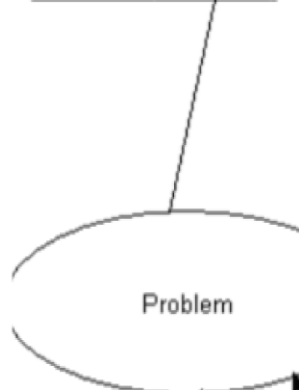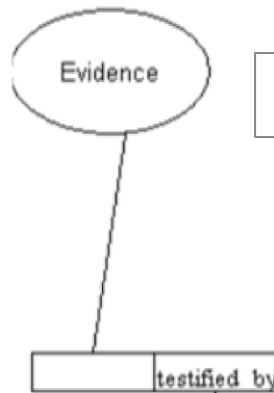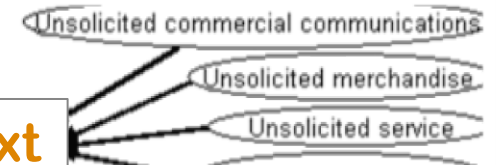
❹ Ontology Modularization

> Develop an application axiomatization as a set of modules and later compose to form one module.
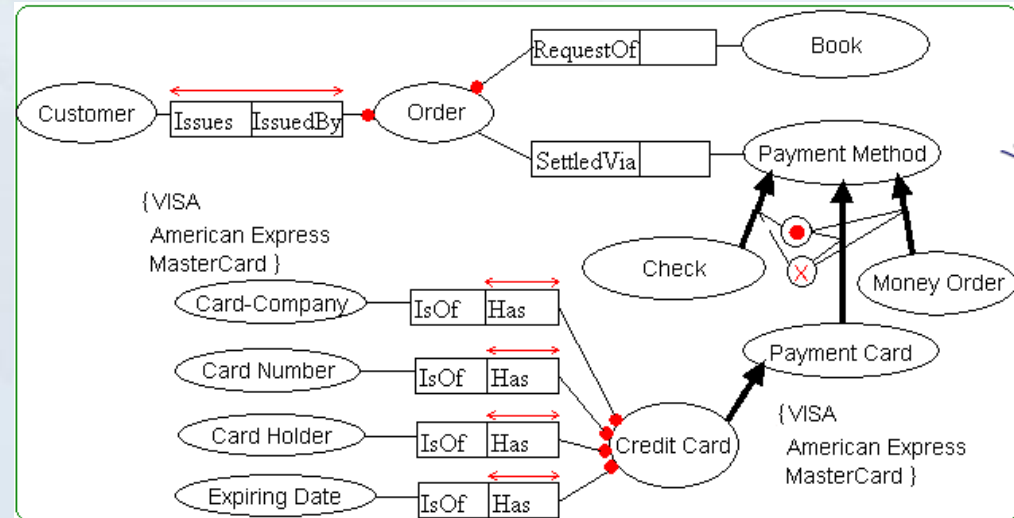
Ontology Modularization (why? How?)

Why to modularize?

Because Modules are:

1. Easier to reuse
2. Easier to build, maintain, and replace
3. Enable distributed development of modules
4. Enable the effective management and browsing

When to modularize?

Modularity criteria:

1. Subject-oriented
2. Purpose/Task-oriented
3. Stability