
Ontology storage & management and Integration within 3D city models

Gilles Falquet
Claudine Metral
Université de Genève
CUI - IAUG

Contents

Managing multipoint of view ontologies

Storing and interfacing ontologies

Integrating data, ontologies and 3D city models

The ConcepTerm Experiment

Goal:

- build a terminological database in a specific domain (furniture)
- provide description-based access
- provide a description language for terminologists

1000 concepts

drawn from sources in 4 different languages

Do not rely on usual equivalences

(*chair, chaise, sedia*) ⇔ 3 concepts

compute equivalence based on concept definitions

Term equivalence based on characteristics

armoire < meuble de rangement

size: grand

part [1, *]: corps,

part [1, *]: porte,

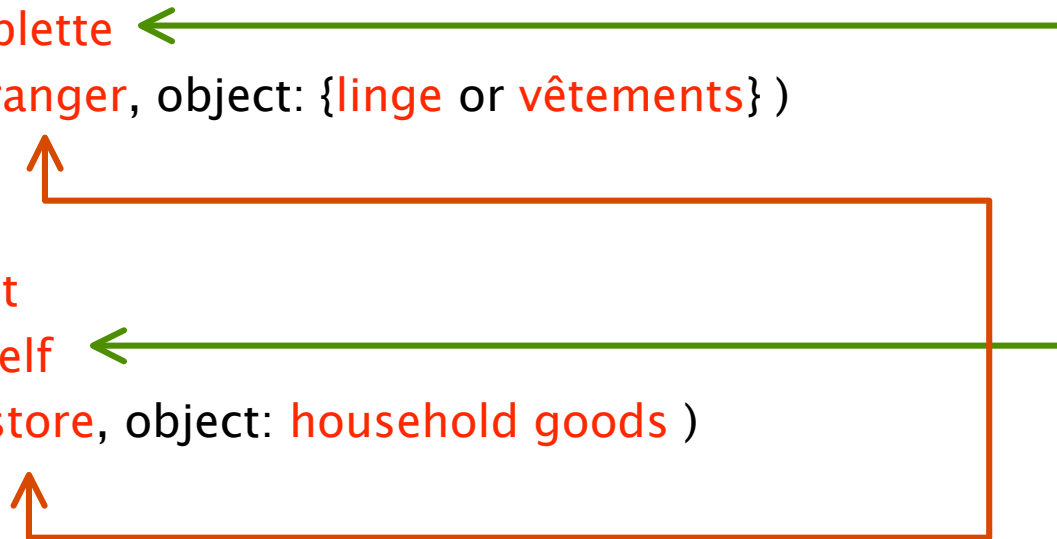
part [several] : tablette

main use: (verb: ranger, object: {linge or vêtements})

cupboard < closet

part [several] : shelf

main use: (verb: store, object: household goods)



Managing conflicts in ontology building

Context:

building an ontology with a lexical level

concept --> {term}

concept --> definition

set of semantic relationships

description logic

Conflict:

in a specialized domain

[Gaines & Shaw] same term --> different concepts

==> resolution process (agree on a definition)

Multipoint of view model

Accept several definitions for a term

Provided

they belong to different point of views
their interpretations are "sufficiently close"

What is an electron ?

physicist: "particle with charge -1 , spin $1/2$ and ... "

chemist: "part of atoms, agent of chemical bounds ..."

electrical engineer: "current carrier ..."

Operations to manage the process

Concept comparison

 syntax-based approach

 semantically sound

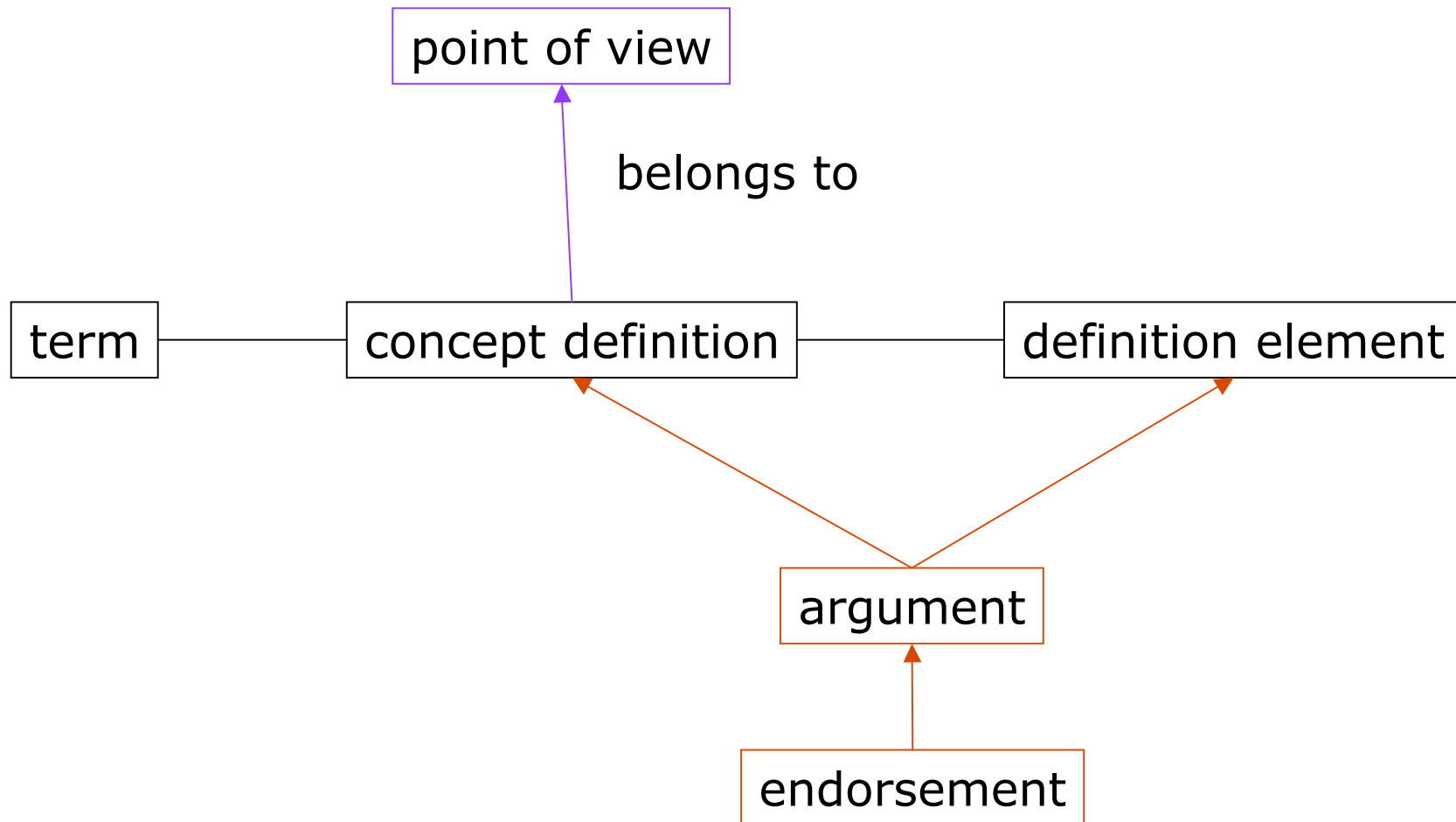
Incompatibility detection

Generation of new alternative definitions

 union, intersection, difference

Add an argumentation layer on top of the ontology

CTIBIS Model



The UNL Plaza

A portal to ontologies developed in several places

Part of the Universal Networking Language project

Goal: build a reference ontology for the UNL

Format: semantic network (60 types of relations)

Concepts designated by "universal words"

bank(icl>building)

bank(icl>organization)

bank(icl>shore)

⇒ language independent, but human readable

Language dictionaries

Universal Word ==> Natural Language Word

Distributed development

Top level ontology

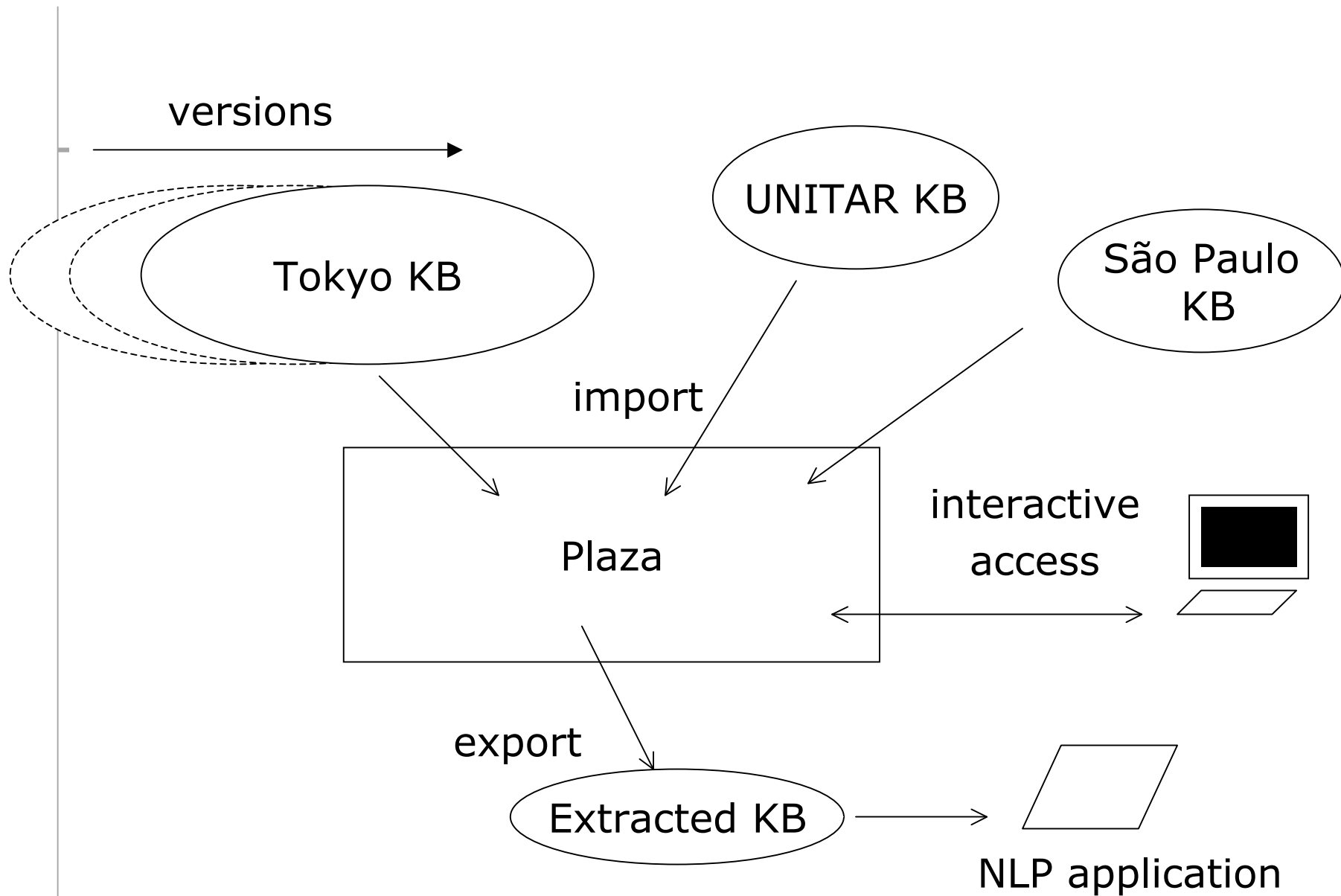
+ UW (re)defined for specific translation projects

The same UW can be defined several times
different point of views, different needs

UW may be updated and even deleted

==> Managing several point of views

==> Managing several versions



Managing point of views

Point of view == source

Each concept definition is tagged with its (creator/date)

one concept (UW) ---> 1 or more definitions

Semantics

references to UWs, not to definitions

to fix the semantics => define a selection order among creators

used to produce extracts (partial KBs)

Managing versions

Each definition has a version no.

Different kind of versions

- working

- stable (current)

- archived

- deprecated (the creator does not define this UW anymore)

Ontology storage and interfacing

Ontology representation in relational databases
simple

semantic networks (2 tables)

description logic definitions (3-4 tables)

Take advantage of DBMS functionalities

concurrency control, security, integrity

query languages

Integration with organizational data and documents

Tools to create basic Web interfaces

User interfaces

Very simple data structure (a few tables)

Complex contents (concept definitions, graphs)



"Standard" DB interfaces are insufficient, unusable

Hypertext view approach

- the user sees the ontology as a hypertext

 - easy to use / learn

 - reflect the connected nature of knowledge

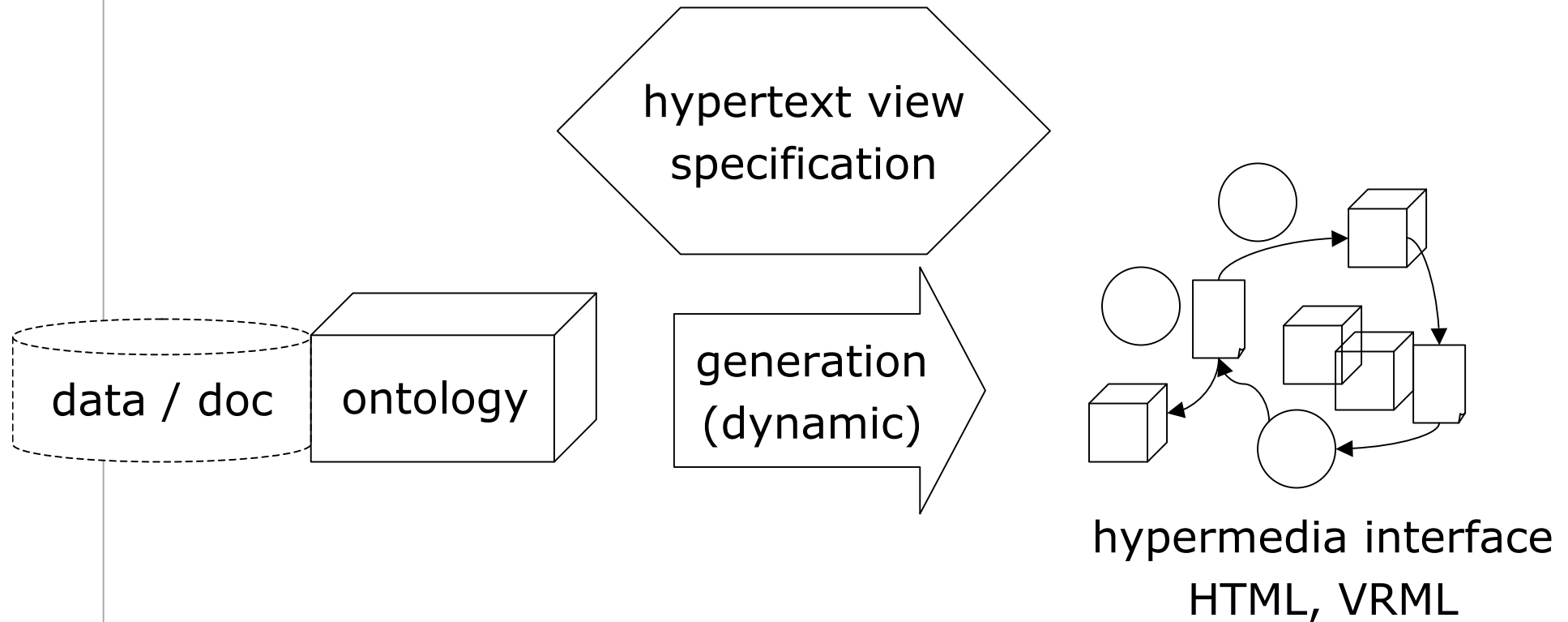
- integrate (multimedia) data and documents

- adapt to specific domains / tasks

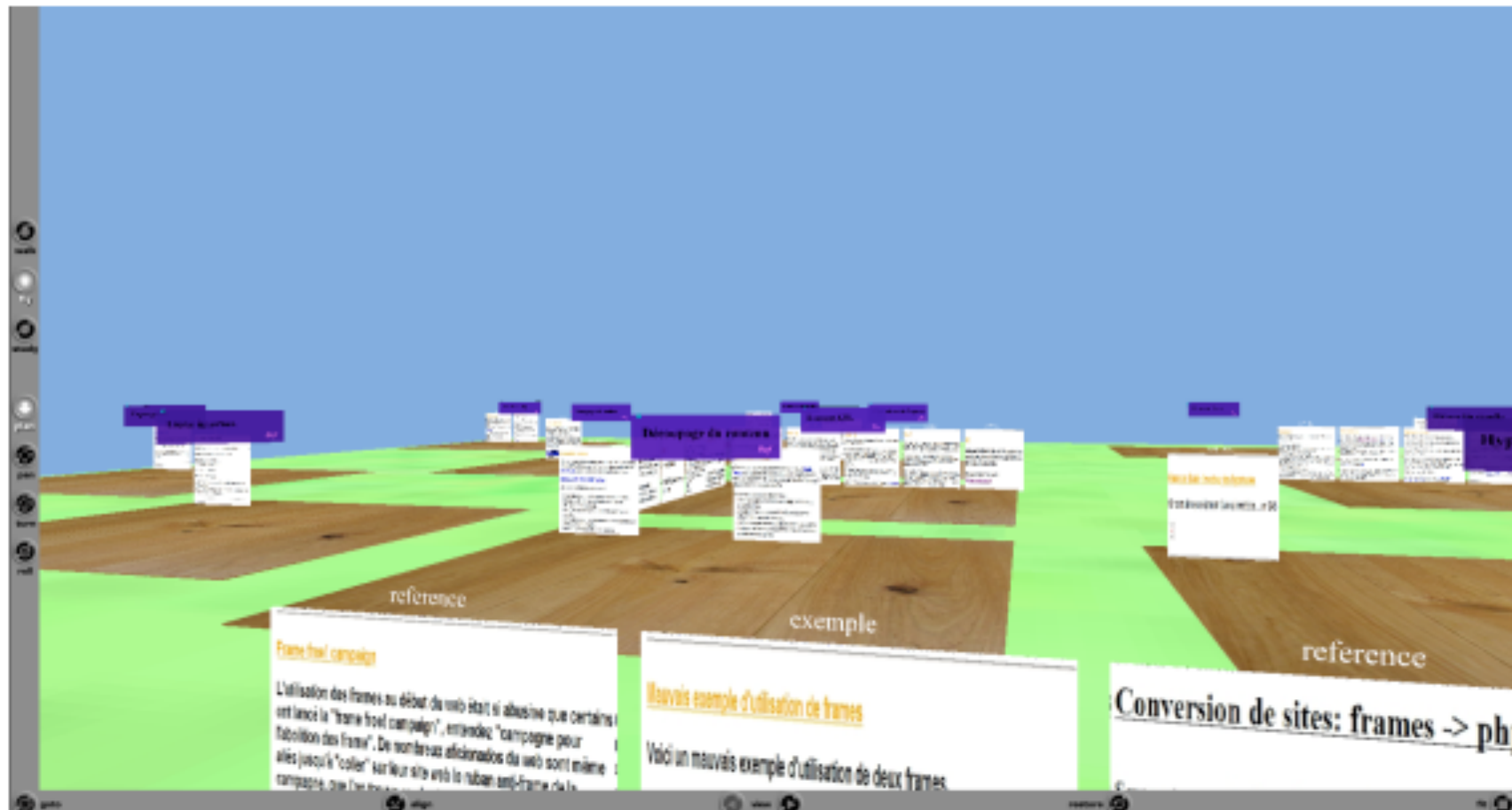
A specification approach to interfaces

Specify hypertext node and link types and how to create them from the ontology

Easy to adapt to various needs



Ontology + documents => exhibition



Beyond relational storage

Difficult to compute inferences

Relational DB languages are not Turing complete
storage is not efficient for this type of computation

Next step

use RDF/RDFS storage systems
automatic inference of properties (inheritance)
query with SPARQL (and others)

develop connections with instances stored in relational
databases

Future work

3D City Model based spatial hypertext

Hypertext nodes geometrically constrained by a 3D city model:

geometric position is **relative** to the city model

they can't be placed anywhere

the underlying city model used to specify their location

Examples:

on top of building A

on the south façade of building B

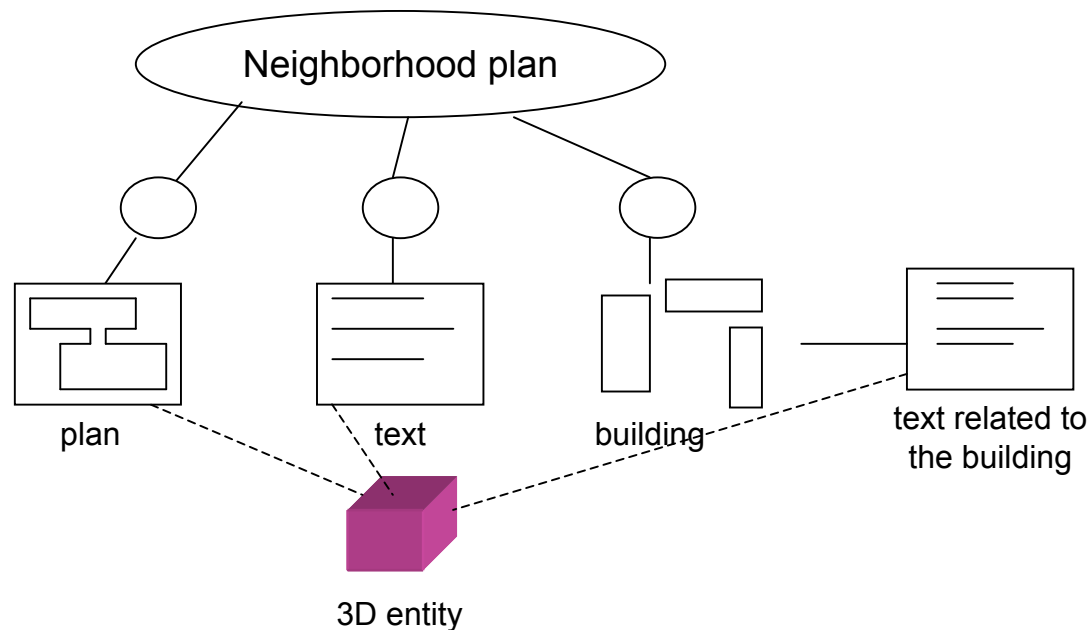
between C and D

A formalism to specify the 3D representation of urban concepts

In collaboration with urban actors

Not a fixed representation: related to the designer's choices

Define 3DCM layout managers



Implement an interface generator

Takes as input

- a 3D city model,
- a knowledge base
- a 3D representation specifications
- 3DCM-bases layout specifications

Produces as output

- a spatial hypertext
- constrained by the 3D city model
- presents the knowledge base according to
- the 3D specifications chosen by the designer

Project IUK3D

