From models to data: a prototype Query Translator for the cadastral domain

Bamberg, Workshop on Standardization in the Cadastral Domain, 9-10 December 2004
Claudia Hess and Marian de Vries
21 December 2004
Overview

- Introduction
- Information integration
- Query Translator prototype
- Evaluation
- Conclusions
- Recommendations
Introduction

- This paper
  - ‘Query Translator’ between cadastral data models
  - Practical tests with ‘real’ Dutch and Greek cadastral data (spatial and legal)
  - Use the core cadastral model as ‘mediator’ model
- Work in progress, first conclusions
- Aim: be able to access a cadastral database in another country with a query interface based on one’s own national system
Information integration

• One of the challenges of the European and global SDI
• Also true for the cadastral domain
• At present
  • No (easy) information exchange between national cadastral systems
  • E.g. finding the real estate of some person in another country, or his/her mortgage debts
Cadastral domain

- EULIS (web portal)
  - access to different cadastral registrations/databases ‘as they are’ (as first step)
- ArcCadastre (desktop application)
  - standardized storage model
- ‘Core cadastral domain model’ initiative (other presentations)
  - ‘Core’ information model for common classes, attributes etc. in cadastral systems
‘Pan’-European selection queries

- Ultimate goal: ‘parallel’ selection queries over different national cadastral databases from one (Web) application

  “select all real estate property owned by person with this name and birth date in Greece, Italy, Switzerland, Holland”
Overview

• Introduction
• Information integration
• Query Translator prototype
• Evaluation
• Conclusions
• Recommendations
Query Translator prototype
Query Translator prototype

- Translation between
  - ‘query model’ (as presented to user)
  - and ‘local’ storage model of data source
- Query model can be
  - The core cadastral model (one step)
  - or other national cadastral model (two step, via core model)
- Options
  - Only ‘lexical’ translation (synonyms)
  - Also structural translation (type/subtype, attributes, associations)
Uses cases / selection queries

- Simple case (class and attribute names, ‘lexical’)

Greek model
Select * from natural where lname = …

Dutch model
Select * from mo_subject
where gesl_naam = …
Use cases / selection queries

- More complicated (associations between classes)

```sql
select name, address, type_of_right
from person, right
where person.id = right.person_id
and municipality = ...
Define ‘equivalence’ between concepts in both models (with OWL)

- `<owl:Class rdf:about="greek_cad.owl#STATE_RESTRICTION">`  
  `<owl:equivalentClass rdf:resource="core_cad.owl#PublicRestriction" />`  
  `</owl:Class>`

- `<owl:DatatypeProperty rdf:about="greek_cad.owl#BENEFICIARY_BEN_ID">`  
  `<owl:equivalentProperty rdf:resource="core_cad.owl#Person_SubjID" />`  
  `</owl:DatatypeProperty>`
Query Translator prototype
Overview

- Introduction
- Information integration
- Query Translator prototype
- Evaluation
- Conclusions
- Recommendations
Evaluation

• Easy
  • Terms/concepts that are synonyms
• More difficult, but can be solved
  • Associations (→ joins between tables)
  • Different (super)type – subtype hierarchies
• Fundamental issues
  • System boundaries of the models
  • Conceptual versus Technical models
  • ‘Real word’ differences: Name, Address
Conclusions

• Core model offers a good starting point for cadastral data integration
• Get the priorities right
  • What are the most important selection attributes?
  • First concentrate on these classes and attributes
• There are ‘real world’ differences
  • Look at ‘localization’ theory and technology
  • Incorporate other standards (‘Address’)

21 December 2004
Recommendations

• Core model should also contain ‘administrative’ classes
  • Address, PostalAddress, VisitorsAddress
  • Name, NaturalPersonName, OrganizationName
• Or ... incorporate classes from other standards
  (European and global Address-standardization, Name conventions)
• These can be ‘Abstract’ classes without attributes as placeholders in model structure
Recommendations #2

• Not only standardize ‘structure’ but also ‘content’
  • ‘enumeration types’ for important selection attributes -> drop-down lists to assist user
  • ‘hidden’ subtypes can be found and solved this way
• Capture the legal process in UML or work flow language (e.g. property transactions)
• Not only test in countries, but also between countries