

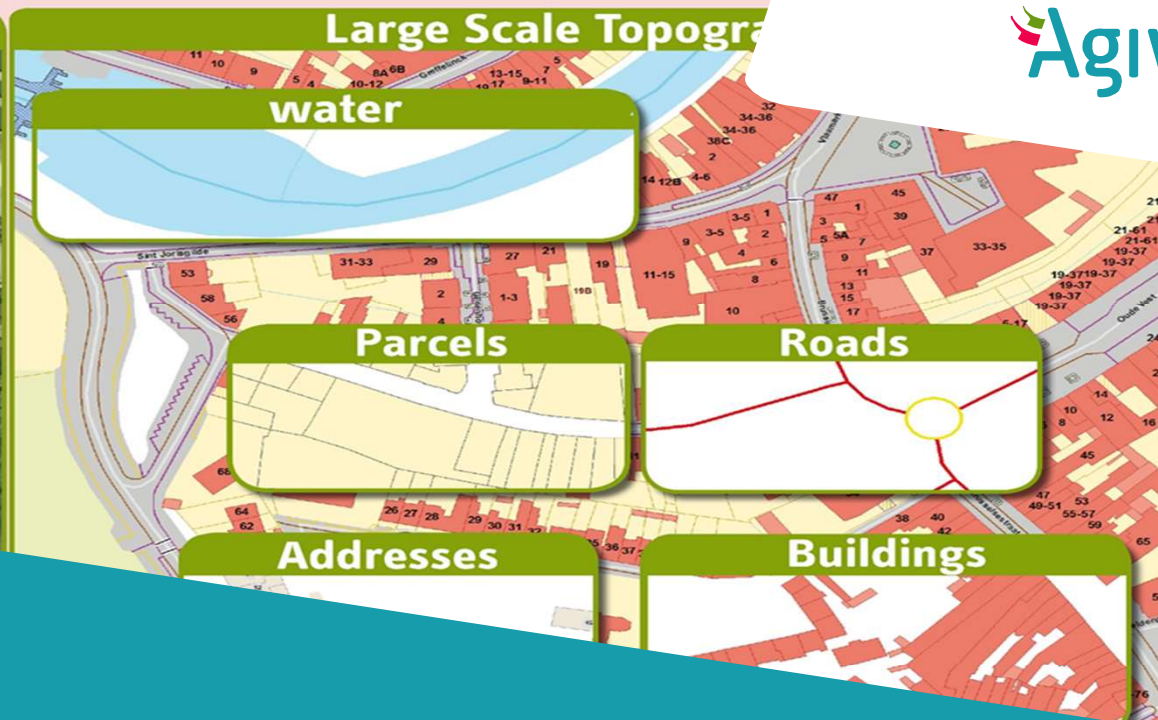
Basic Registries for Flanders



Terrain



Large Scale Topography



water



Parcels



Roads



Addresses



Buildings



Persons



Enterprises



Basic Registries for Flanders

Ziggy Vanlshout
Jef Vanbockryck

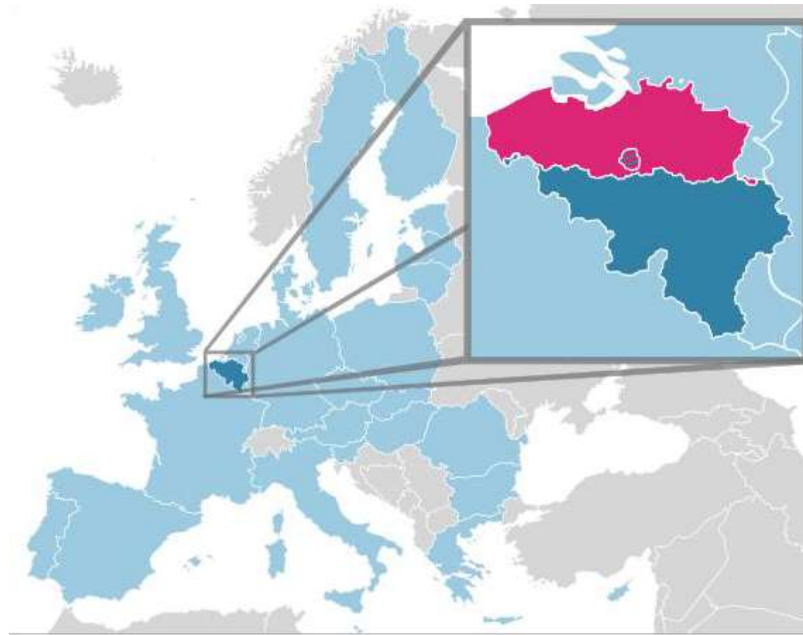
Flemish Geographical Information Agency
DMworkshop Copenhagen, 28–30 january 2015

Where is the what if the what is in how?

- Project 'Basic registries for Flanders'
 - FGIA
 - Authentic source data
 - Basic registries
- Model-driven architecture
 - Modeling rules
 - Modeling process
 - Methodology
 - Tools
- Challenges / questions to forum



SDI-Flanders - FGIA



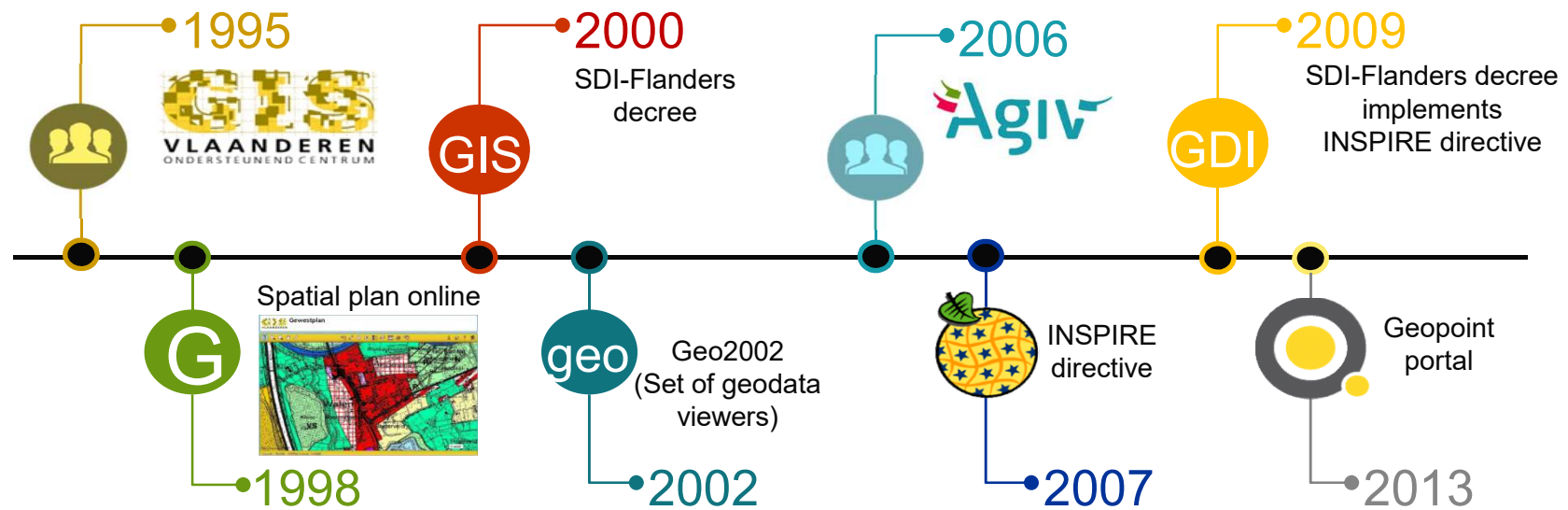
SDI-Flanders is a collaborative framework in order to develop and implement a sound communication and management system for geographical information in Flanders

Partnership between Flemish public authorities (local and regional)

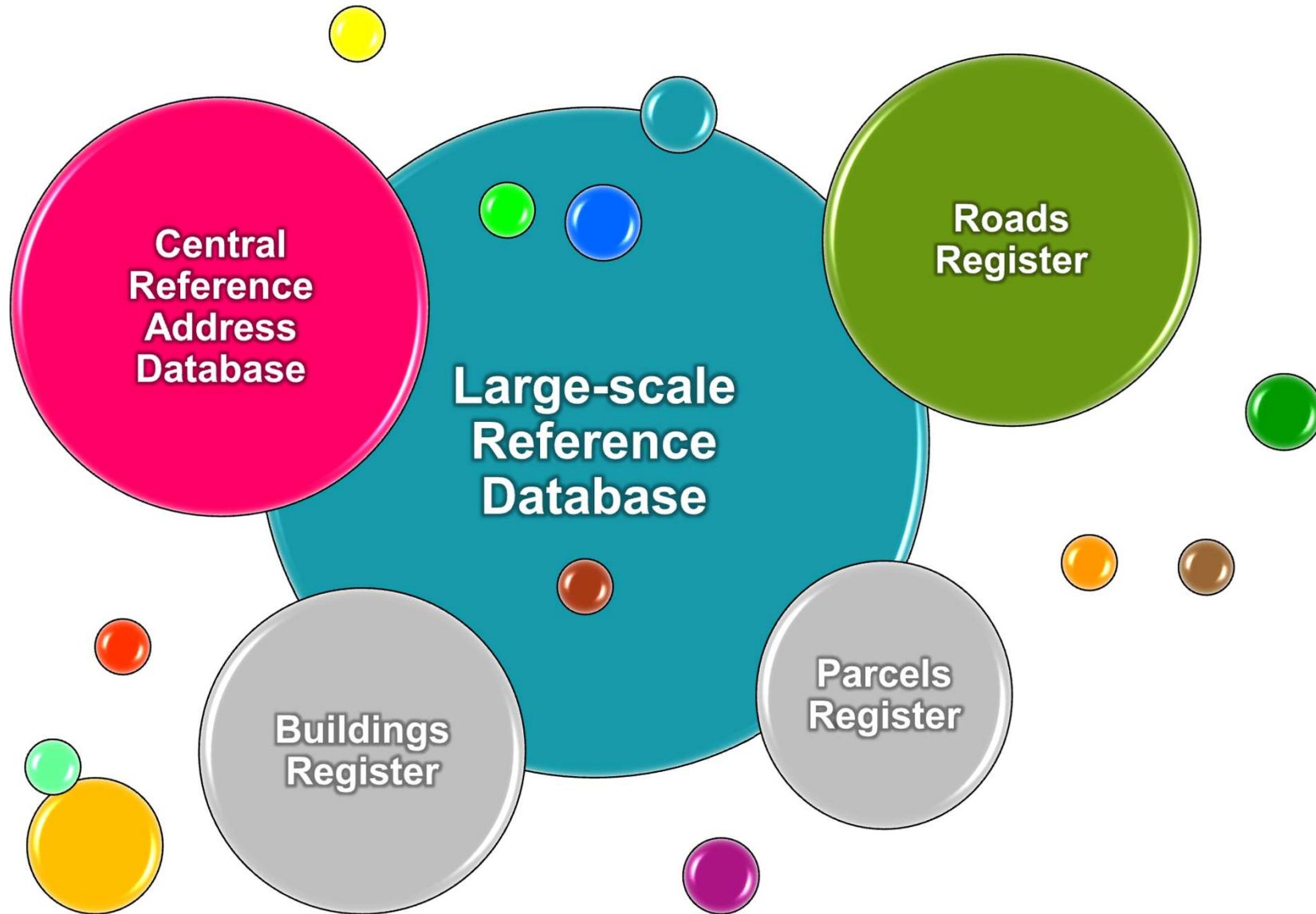
The Flemish Geographical Information Agency (FGIA) is the executive body of the SDI-Flanders

FGIA

- Centralised product and service supply
- Standardisation of products and services
- Cooperation with partners
- Partner support



Authoritative data FGIA



Large-Scale Reference Database (LRD)

- Start data collection 2000 - Full coverage since 2013
- Official recognition authentic data source 2015
- Free open data 2015



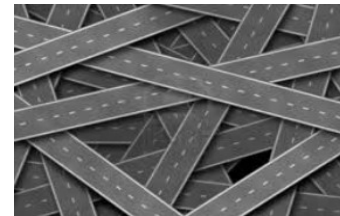
4.300.000

buildings



4.700.000

parcels



64.000

km roads



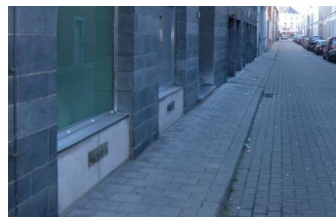
1.000.000

manhole covers



1.300.000

poles



40.000

km curbstones



20.000

km job canals



11.000

bus shelters

Central Reference Address Database



- Authentic data since 2011
- Use / error reporting compulsory for government institutions
- Decentralized maintenance by municipalities
- Free open data



Roads register

- 2014: first release
- 2015: second release (update)
- Validation by municipalities
- Update on the basis of
 - Address database
 - Large-scale Reference Database



Motivation for basic registries

Data operate as independent silos

- Confusing, complex
- Slow and inefficient
- Contradictory information

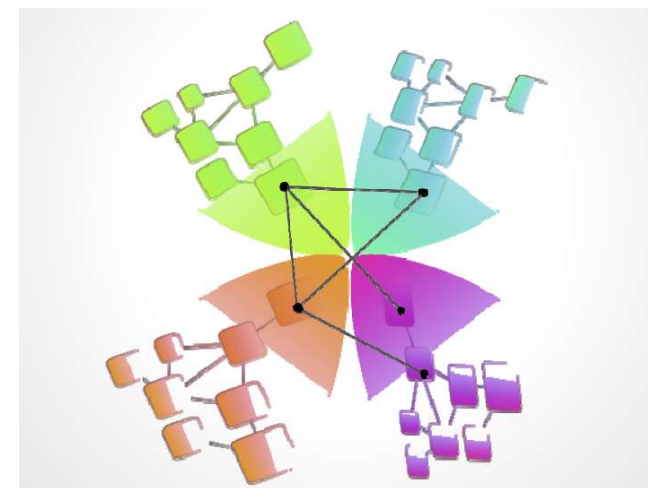
Need for a buildings register

- Building information in both CRAB and LRD

Need for 'linked data'

- Data work together
- Unambiguous objects and relations

Data harmonisation INSPIRE (2017)

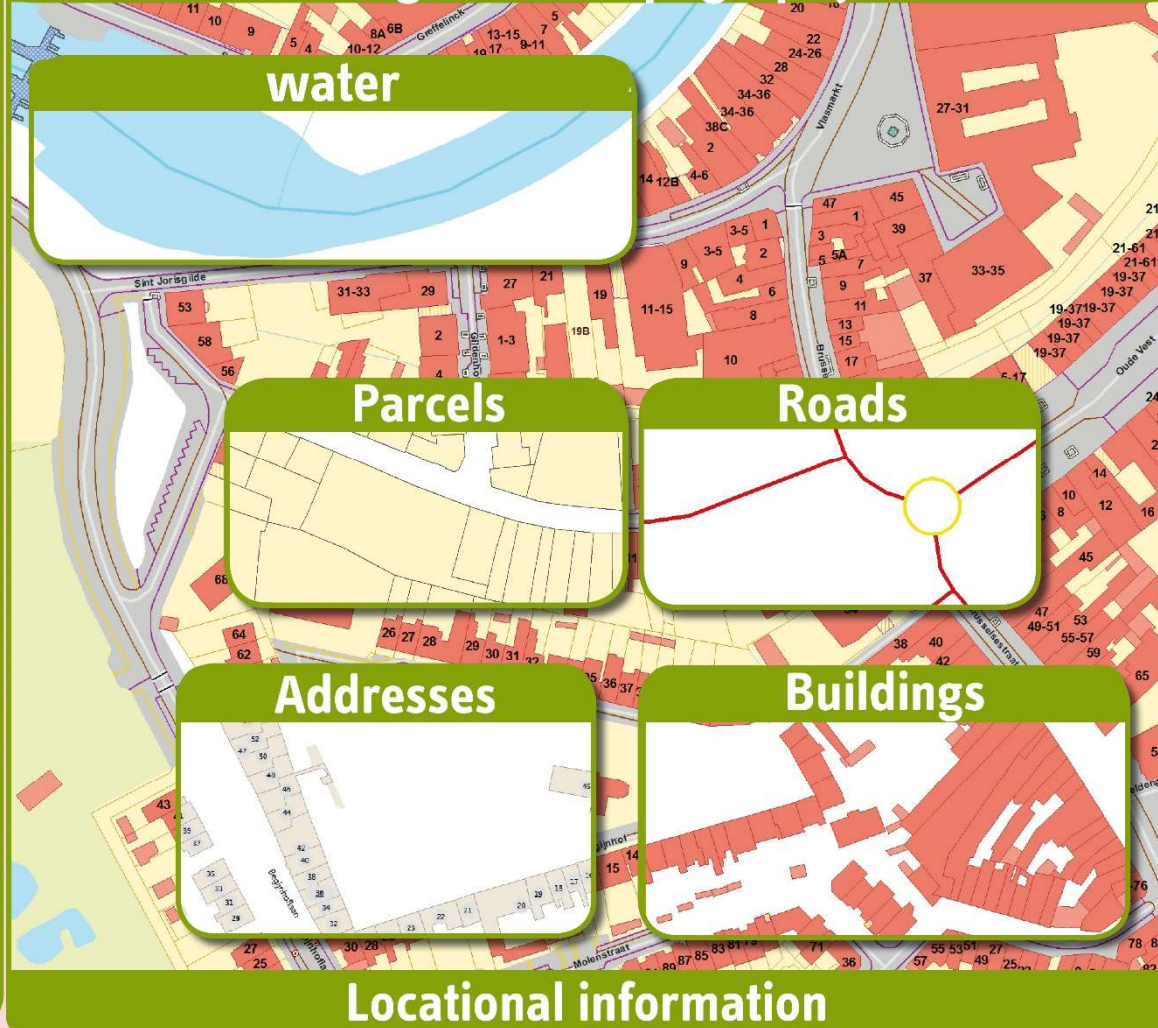


Basic Registries for Flanders

Terrain



Large Scale Topography



Persons

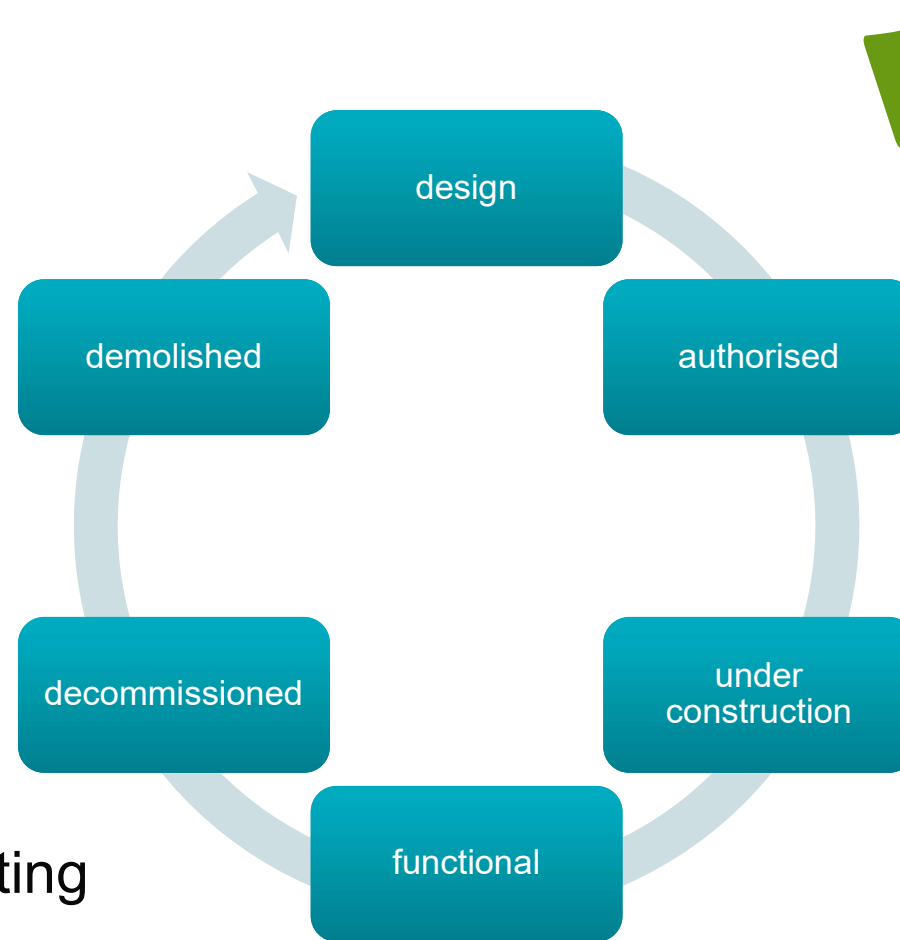


Enterprises

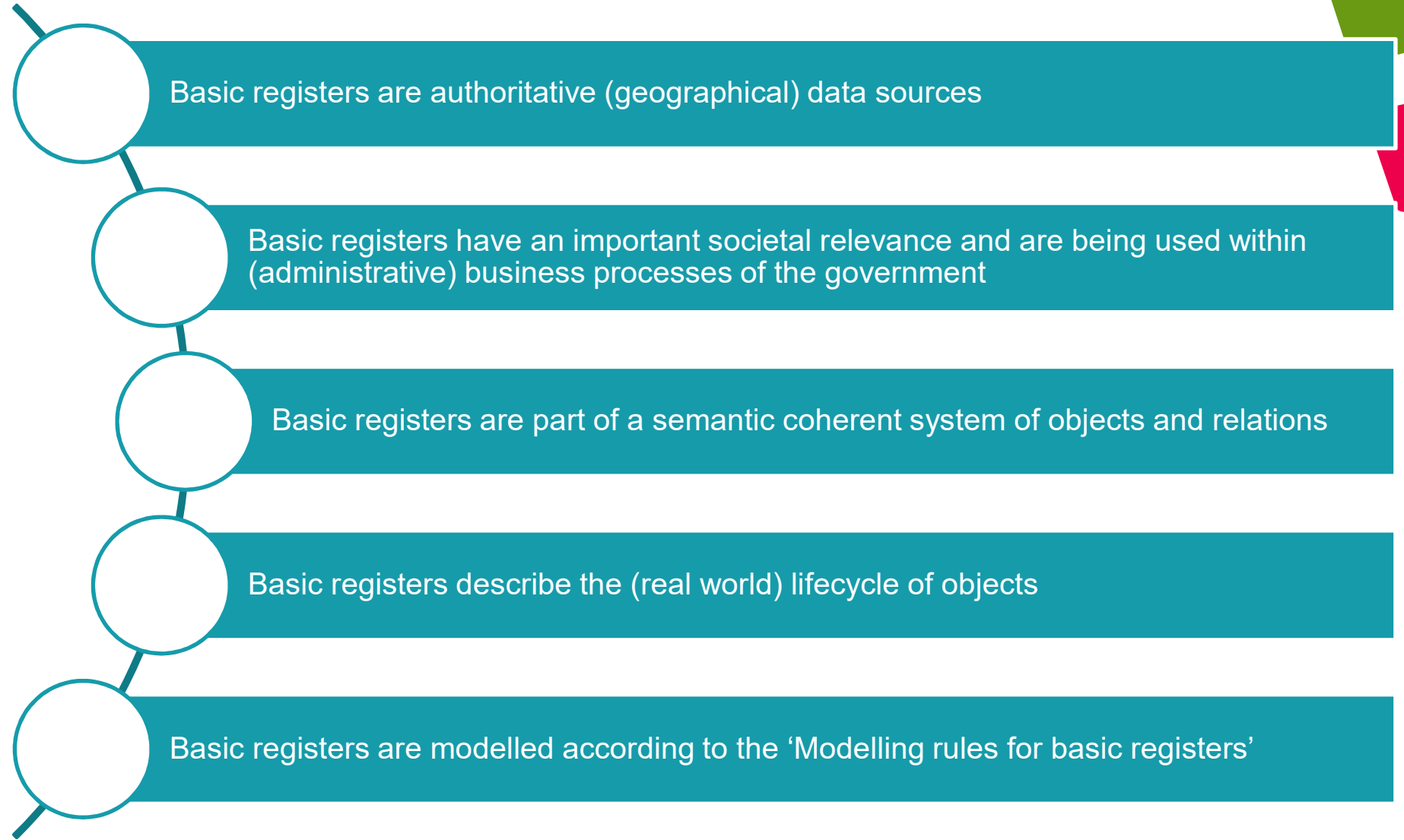


Concept

- Registers only contain core information
- Data in registers work together
- Persistent identifiers / lifecycle
- Combined centralised and decentralised data management
- Compulsory use and error reporting
- Free open data



Definition



Work packages

Business case

- Cost benefit analysis
- scenarios

Data model

- Uniform methodology
- Meets known user requirements
- Compatible with INSPIRE data specifications / ISA core

Architecture

- Implements the data model into databases
- Makes data operations manageable
- Is flexible and extensible

(De)centralised data management

- Centralised administration: based on generic FGIA-processes
- Decentralised administration: user friendly, one stop shop
- Organised for the whole system of basic registers
- Based on agreed upon business rules

Lego mixels



Information architecture based on a model-driven architecture (MDA)

- Applying “modeling rules” as basis for modeling basic data
- Step-wise and incremental modeling process
- Using INSPIRE UML Profile and tools
- For “management” of the basic data



Modeling rules

- “Inspired” by Danish “Modelregler for grunddata”
 - More or less equal document structure
 - Similar content, but different level of detail
 - Different approach (INSPIRE UML Profile used)

- Modeling rules are applied to modeling of basic data, but
 - Are also tested “during” the modeling process
 - Are modified if the need occurs from the modeling process
 - This is possible because FGIA is (currently) the only party using the modeling rules



Step-wise incremental modeling proces

First step: Model for Core Objects and Core Relations

- First test against known business processes
- Constraining them via rules:
 - Constraints are referenced inside the data model, but (currently) described in Excel

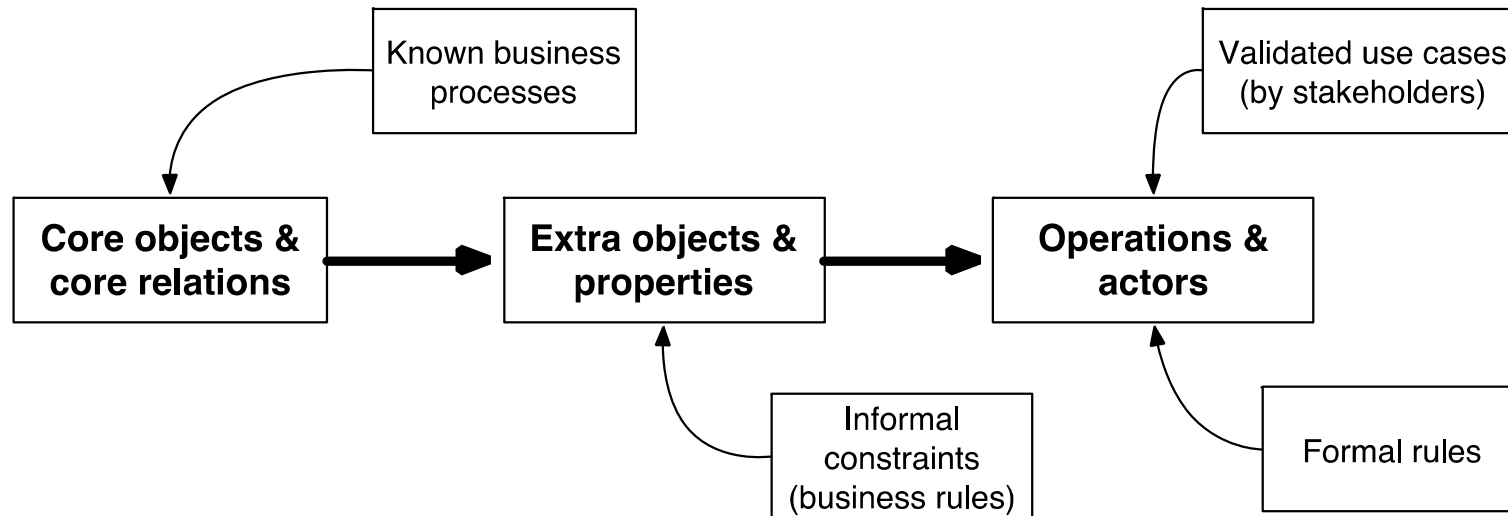
Second step: Adding just enough detail

- Essential properties and extra objects
- Description of identifiers and (real world) lifecycle of objects
- Validate against known use cases with stakeholders

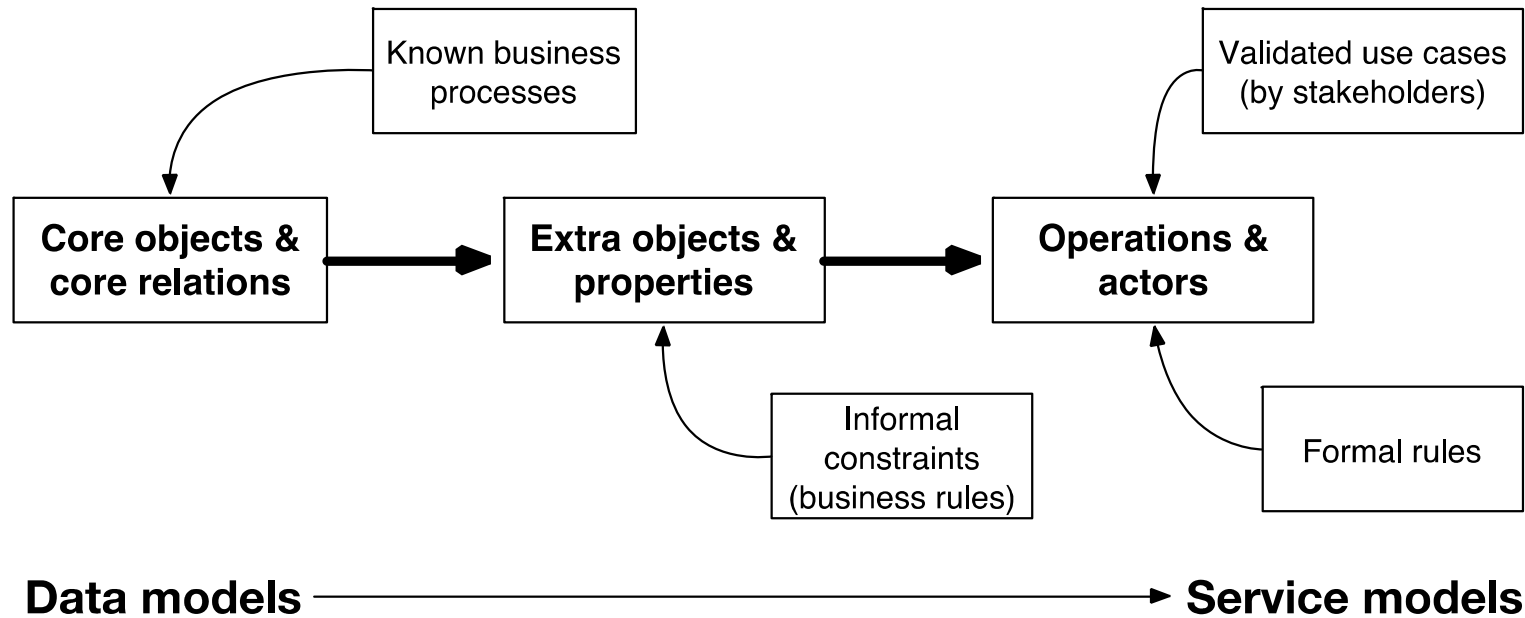
Third step: Adding more detail

- Operations and (potentially) actors
- Good enough to start “service modelling”

Step-wise incremental modeling process

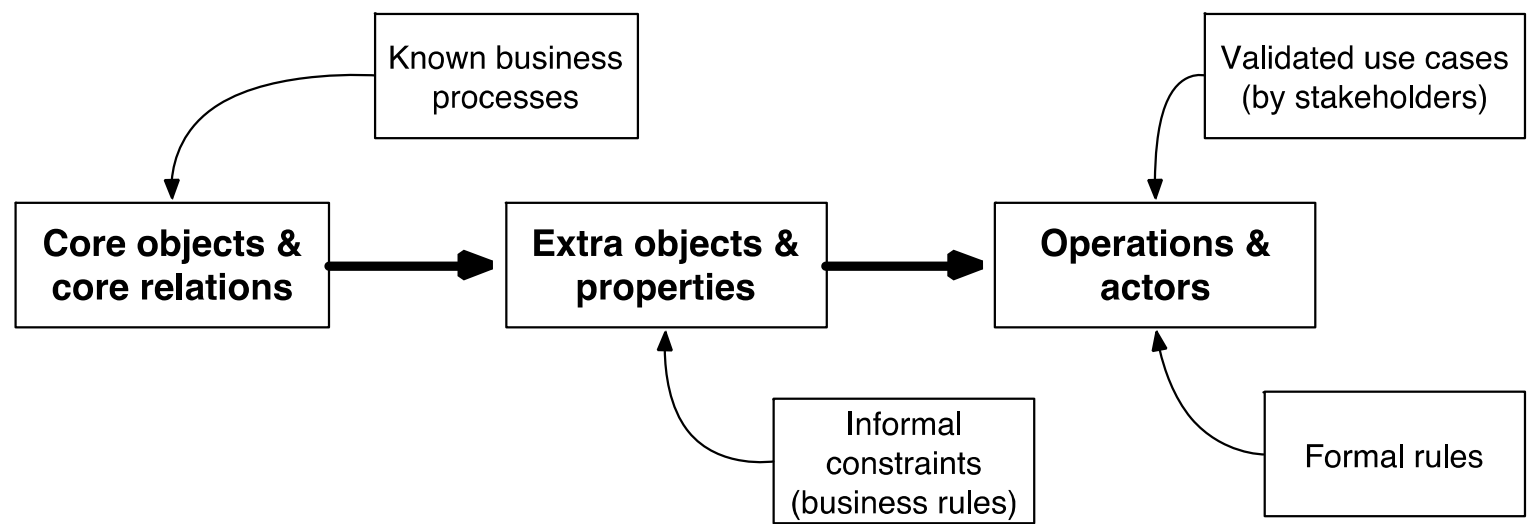


Step-wise incremental modeling process

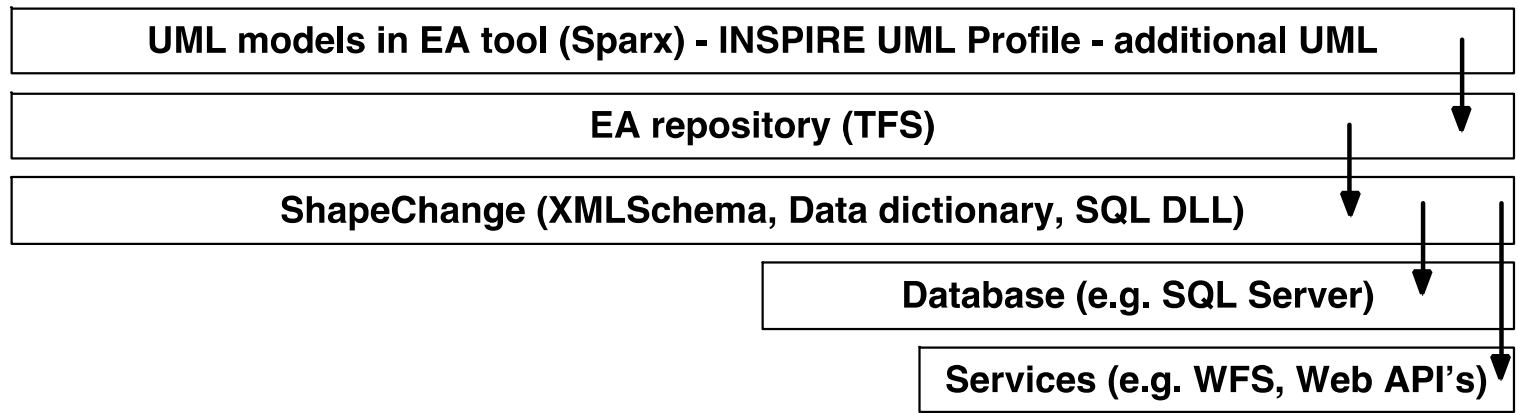


Iterative evolution from high-level data models, to detailed data models to service models

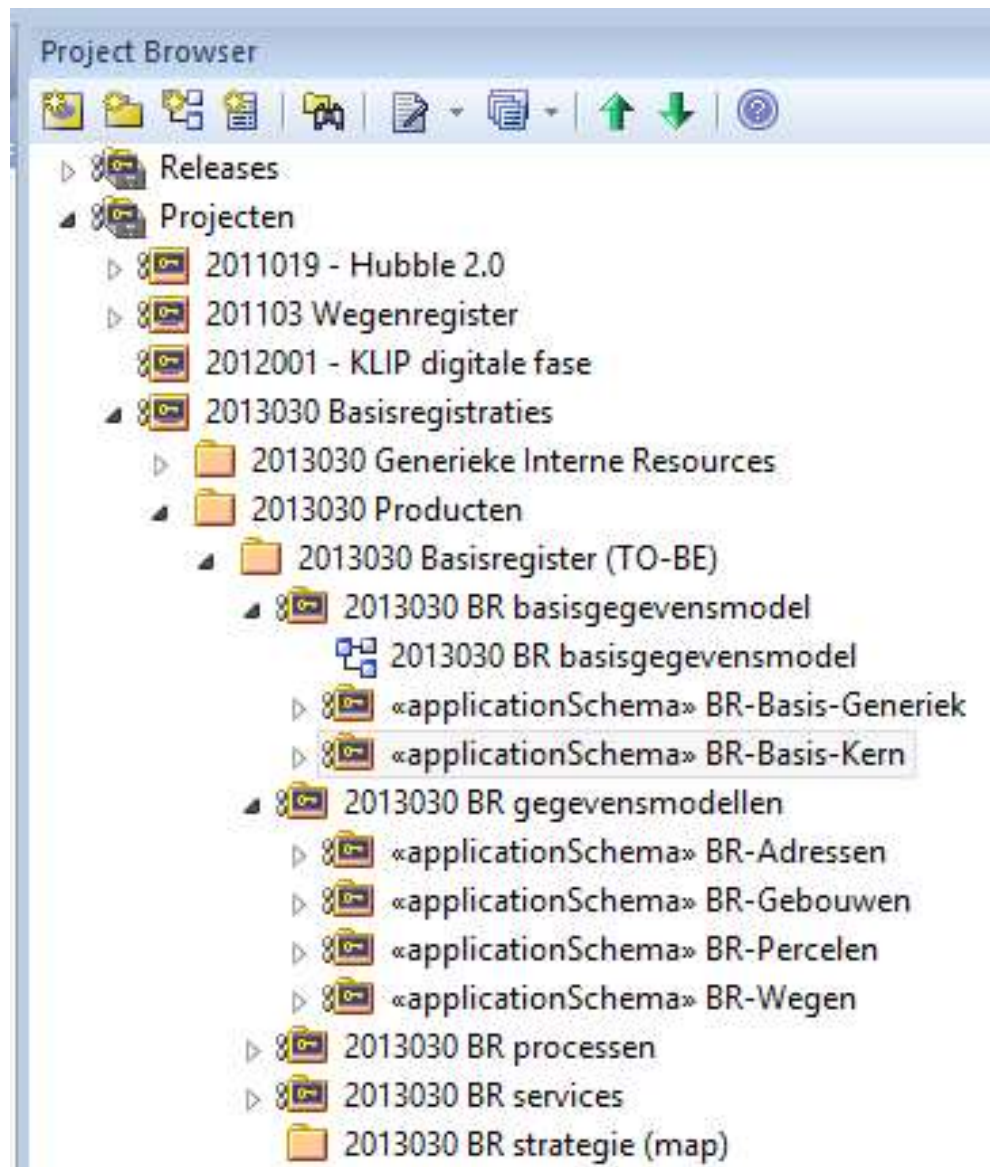
Using INSPIRE UML Profile & tools



Data models —————> **Service models**

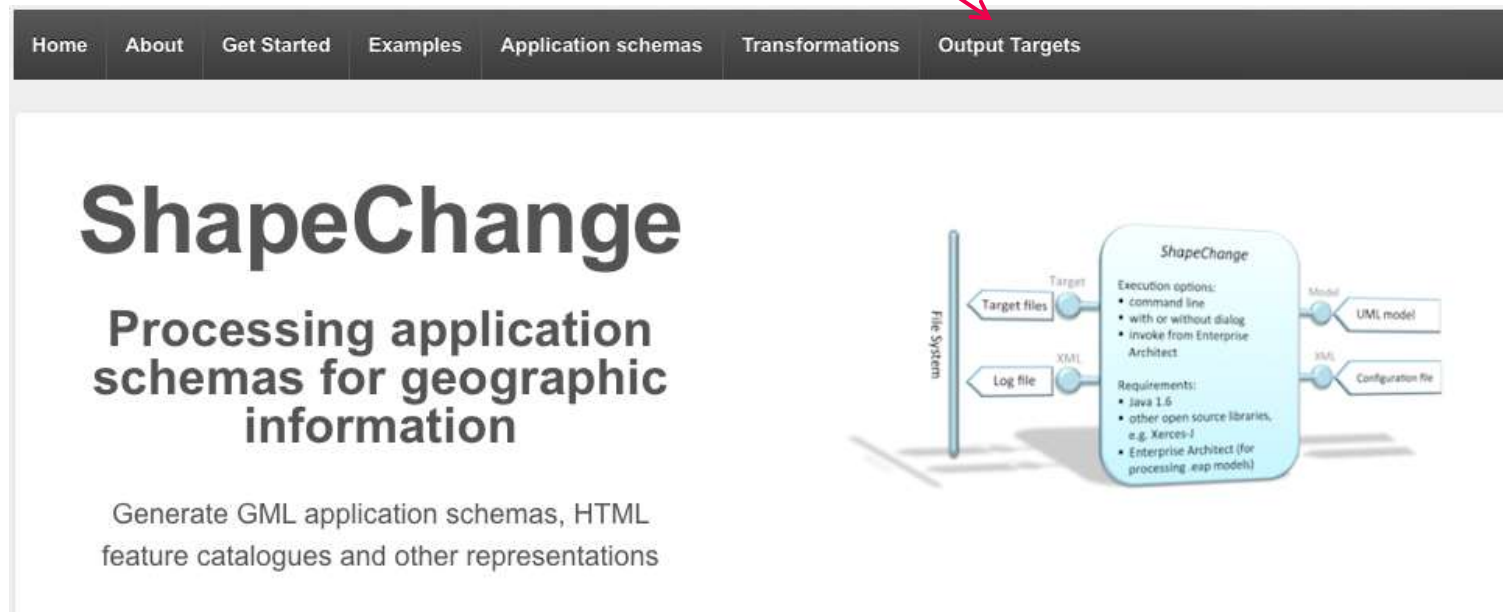


View on (TFS) repository connected to EA



ShapeChange as MDA transformer

- ShapeChange transforms UML data models (INSPIRE UML Profile) to:
 - XMLSchema
 - Feature (or object) catalogues
 - SQL DLL
 - JSON schema
 - RDF/OWL ontology
 - Etc.

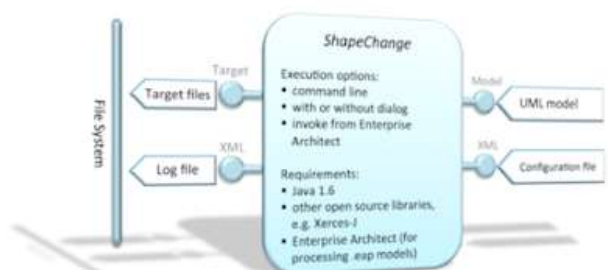


Home About Get Started Examples Application schemas Transformations Output Targets

ShapeChange

Processing application schemas for geographic information

Generate GML application schemas, HTML feature catalogues and other representations



File System

Target files Target

Log file XML

ShapeChange

Execution options:

- command line
- with or without dialog
- invoke from Enterprise Architect

Requirements:

- Java 1.5
- other open source libraries, e.g. Xerces-1
- Enterprise Architect (for processing .esp models)

Model UML model

XML Configuration file

“Management” of basic data

- Basic data models designed for management of basic data:
 - Additional metadata is added for management purposes
 - E.g. provenance
 - Concepts (and data dictionary) are in Dutch language
 - Facilitates support of decentralized data management (e.g. by municipalities)

- Our basic data models are not equal to “data products”:
 - Should be able to be “transformed” to
 - INSPIRE data specifications
 - POI’s
 - Open Linked Data (RDF, JSON-LD)
 - Legacy data formats (?)
 - ...



Our MDA in a diagram

Real world



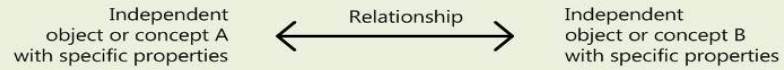
Modeled world



Registered world

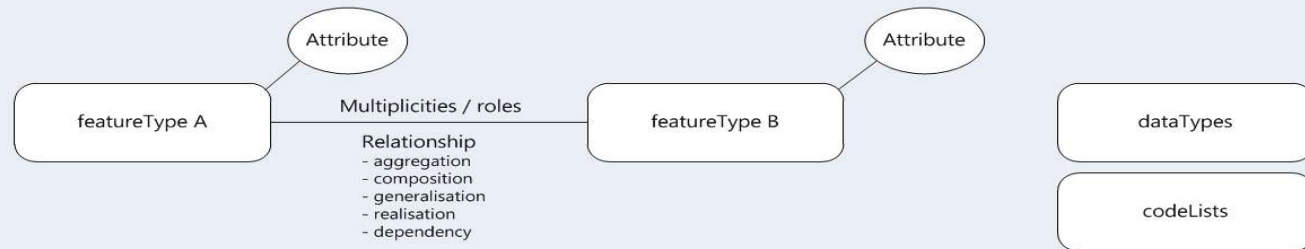
	wegsegmentID	beheerder	begintijd
1	1	71022	2014-02-20 14:35:32.237
2	2	-8	2014-02-20 14:35:32.237
3	3	-8	2014-02-20 14:35:32.237
4	4	AWV712	2014-02-20 14:35:32.237
5	5	71022	2014-02-20 14:35:32.237
6	6	AWV712	2014-02-20 14:35:32.237
7	7	71022	2014-02-20 14:35:32.237
8	8	AWV716	2014-02-20 14:35:32.237

Universe of discourse



rigorously described in

Conceptual schema



Unified Modeling Language (UML)

Modeling independent from chosen database model, technology or platform

metadata added

Application schema

UML package

Feature catalogue

converted using encoding rules

SQL DDL (database)

Tables
Primary/foreign keys
Indexes
Schemas
Datatypes
...

XML Schema

Namespaces
Declarations
- Elements
- Types

RDF

URIs
Statements
- Subject
- Predicate
- Object

...

Our MDA applied to basic data registers

Real world

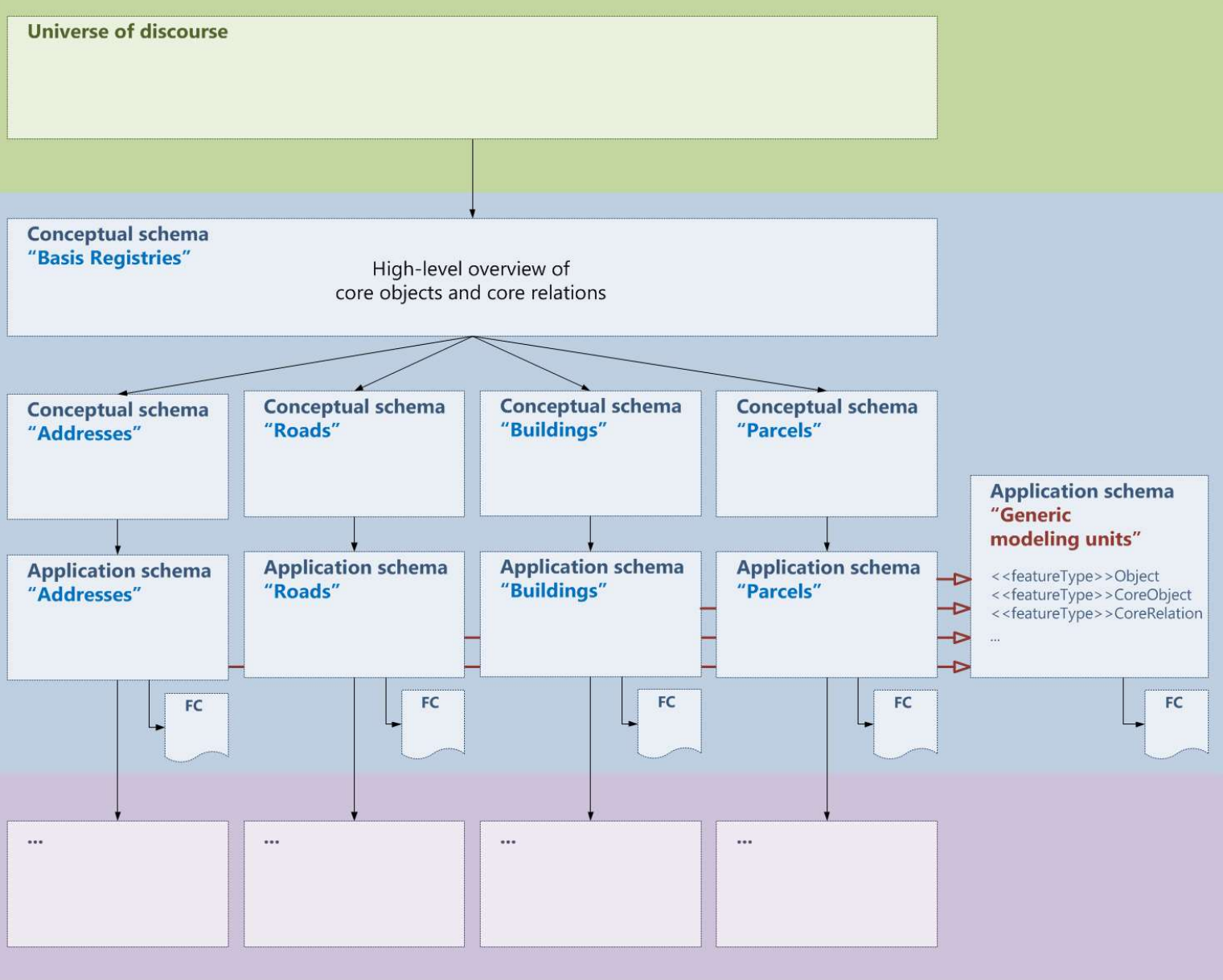


Modeled world

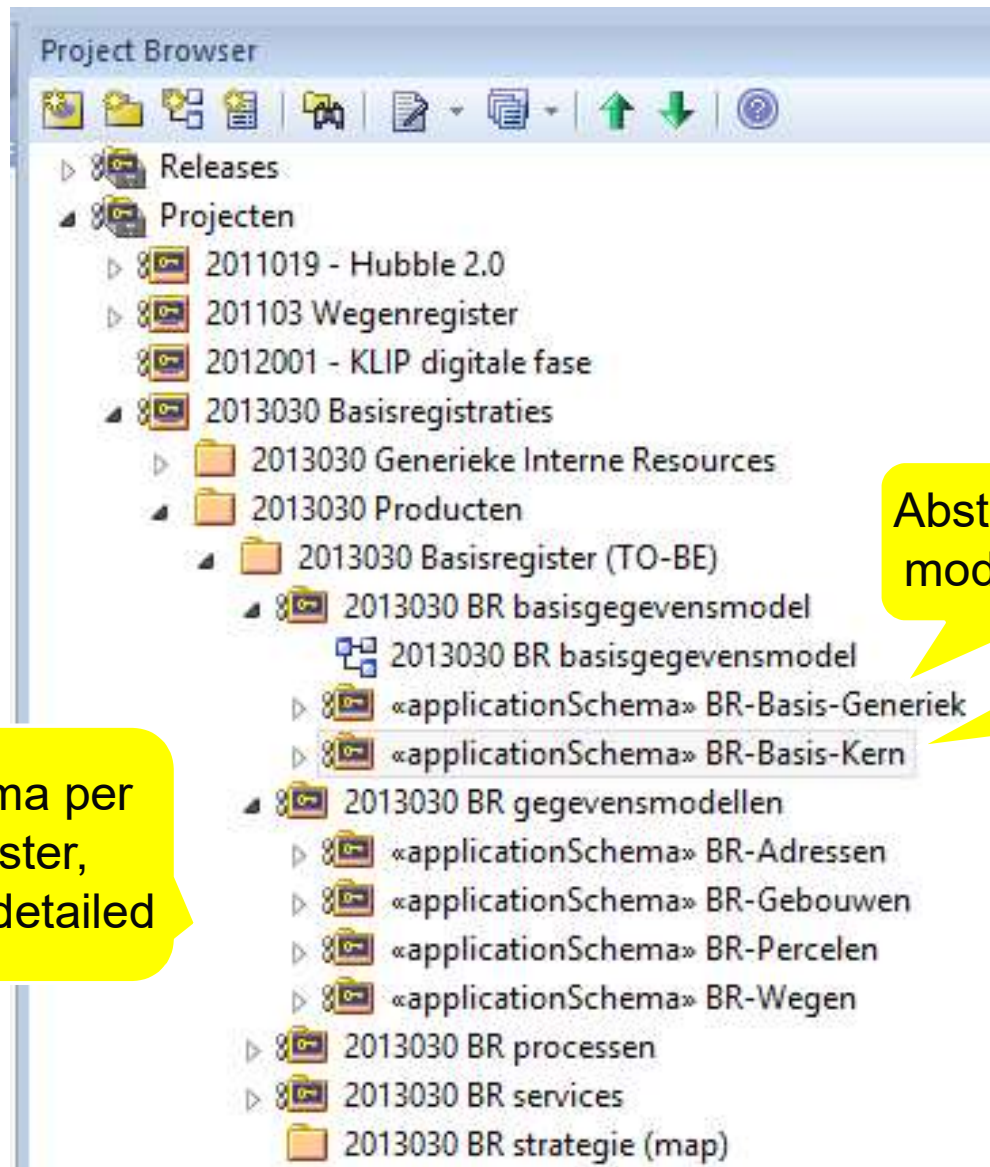


Registered world

wegsegmentID	beheerder	begintijd
1	1	71022
2	2	-8
3	3	-8
4	4	AWV712
5	5	71022
6	6	AWV712
7	7	71022
8	8	AWV716



View on (TFS) repository connected to EA



Schema per register, more detailed

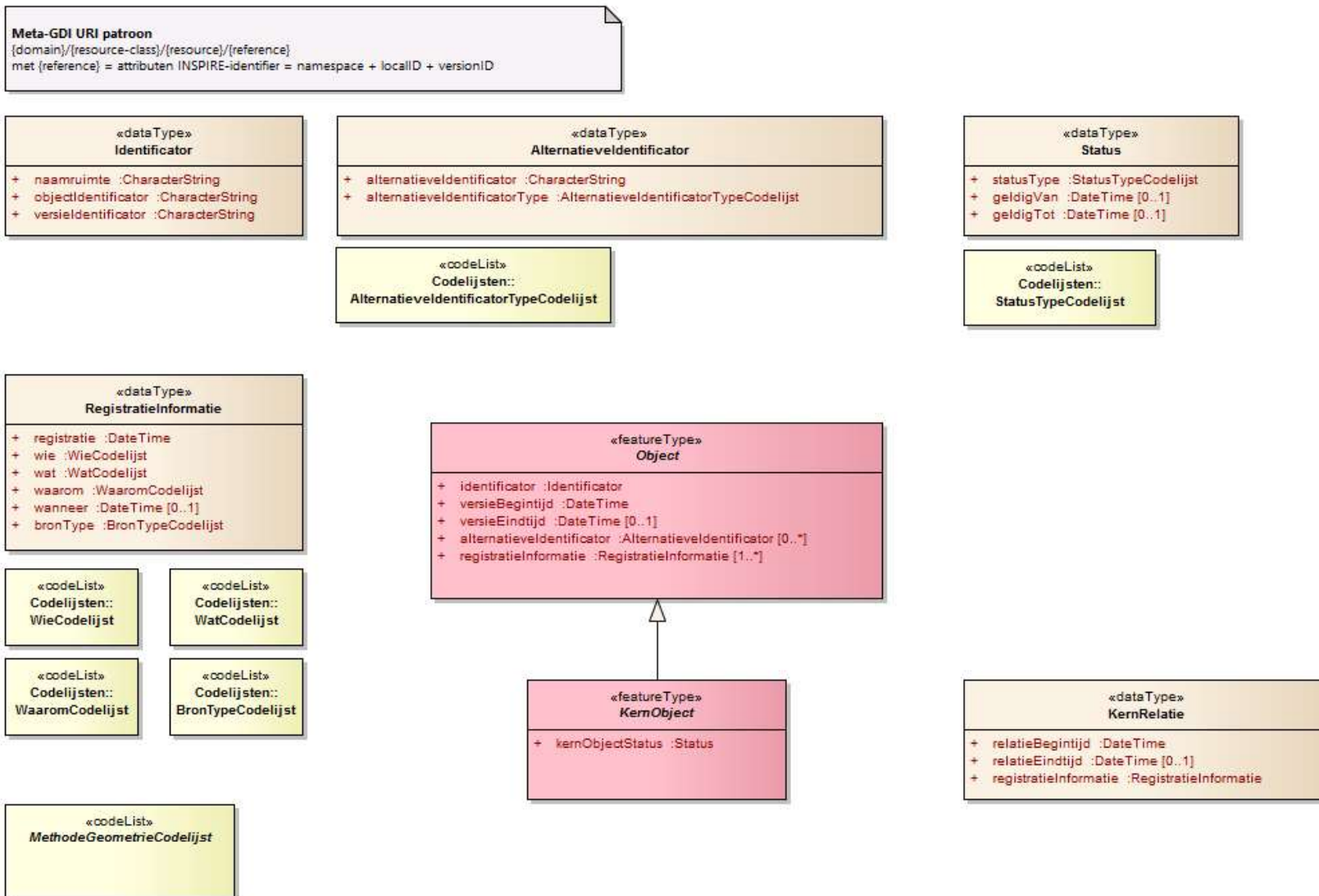
Abstract, generic model elements

Only core objects & core relations

Generic model elements

- Generic model elements are extended in concrete model elements:
 - **Object** or normal register object
 - **KernObject** or core object
 - **KernRelatie** or core relation
- These are abstract model elements containing common and essential properties for:
 - Identification, lifecycle mgt, provenance
 - Status management (only applicable to core objects)

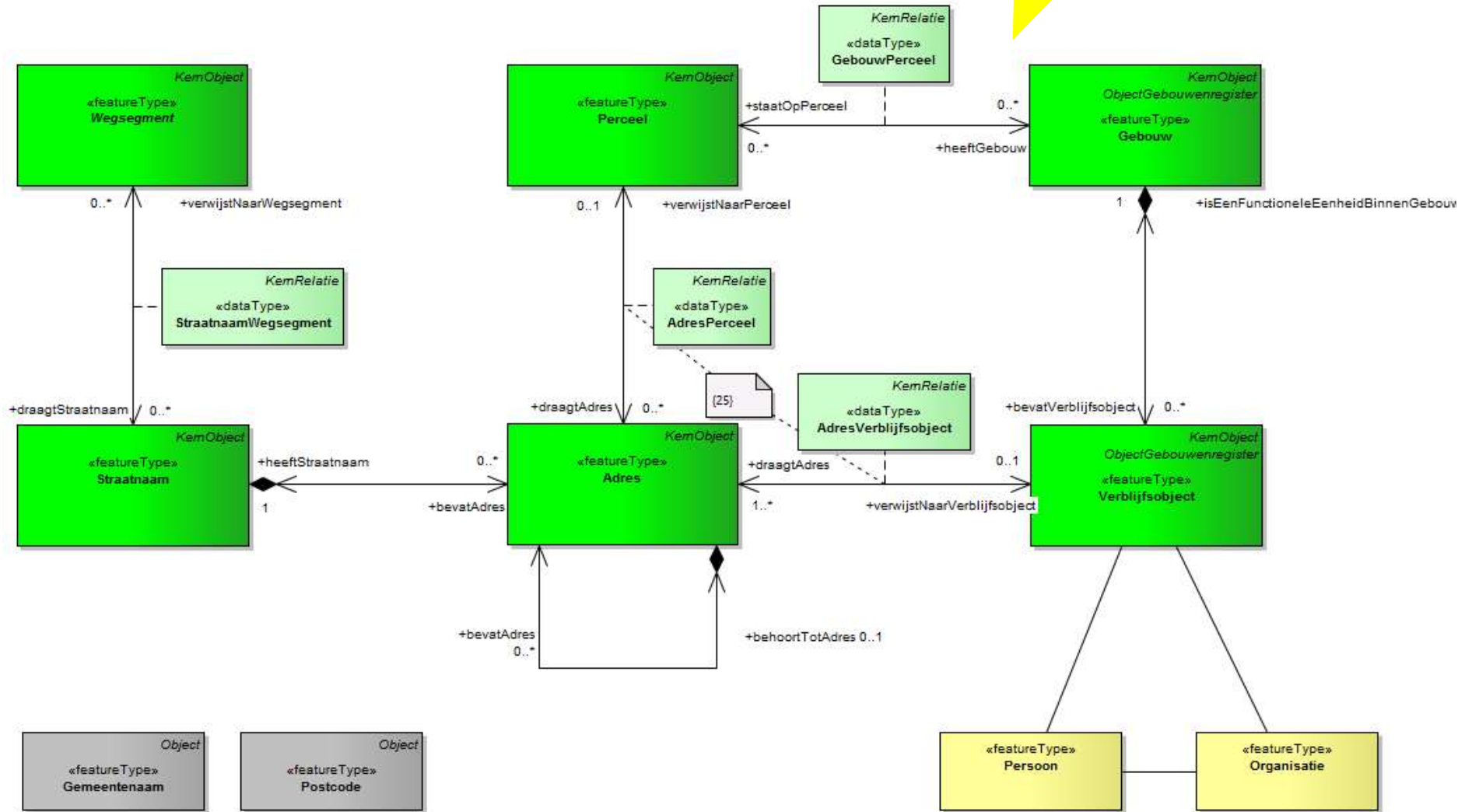
UML diagram of generic model elements



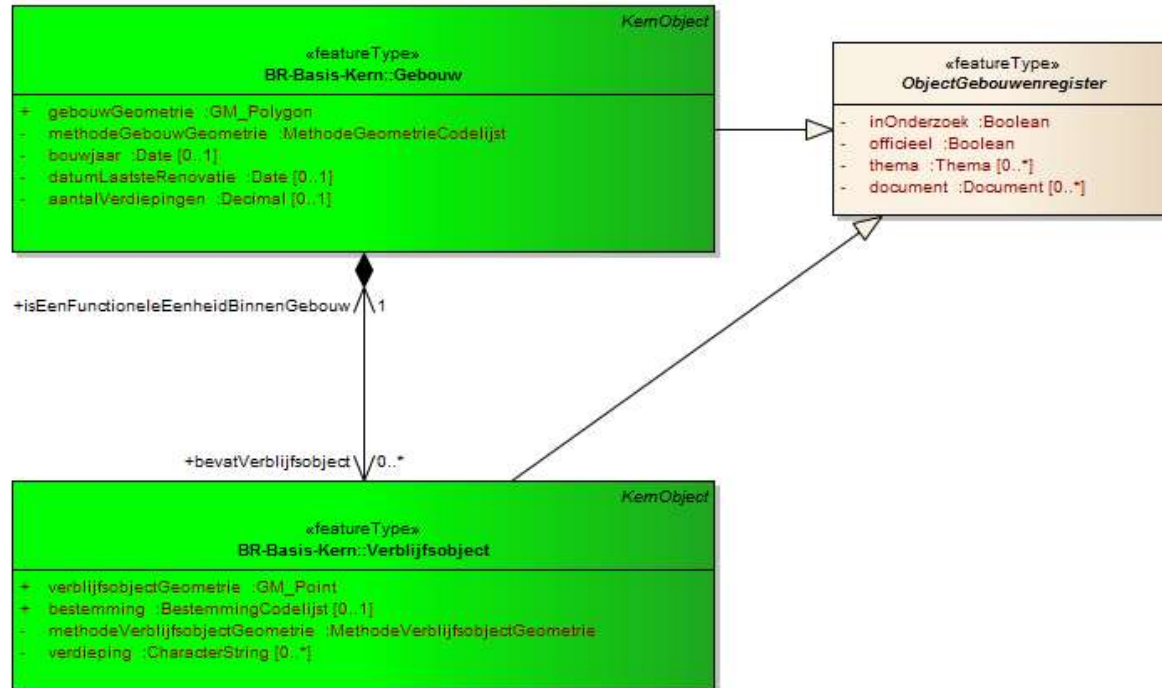
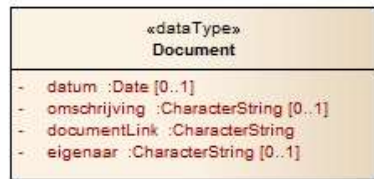
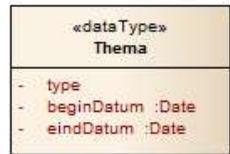
Basic data model

(Details are hidden)

Only core objects & core relations



Example: More detail in “Building” model



Schema per register, more detailed

Challenges / questions to forum

- *What about our approach of using « association classes » for modeling core relations (extension of the association class via generalization)?*
- *Should a data model for « management » of basic data have strong multiplicities? Should they be more loose and be managed via constraints/rules?*
- *What about our approach of this data model for management of basic data versus products (e.g. INSPIRE, ISA)?*

