General Indicator Modeling for Decision Support based on 3D city and landscape models using Model Driven Engineering

Mostafa Elfouly, Thomas H. Kolbe
Chair of Geoinformatics
Technische Universität München
mostafa.elfouly@tum.de
6th of June 2015
Measuring Landscape / City Performance

- Energy Indicators
- Ecological Indicators
- Mobility Indicators
- Financial Indicators
- Social Indicators

Evaluation is typically based on indicators, the most relevant are called Key Performance Indicators (KPIs)
Indicators

Energy Indicators
Mobility Indicators
Ecological Indicators
Social Indicators
Financial Indicators

Geobase data
CityGML Data
ALKIS Data
ATKIS Data
INSPIRE Data
BIM Data
Observations

1. Geobase data are available for entire countries and can be used for computing indicator values
   - (however, typically additional domain specific data are required)

2. All these geospatial information are based on standardised semantic data models / ontologies

3. So far, indicators are typically not formally modelled using a standardised framework

4. Furthermore, no systematic model exists yet for linking indicators and geobase data
Model Driven Engineering (MDE)

- ... is a software engineering paradigm which began to evolve in the 1980s

- MDE puts the “model” in the form of formal specifications in the centre of software analysis and design
  - Application relevant structures are represented by formal data models (e.g. using *Unified Modeling Language, UML*)
  - Program code is automatically derived from models

- MDE also addresses the linking of different models
  - This is called *Model Weaving*
  - Different models are linked by a *weaving model* which takes care of data transformation across the models
**Geospatial Information Modelling**

This is the general schema which all geospatial data models follow (e.g. ALKIS, INSPIRE, CityGML)

This is the data model of the 3D city model (here: CityGML)
It defines the structures of all possible 3D city models

3D city model data, e.g. the objects of the 3D city model of Berlin
Indicator Modelling

Domain specific indicators follow a General Ind. Model

These are the indicator models from different application domains

Concrete indicators for concrete city / landscape objects
Requirements for Indicator Models

- **Different types of indicators** need to be distinguished (i.e. numerical, textual, categorical indicators)

- **Complex indicators** can be composed & computed from
  - attribute values from associated city / landscape model objects
  - constants
  - mathematical expressions (unary / binary arithmetic operations) on other indicators

- **Indicator value aggregation** (e.g. summation, average, maximum, etc.) of other indicators

- Augment indicator values with **meta information** like accuracy, lineage / source etc.
  - allowing for **automatic sensitivity analysis**
General Indicator Model (GIM) in UML
Domain Specific Indicator Modelling

Domain Indicators | General Indicator Model

HeatDemand | Numeric Indicator
+ value

Where do I get the data from?
Domain of the stakeholder/application specialist

Energy Planner
Domain Specific Indicator Modelling

Many of the reference objects in the context of urban or landscape indicators are spatial objects.

Where do I get the data from?
Domain of the stakeholder/application specialist

Energy Planner
Linking Geospatial and Indicator Models

Geospatial Application Model (e.g. CityGML) -> Weaving Model -> Reference Objects -> Object Related Domain Indicators -> Domain Indicators -> General Indicator Model

- CityObject Group
- Building
- District Connector
- District
- EnergyDemand
- «Aggregation» num
- HeatDemand
- Numeric Indicator + value

OCL Rule 1
OCL Rule 2

Domain of the geodata provider
Domain of the stakeholder/application specialist

What can we do with our data?
Where do I get the data from?

City Modeler
Energy Planner
Rules in Object Constraint Language (OCL) - 1

Reference Objects

- District
  - refDistrict

- Building
  - refBuilding
  - refBuildingHeat
  - volume
  - TypeClass

Object Related Domain Indicators

- DistrictHeatEnergyDemand
  - refDistrict

- BuildingHeatEnergyDemand
  - refBuilding
  - refBuildingHeat
  - + compute()

Domain Indicators

- HeatDemand

General Indicator Model

- Numeric Indicator
  - + value

Context:

context BuildingHeatEnergyDemand inv:
  self.value = refBuilding.volume * 0.97

context DistrictHeatEnergyDemand inv:
  self.value = Sum(refDistrict.refBuilding.refBuildingHeatEnergyDemand.value)
Rules in Object Constraint Language (OCL) - 2

context BuildingConnector inv:
refBuilding.volume = refCityGMLBuilding.volume
Conclusions

► **General Indicator Model**: new framework for model based representation and automated computation of indicators

- Indicators for different domains are specified in a standardised and interoperable way using UML class diagrams and OCL rules
- Indicator models are linked to geobase data models using model weaving

► The framework facilitates

- systematic analysis of (also very complex) indicators and their relationships to digital landscape and city models
- representing and explaining key performance indicators for evaluation of landscape (aspects) represented by 3D models
- automatic derivation of programs to compute indicator values
Why KPIs

► A good indicator alerts you to a problem before it gets too bad and helps you recognize what needs to be done to fix the problem (Sustainable Measures).

► In order to support automated decision-making, the evaluation of different geodesign methods became a necessity.

► Evaluation and assessment of these models is to be considered from the basic usages during the geodesign process by using indicators for determining, monitoring, and detecting the impact of a specified change on a given model (Steinitz 2012).
KPIs Scope

► The KPI module can be applied to different domains (HED, Transportation, Climate, etc…). It’s not solely applied to one domain.

► This in turn drove us to the conclusion, General Indicator Model (GIM) should be at the meta-level bringing down the ideas/concepts of meta-level (M2) down to application-schema-level (M1).
Questions:

1. To which degree can Indicators be related to objects from $X$?
2. To which degree can Indicators be expressed in terms of objects from $X$?
3. To which degree can Indicators be computed from objects from $X$?
4. Can all the required data from $Y$ be taken from or derived from $X$?
5. Do the indicators refer to the same or similar concepts that are represented in $X$?
   (e.g. $X$:Building $\leftrightarrow Y$:BuildingHeatEnergyDemand)
6. Are the indicators and their relations of Y similarly structured than the represented concepts in X?
   (e.g. X:District $\diamondsuit$ X:Buildings $\leftrightarrow$ Y:DistrictHeatEnergyDemand = Sum(buildingHeatEnergyDemand))

7. What is missing in X with respect to the required information in Y?

8. Can X be augmented by the missing information and if yes; how?
   (e.g. ADEs, but also: Weaving Classes)

9. Which of the available geospatial application schemas is the most suitable regarding 1-8?

   How to measure this?!
Indicator Data Model linkage for different application domains

Indicator Model for Domain A

Object Related Domain Indicators

BuildingHeatEnergy Demand

DistricHeatEnergy Demand

Heat Demand

Domain Indicators

General Indicator Model

NumericIndicator

Reference Objects

District Population

Building Population

District

Building

Indicator Model for Domain B

Object Related Domain Indicators

DistrictPopulation

BuildingPopulation

Domain Indicators
Model Weaving Concept

- Reference Objects in different domains (Energy-Related Indicator, Climate-Related Indicator, etc…) can be connected to the different geospatial application models (e.g. CityGML, INSPIRE, etc…) using the model weaving concept.
Weaving Classes between different geospatial application models
The Structuring Information Framework using OCL for Heat Energy Demand Estimation
context BuildingHeatEnergyDemand
inv HeatLoss: self.HeatLoss=
{ refBuilding.ReductionNightWeekend *
(0.34 * refBuilding.EnergyEffectiveAirExchange * refBuilding.BuildingVolume * Building.Therma) *
(Sum( refBuilding.WindowsSolarIrradiationReductionFactor * refBuilding.ComponentSurfaceNumber * refBuilding.UValue ) * Building.Therma) }
)
)

context BuildingHeatEnergyDemand inv FreeHeat: self.FreeHeat=
{ 0.024 * refBuilding.SpecificHeatOutput * refBuilding.HeatingPeriod * refBuilding.EnergyReferenceArea }
)

context BuildingHeatEnergyDemand inv HeatDemand: self.HeatDemand=
{ (refBuilding.GlassSurfaceOverallEnergyTransmittance * refBuilding.ComponentSurfaceNumber * refBuilding.UValue ) * 
(Sum( refBuilding.TotalWindowSurface * 240 ) ) }

context DistrictHeatEnergyDemand inv: value = Sum(refDistrict.refBuilding.buildingHeat.value)

Object-Related Domain Indicators

Domain Indicators

General Indicator Model

06.06.2015 General Indicator Modeling for Decision Support based on 3D City & Landscape Models