INSPIRE application schemas COMPLEXITY

Agnieszka Chojka

University of Warmia and Mazury in Olsztyn

Geospatial World Forum, 25-29 May 2015, Lisbon Congress Center, Portugal

Application schemas

- **integral part of** INSPIRE data specifications
  - UML application schemas
  - GML application schemas

- define coherent and homogenous database structures
- worked out according to ISO 19100 series of International Standards in the geographic information domain
- allow to ensure the interoperability of spatial data sets

- some of them are very complex and interdependent

Interoperability in danger

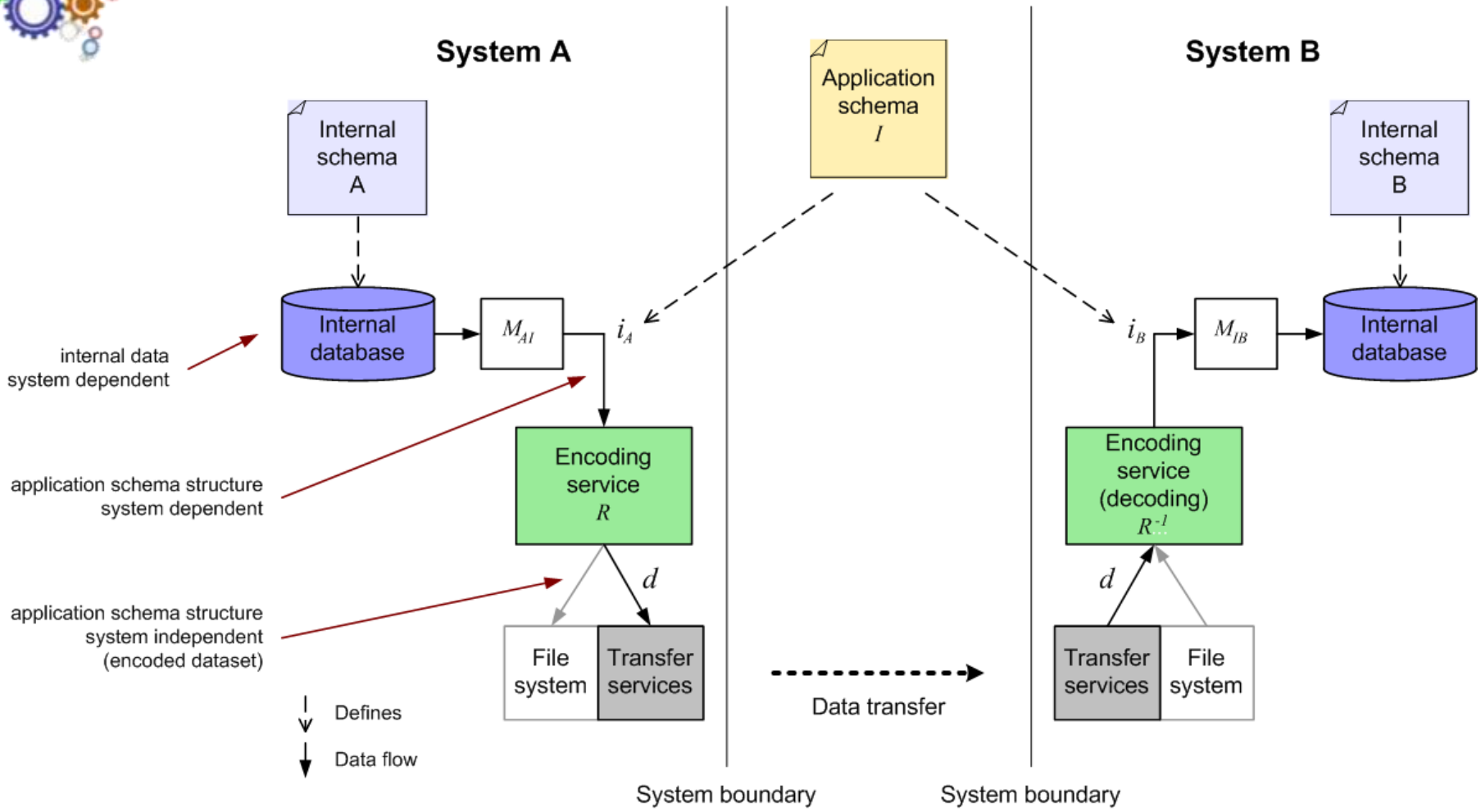
- **incorrect or too complex data structures**
  - have direct influence on the ability to generate GML data sets with concrete data (objects)
  - can cause various problems and anomalies
    - at the data production stage
    - during processing and operating GML data in GIS environments

- **solution**
  - measure application schemas complexity
    - propose their optimization and simplification
    - improve their quality and databases based on them

Why it's so important?

- application schema
  - **basis of successful data interchange**
  - **conceptual schema** for data required by one or more applications
    - formal description of a **conceptual model** in specified conceptual schema language
    - **model** that defines concepts of a universe of discourse (application domain)
    - **simplification** of relevant aspects of situation or object in the real world

Interoperable data exchange



[prCEN/TR 15449:2006; ISO 19118:2005]

Complexity measures

- **computer science**
  - **software metric**
    - measure of some property of a piece of software or its specifications
    - **structural complexity measure**
      - software quality estimation (final product)
      - complexity monitoring of all software components
        - e.g. system information model in the form of UML class diagram

UML complexity

- metrics for UML class diagram structural complexity
  - size metrics
  - structural complexity metrics

UML complexity

- **size metrics**
  - **NC** (number of classes)
  - **NA** (number of attributes)
  - **NM** (number of methods)

UML complexity

- **structural complexity metrics**
  - **NAssoc** (number of associations)
  - **NAgg** (number of aggregations)
  - **NDep** (number of dependencies)
  - **NGen** (number of generalisations)
  - **NGenH** (number of generalization hierarchies)
  - **AscNoRole** (associations without role)
  - **LoneClass** (lonely classes)

XML Schema complexity

- metrics for XML Schema complexity
  - XML-agnostic
  - XSD-agnostic
  - XSD-aware

XML Schema complexity

- **XML-agnostic**
  - do not consider any XML-related information

  - **KB** (file size in kilobytes)
  - **LOC** (lines of code)

XML Schema complexity

- **XSD-agnostic**
  - do not consider any information related with XML Schema, but use XML-related information
  - **#NODE** (number of all XML nodes (attributes and elements))
  - **#ANN** (number of all XML nodes for annotation)

XML Schema complexity

- **XSD-aware**
  - consider metrics concerned with schema information

  - **#El$_g$** (number of global element declarations)
  - **#CT$_g$** (number of global complex-type definitions)
  - **#ST$_g$** (number of global simple-type definitions)
  - **#MG$_g$** (number of global model-group definitions)
  - **#AG$_g$** (number of global attribute-group declarations)
  - **#AT$_g$** (number of global attribute declarations)
  - **#GLOBAL** (sum of all of above)

XML Schema complexity

- **$C(XSD) = C(V_g) + C(G_g) + C(T_g)$**
  - considers internal structure of XML schemas (not only counts schema components or features)
  - pays special attention to the use of recursive structures (as a source of complexity to schema users)

    - **$C(V_g)$** – total complexity values of all global elements and attributes that can be included/imported from external XSDs or can be declared/defined in the current XSD
    - **$C(G_g)$** – total complexity values of unreferenced global elements and attributes group that can be declared/defined in the current XSD
    - **$C(T_g)$** – total complexity values of unreferenced global complex and user-defined/built-in simple type definitions/declarations of XML Schema document

Software tools

You can't control what you can't measure
(DeMarco)

- **examples**
  - *SDMetrics* (UML)
  - *UML Metrics Producer* (UML)
  - *Castor* (XML Schema)
  - *GraphViz* (XML Schema)
  - … *GIS*
    - graphs
    - network analysis

Complexity analysis

- **assumptions**
  - simple application schemas selected
    - easy to prove that sth complex is really complex
  - 3.0 version of application schemas considered
  - ”foreign” classes not included
  - chosen complexity metrics
  - ”manual” analysis

UML complexity analysis

INSPIRE UML application schema	UML class diagram metrics				
	NC	NA	NAssoc	NAgg	NGen
Addresses	20	44	8	1	4
Administrative Units	8	30	4	1	0
Bio-geographical Regions	8	7	0	0	4
Cadastral Parcels	5	38	4	0	0
Geographical Names	9	23	0	0	0
Natural Risk Zones	22	52	5	0	12
Population Distribution	15	24	4	2	4
Protected Sites Simple	13	11	0	0	7
Species Distribution	20	30	2	1	0

GML complexity analysis

INSPIRE GML application schema	XML Schema metrics				
	KB	LOC	NODE	CTg	STg
Addresses	61,7	1039	86	26	0
Administrative Units	24,8	501	31	8	2
Bio-geographical Regions	5,46	129	10	2	0
Cadastral Parcels	31,2	661	44	8	0
Geographical Names	23,4	470	31	8	0
Natural Risk Zones	38,2	978	100	36	1
Population Distribution	17,4	450	37	10	0
Protected Sites Simple	11,4	220	13	4	1
Species Distribution	26,9	651	52	12	0

Conclusions

- **application schemas complexity results from**
  - wide thematic range
  - maybe ineffective database structure design
- **testing metrics**
  - not include e.g.
    - «voidable» (UML), ”nilReason” (GML)
    - abstract classes (UML, GML)
    - different geometry types (UML, GML)
    - attribute constraints (UML)
    - relations between application schemas (UML, GML)

Further challenges...

- **complexity examination** of some samples
  - GML data with concrete objects
- **verification** of application schemas complexity influence on data quality (including data complexity)
- **elaboration** of some original complexity metrics
  - adjusted to INSPIRE application schemas
- **testing** of GIS functionality to measure application schemas complexity
  - implementation of own tool alternatively

Thank you for your attention!!!

- **PhD Agnieszka Chojka**
  agnieszka.chojka@uwm.edu.pl

  Department of Land Surveying and Geomatics
  Faculty of Geodesy, Geospatial and Civil Engineering
  University of Warmia and Mazury in Olsztyn