Separate Conceptual Model and Encoding(s)

SIG 3D – OGC International Workshop on CityGML 3.0

Technical University Munich – 20-21 June 2013

Steve Smyth – OpenSitePlan + CityGML SWG Co-Chair

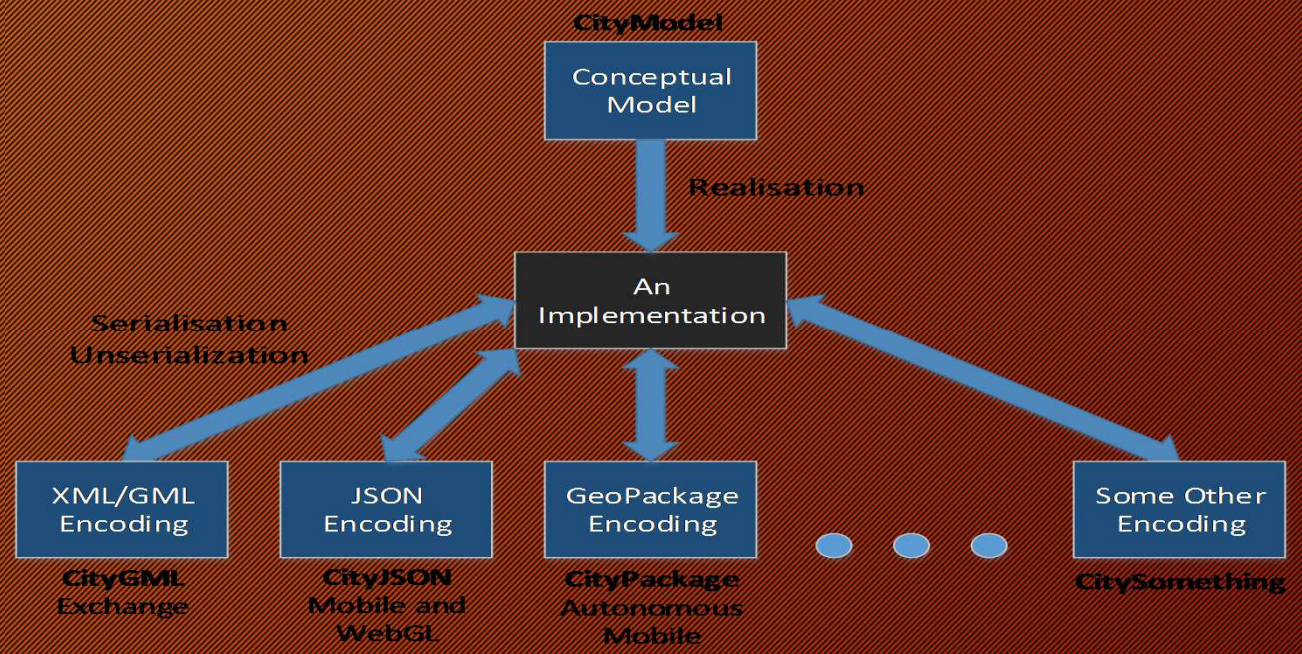
The Proposal

- The underlying conceptual model should be defined by version 3.0 in a way that is independent of any specific realization (e.g. using UML).
- Implementations of the version 3.0 conceptual model should have internal representations that are at least as expressive as and conformant with the conceptual model (testable via serialized artifacts).
- Implementations of the version 3.0 conceptual model should have serialization/deserialization methods that produce or consume encodings according to version 3.0 or later encoding specifications that relate constructs in the conceptual model to constructs in each supported encoding (e.g. JSON, GeoPackage, GML).

Background

- Definition of the CityGML conceptual model is shared between a set of XML Schema Language (XSD) files and UML model diagrams.
- The superiority of one or the other of the XSD and XML definitions is not clear and some adopters of CityGML have picked one as fundamental and some have chosen the other.
- A strong argument can be made that there should be one fundamental definition, that it should not be expressed in terms of a particular encoding, and that there are significant advantages to a separation of the conceptual model from its realization in specific encodings.

Multiple Encodings



Argument 1

There should be a single “root”, “parent”, or “base” conceptual “CityModel” definition because the existence of multiple definitions, even if believed to be equivalent, will always have differences in practice and lead to problems with interoperability and create market confusion.

Argument 2

- The primary problem with XSD as a definition of the CityModel arises when references are made to external schemata, and those schemata have types and relationships that are not essential to CityModel. These non-essential parts contaminate the definition with extraneous information. This extraneous information cannot be ignored because there is no automated mechanism to prune branches of an external schema (i.e. make a profile).
- A secondary problem is that reliance on a definition that is not encoding-neutral makes definition and implementation of alternate encodings difficult or impossible.

Argument 3

There are many practical advantages for development of additional encodings beyond GML, including the new OGC GeoPackage and JSON/GeoJSON.

- GeoPackage is based on SQLite, targets smartphones, and is usable as a runtime format for incident and disaster response. It supports autonomous as well as connected devices.
- JSON is a web-friendly delivery format that facilitates development of JavaScript applications, especially those targeting WebGL.

Opening up these and other possibilities for multiple encodings would enhance both the market appeal and the breadth of use of the CityGML conceptual model.

Conclusion

- The underlying conceptual model should be defined in a way that is independent of any specific encoded realization (e.g. using UML).
- Implementations of the conceptual model should have internal representations that are equivalent to those of the conceptual model.
- Implementations of the conceptual model should have serialization/deserialization methods that produce or consume encodings according to encoding specifications that relate constructs in the conceptual model to constructs in each supported encoding (e.g. JSON, GeoPackage, GML).
- Multiple encodings broaden the market for City___.