Data Modeling Concepts introduced by CityGML and their Transferability to other Geospatial Application Models

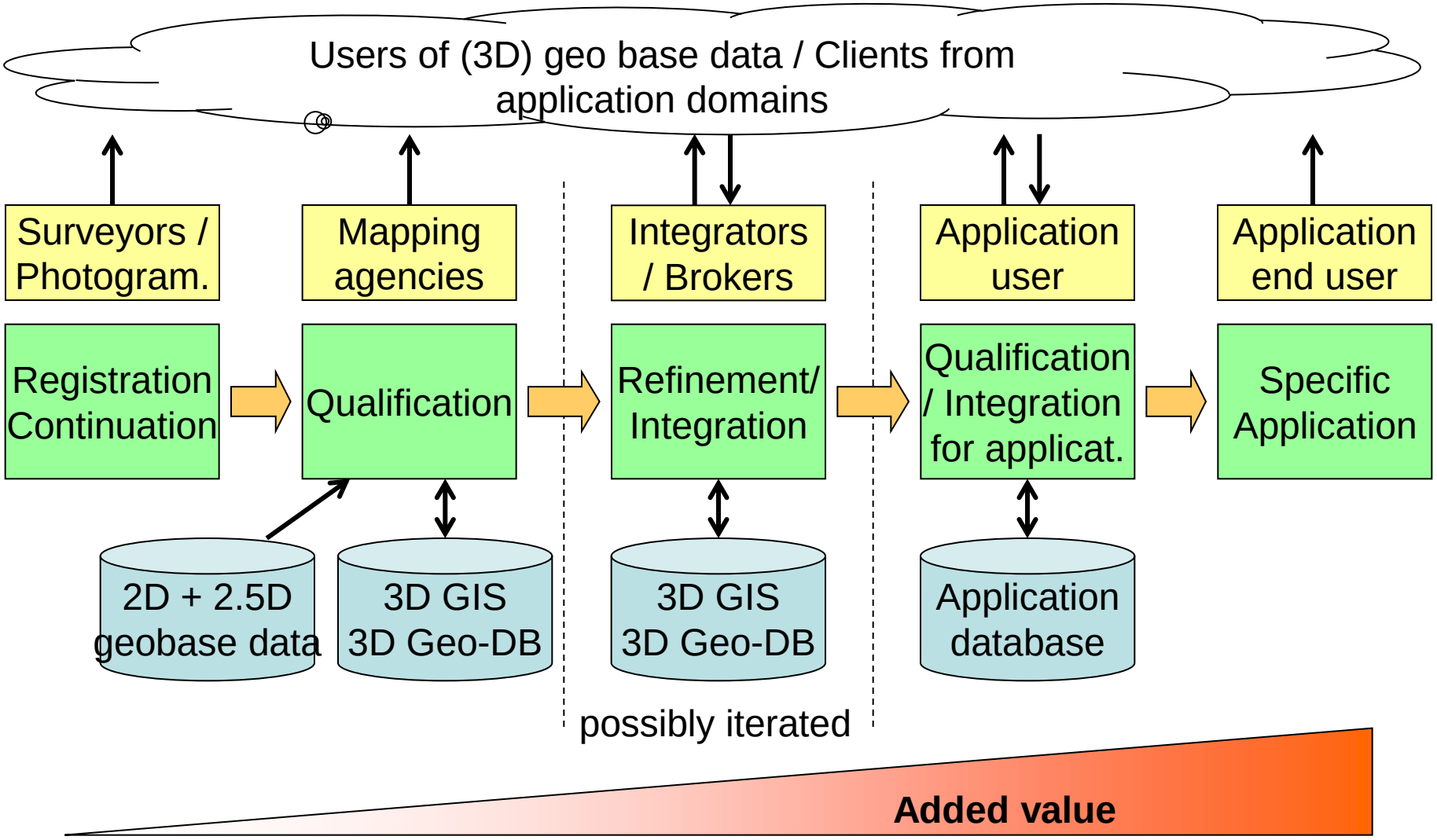
Tatjana Kutzner, Thomas H. Kolbe

Lehrstuhl für Geoinformatik
Technische Universität München

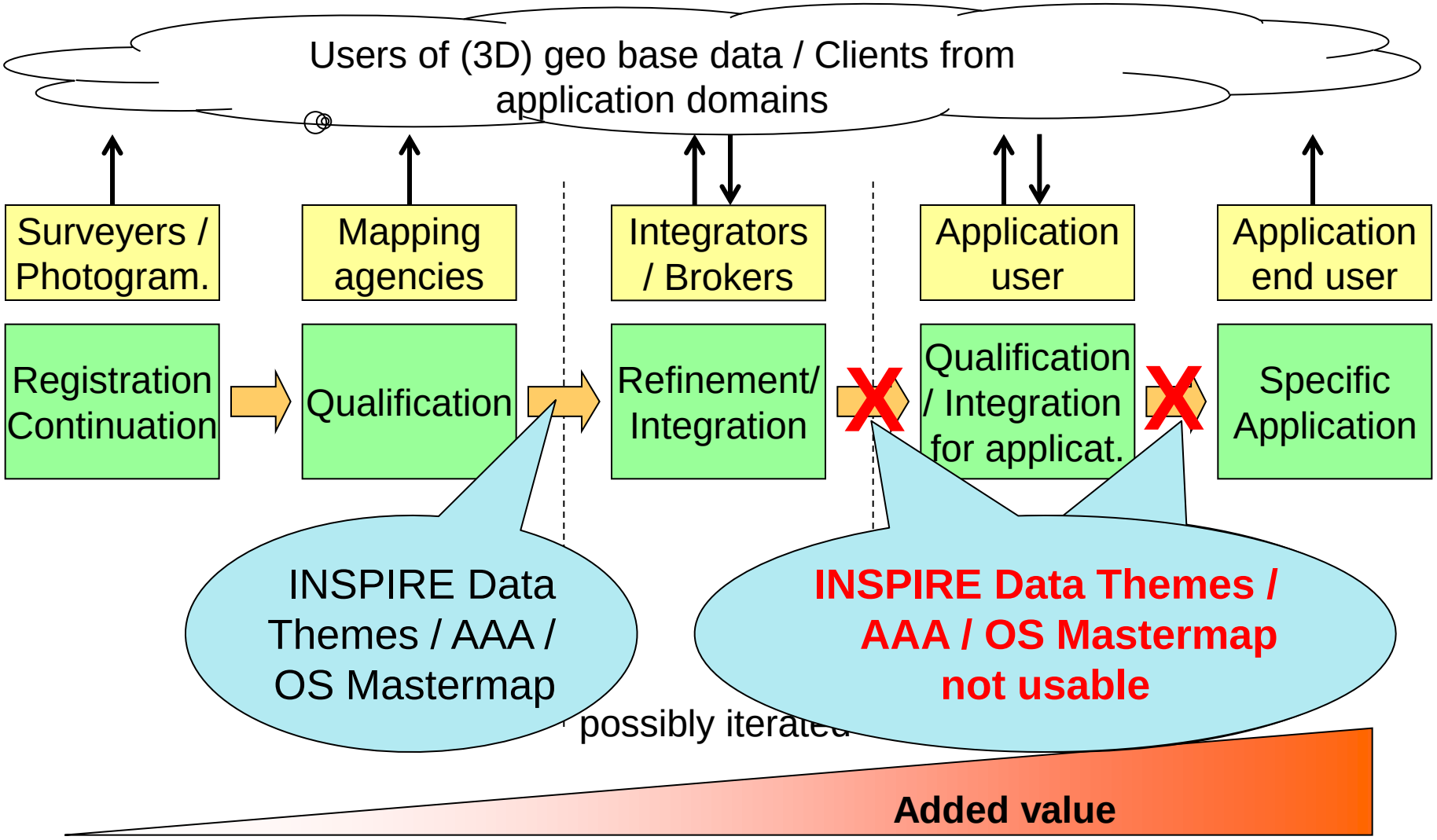
kutzner@tum.de

Data Modelling and Model Driven Implementation of Data Distribution Workshop, Copenhagen, 28.-30.01.2015

Data Processing Chain



Usability limitations due to unflexible data models



City Geography Markup Language (CityGML)

- Application independent geospatial information model for semantic 3D city and landscape models
- comprises different thematic areas (buildings, vegetation, water, terrain, traffic, tunnels, bridges etc.)

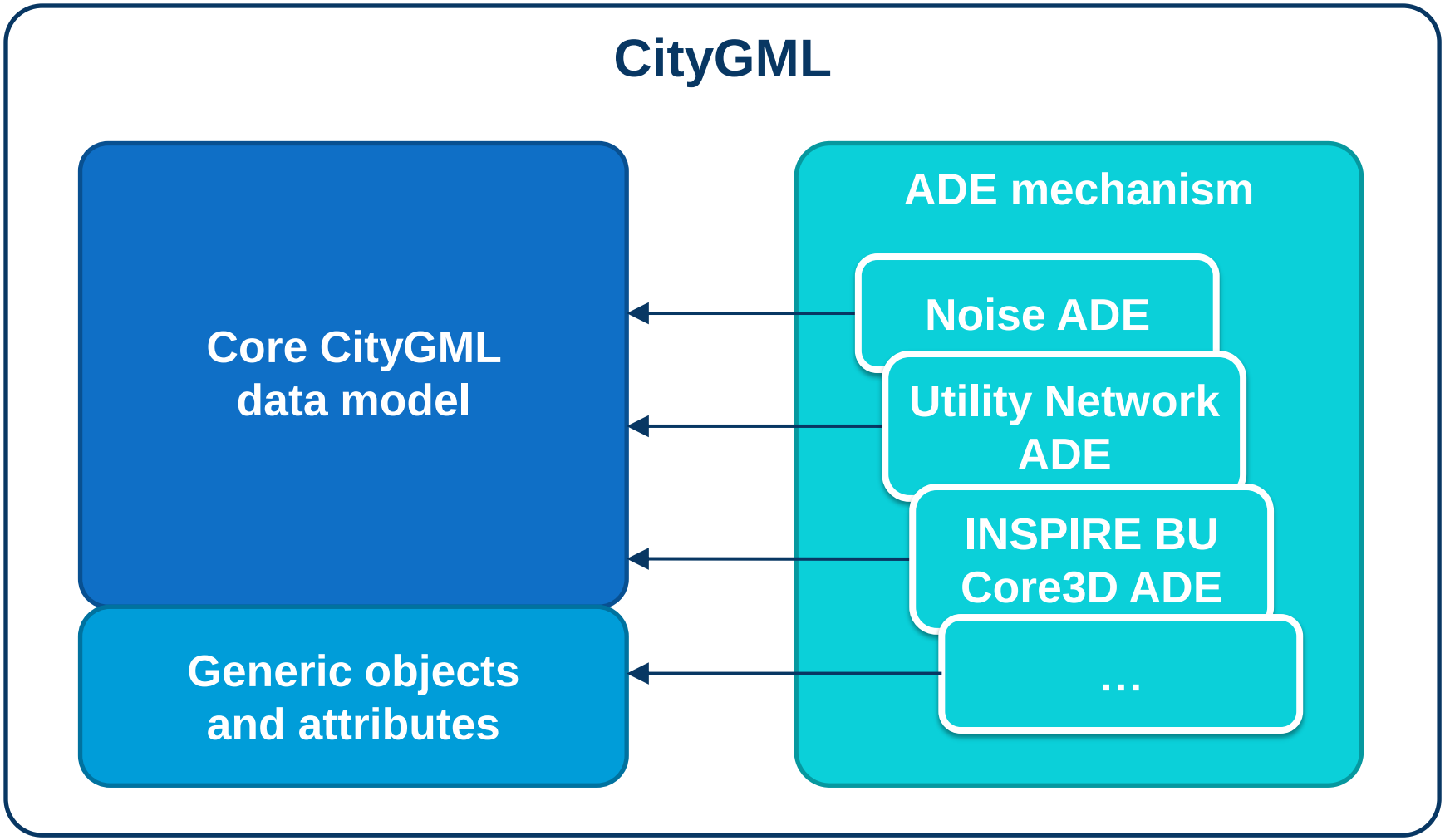
- International standard of the Open Geospatial Consortium
  - V1.0.0 adopted in 08/2008; V2.0.0 adopted in 3/2012
  - V3.0.0 development started in 6/2014
- Data model (UML) + Exchange format (based on GML3)
- CityGML represents
  - 3D geometry, 3D topology, semantics, and appearance
  - in 5 discrete scales (Levels of Detail, LOD)

(Inter)national Usage / Availability of CityGML

- **Cities / Municipalities**
  - e.g. almost all German cities with 3D city models; Rotterdam, Zürich, Geneva, Paris, Marseille, Istanbul, Vancouver, Montreal, Kuala Lumpur, Yokohama, Doha; however, few implementations in the USA (Blacksburg, Boston in preparation)
- **Organisations**
  - e.g. IGN France, Ordnance Survey UK, State Mapping Agencies of Bavaria, Baden-Württemberg, Hesse, Rhineland-Palatinate, North Rhine-Westphalia, BIMTAŞ in Istanbul, **many companies, research institutes, and universities**
- CityGML is **reference model** in the European **INSPIRE** initiative (→ full EU coverage)
  - INSPIRE Building model is based on CityGML

- The official national and municipal 3D geoinformation standards of Germany and the Netherlands base on CityGML

The CityGML Tripartition Model (I)

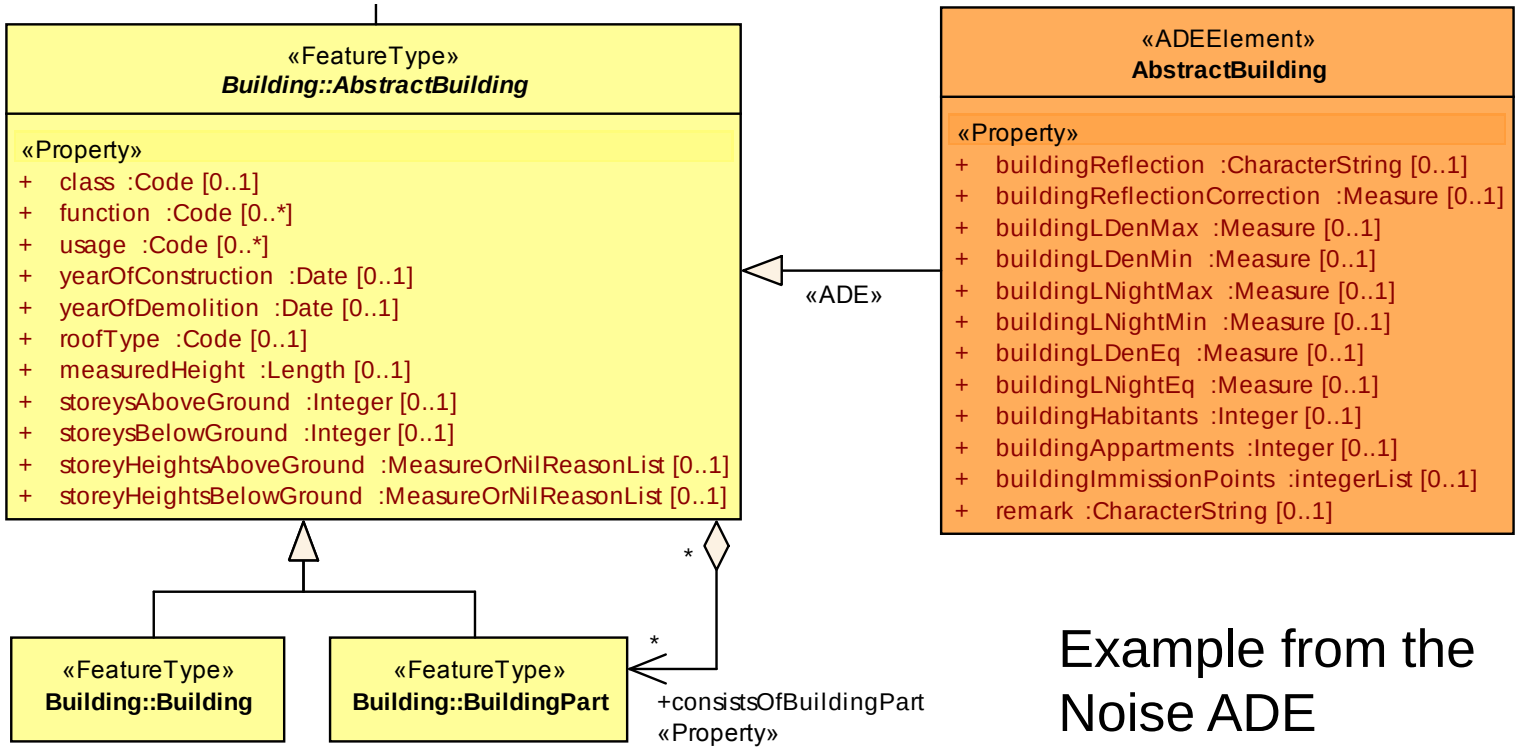


The CityGML Tripartition Model (II)

1. Core CityGML data model
   - A structured, strict core which consists of a set of well-defined classes and codelists
   - is extensible by generic objects and attributes as well as ADEs
2. Generic objects and attributes
   - Semi-structured extension mechanism which allows flexibly extending the core without making a schema modification necessary
   - Provides the ability for ad-hoc enrichment, e.g. by intermediate computation results → "extension during run-time"
   - Makes CityGML less prescriptive than e.g. INSPIRE
3. ADE mechanism
   - Systematic extension mechanism which allows extending every CityGML object type by additional attributes and introducing new object types
   - Also extended datasets can be validated against the CityGML and the ADE schema

How the ADE mechanism works (I)

- Within a separate ADE model the new attributes are modelled as subclasses of existing classes
  - Subclasses receive the stereotype «ADEElement»
  - Generalisation relationships receive the stereotype «ADE»



Example from the Noise ADE

How the ADE mechanism works (II)

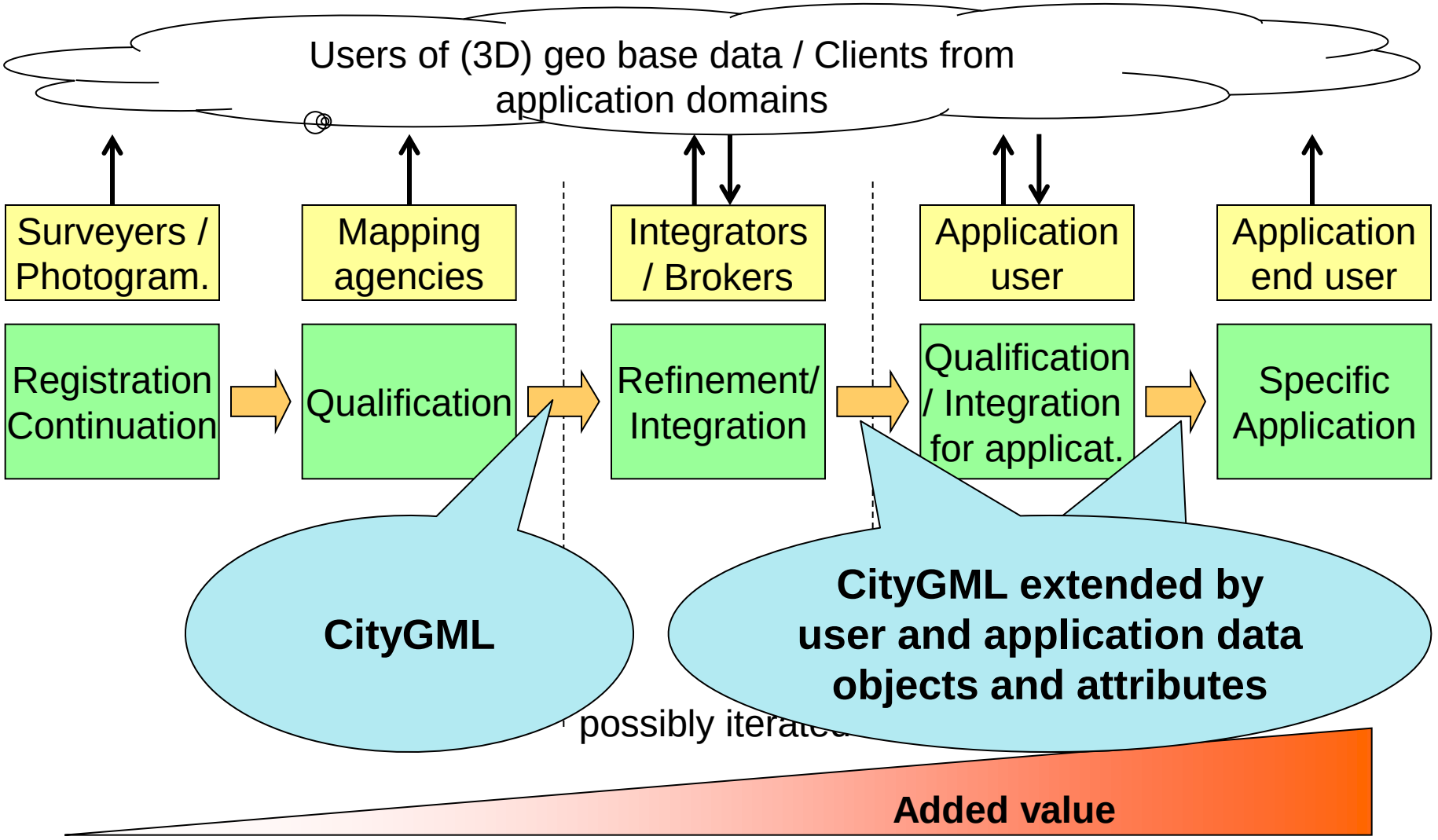
- During encoding the new attributes are injected into the respective superclass → superclass strategy

```
<cityObjectMember>
 <bldg:Building gml:id="ef6e19e3-c412-440b-8ba9">
  <bldg:function>1060</bldg:function>
  <bldg:measuredHeight uom="m">2.38</bldg:measuredHeight>
  . . .
  <noise:buildingReflection>Fassade</noise:buildingReflection>
  <noise:buildingReflectionCorrection uom="dB">
    3.23
  </noise:buildingReflectionCorrection>
  <noise:buildingLDenMax uom="dB">10</noise:buildingLDenMax>
  <noise:buildingLDenMin uom="dB">30</noise:buildingLDenMin>
  <noise:buildingLDenEq uom="dB">20</noise:buildingLDenEq>
  . . .
 </bldg:Building>
</cityObjectMember>
```

standard CityGML attributes of a building

Noise ADE attributes

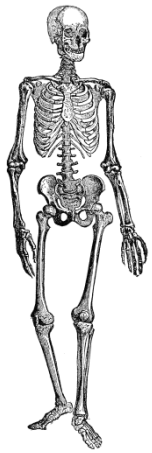
CityGML is usable along the entire process chain

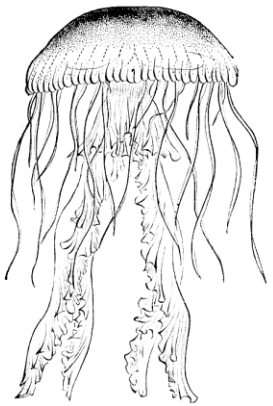


CityGML – A Modular System

- The Core CityGML data model is decomposed into thematic modules
- By using the ADE mechanism
  - individual existing modules can be extended
  - new modules can be defined systematically
- Some concepts which have been introduced to CityGML 2.0 as modules originally have been ADEs: e.g. Tunnel and Bridge
- For CityGML 3.0 the Utility Network ADE is planned to become a module

CityGML vs. Linked Data / Big Data





CityGML

- A precisely, well-structured skeleton to which semantically equally structured application-specific data can be attached
- Allows for creating consistent snapshots of data at a certain point in time, e.g. to document decisions

Linked data / Big data

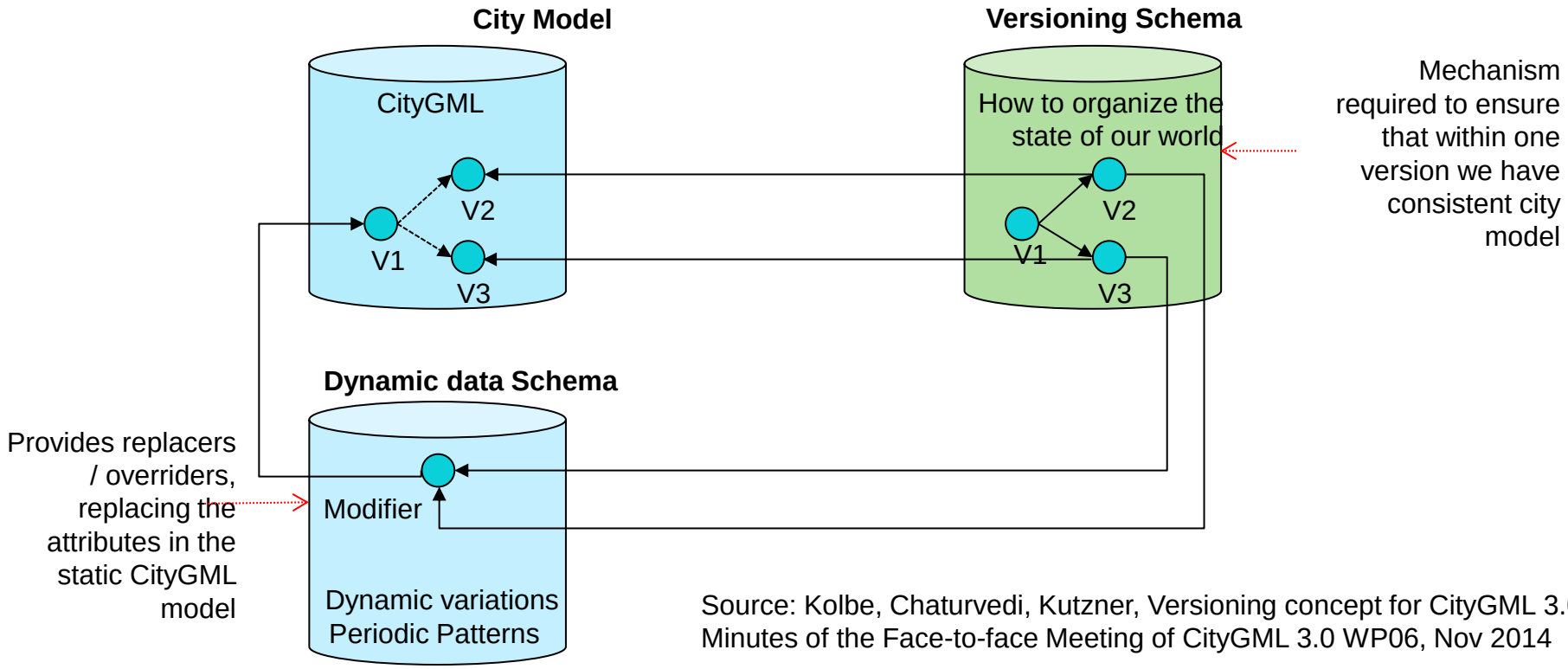
- Semantic data structure often heterogeneous
- States at a certain point in time are not restorable

CityGML – A Continuous Storage Format

- Establishing CityGML as continuously usable storage format along the processing chain prevents from
  - interruption in format/structure
  - information loss
- In this way CityGML can be used as data basis for exchange between different simulation applications
- Aim: Tighter coupling of semantic 3D city models and simulations by extending the CityGML feature representation to support
  - variations of individual feature properties and associations over time
  - complex attributes and metadata at attribute level

Current work in the context of CityGML 3.0

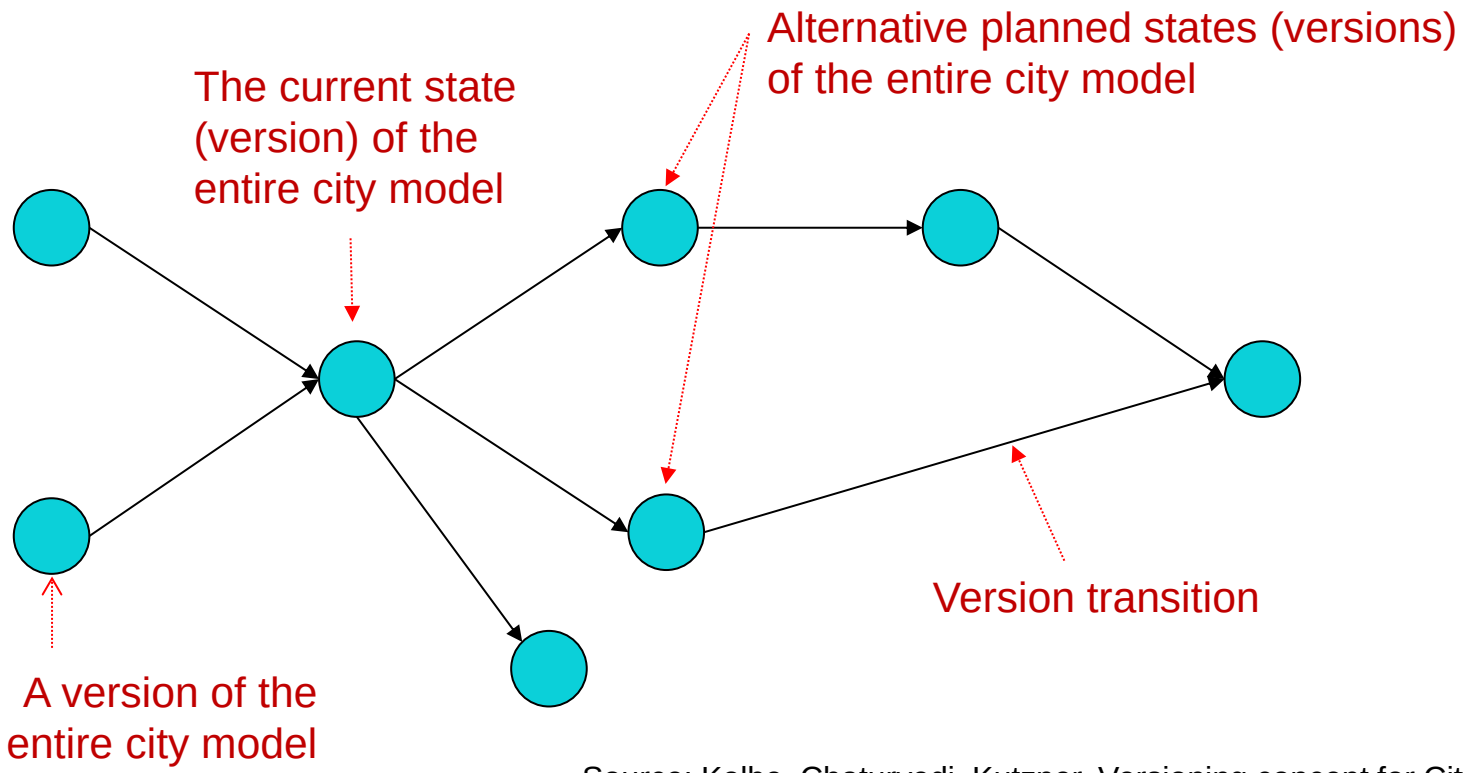
- Development of
  - A versioning schema, which represents the evolution of the city (model) in the form of different versions.
  - A dynamic data schema, where dynamic variations can be stored in special types of features, which would be interpreted as modifiers to the static CityGML model.



Source: Kolbe, Chaturvedi, Kutzner, Versioning concept for CityGML 3.0, Minutes of the Face-to-face Meeting of CityGML 3.0 WP06, Nov 2014

Versioning

- Multiple representations of the city model can be handled within different versions, which can be modeled as feature types
- Directed graph allowing for confluence and forking



Source: Kolbe, Chaturvedi, Kutzner, Versioning concept for CityGML 3.0, Minutes of the Face-to-face Meeting of CityGML 3.0 WP06, Nov 2014

Conclusion

- As long as the application schemas of geo base data do not allow to be extended by application / user data they will only live (and be used!) at the interface between the data provider and the first user / refiner. The data models (and importantly: formats) cannot be used in later steps, where data with added value is being produced and used.
- The ADE mechanism allows for extending CityGML by application-specific information (possibly from multiple application domains) and, thus, for using the same data model throughout the entire processing chain

→ This characteristic makes the ADE mechanism of advantage to other geospatial application schemas