Proposal for a revision of the CityGML LOD concept

Claus Nagel, M.Sc.
cnagel@virtualcitysystems.de

Benefits of the current CityGML LOD concept

- **Common understanding** of the distinct levels of detail
  - City models are categorized into LODs
  - Used in tender documents to specify the requirements for data acquisition and modelling
  - Software tools (and vendors) are associated with LODs
  - Applications of city models are associated with LODs

  → **Strong (market) acceptance**

- **Manageable complexity**

- One of the **most successful** concepts of CityGML



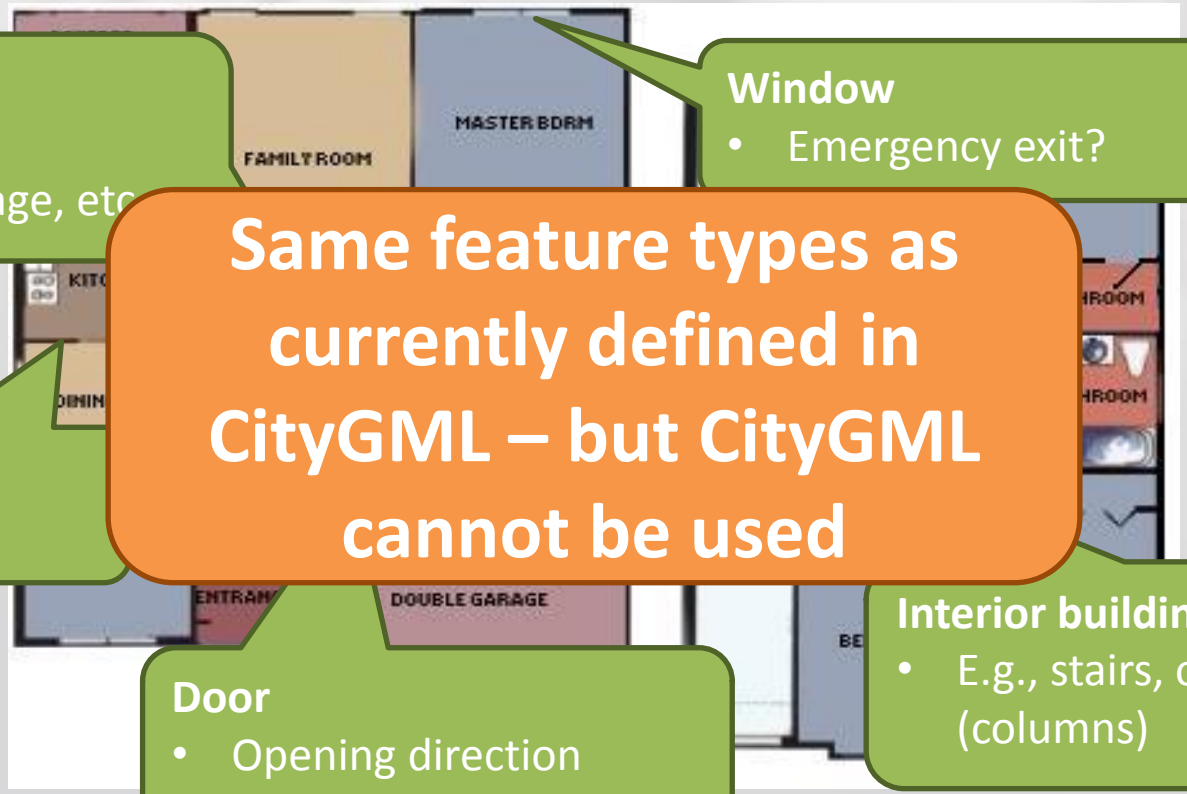


Requirements for a revision of the LOD concept

- **WHY?** Revision of LOD concept must be **use case driven**
  - Every change must be argued and justified against a use case
- **HOW?** Keep it **simple**
  - If a use case can be supported in two alternative ways, there must be very good arguments to not choose the more simple one
- **IMPACT?** Clear **link** to current LOD concept
  - Reliability for and acceptance by existing and new users, software vendors, data providers, …
  - Protection of investment as well as willingness to invest
  - Possibility to migrate existing CityGML models

Floor maps

- *Example use cases:* Indoor space management, indoor navigation, disaster management (e.g. CR 12-044)



Room (Space)
- Number
- Function, usage, etc

Window
- Emergency exit?

Same feature types as currently defined in CityGML – but CityGML cannot be used

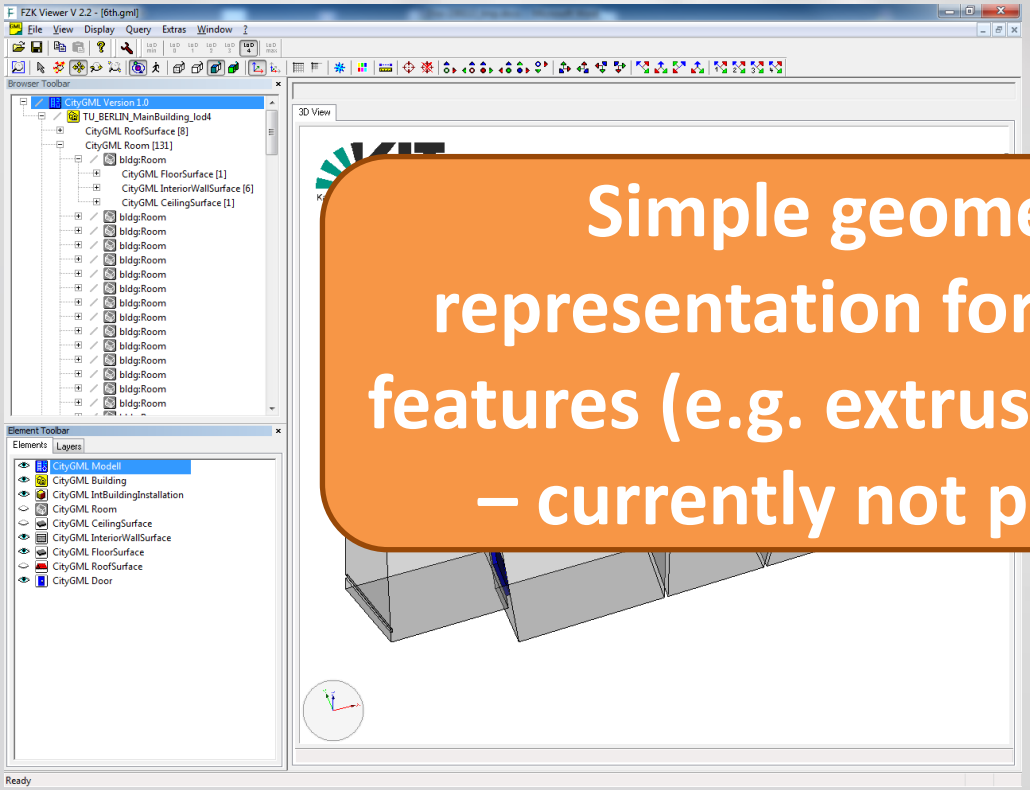
Wall(Surface)
- Material

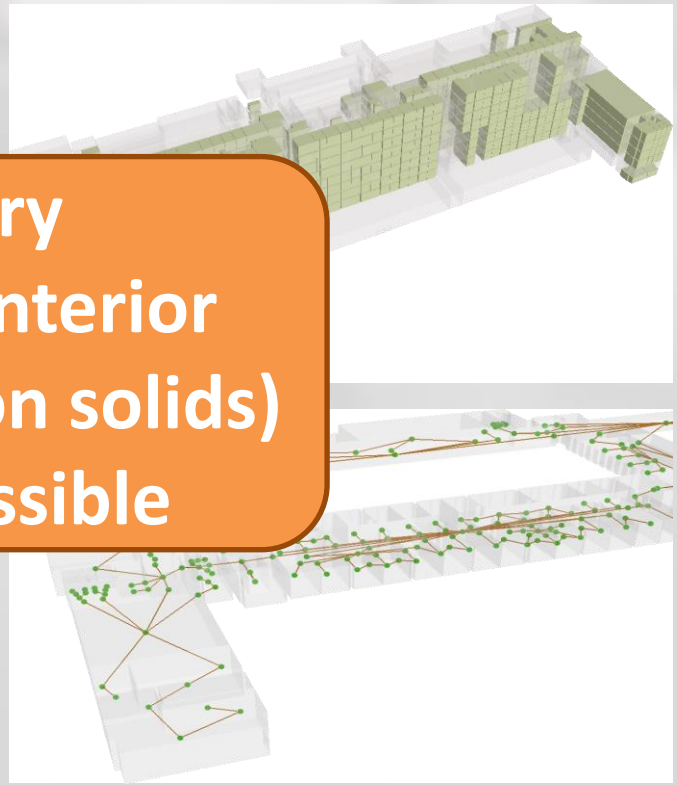
Door
- Opening direction
- Access restrictions

Interior building installation
- E.g., stairs, obstacles (columns)

Simple LODs for interior features

- *Example use cases:* Indoor space management, indoor navigation, disaster management (e.g. CR 12-044)





Simple geometry representation for interior features (e.g. extrusion solids) – currently not possible

Proposal for a revision of the LOD concept

One change: Cancel usage restrictions on feature types in LODs

- *Currently:* BoundarySurfaces from LOD2, openings from LOD3, rooms from LOD4

Idea: Every feature(!) can be ***geometrically*** described in **four LODs**

- LOD0: Projection onto a 2.5d reference surface (e.g. terrain, floor) as point, line or surface
- LOD1: Block representation (simple extrusion solids)
- LOD2: Generalized representation
- LOD3: Detailed representation

That's it! (Current LOD4 is not required any more)

Example LODs for some feature types

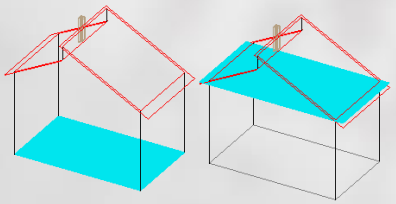
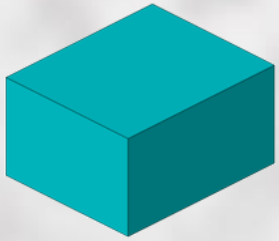
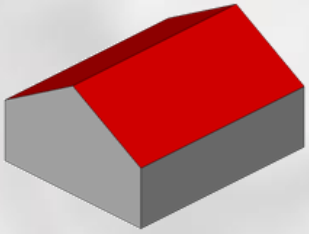
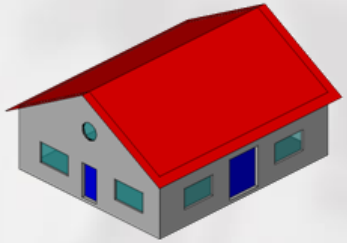
LOD0 **LOD1** **LOD2** **LOD3**

Building (outer shell! → not boundary surfaces)

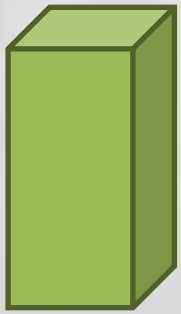
Strong advantage:
Close to CityGML 1.0/2.0

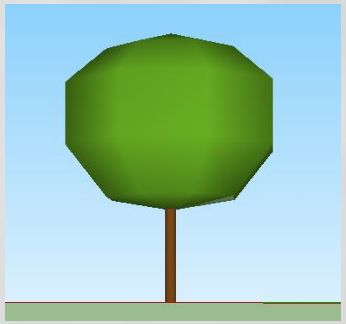
LOD2 building shell with LOD2 openings?
Sure, go ahead
(it is not in the figures though)

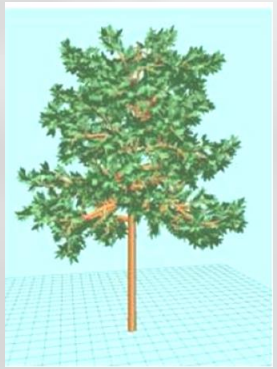
Example LODs for some feature types

LOD0	LOD1	LOD2	LOD3
			

Building (outer shell! → not boundary surfaces)



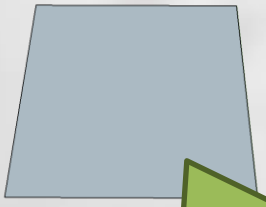
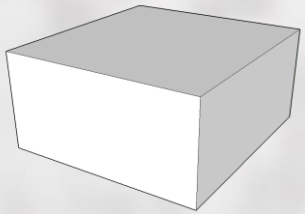
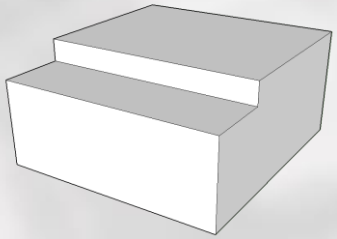
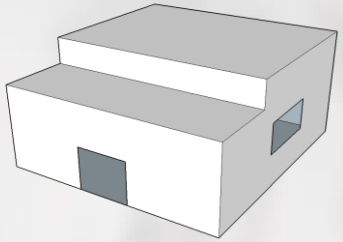
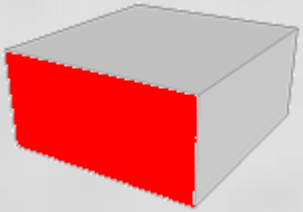
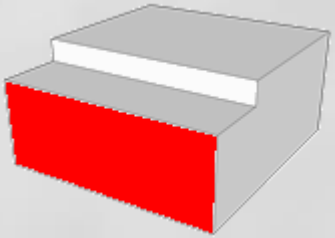
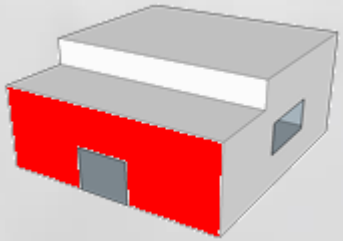




Soli

For example, projected onto terrain

Example LODs for some feature types

	LOD0	LOD1	LOD2	LOD3
Room				
(Interior)WallSurface				

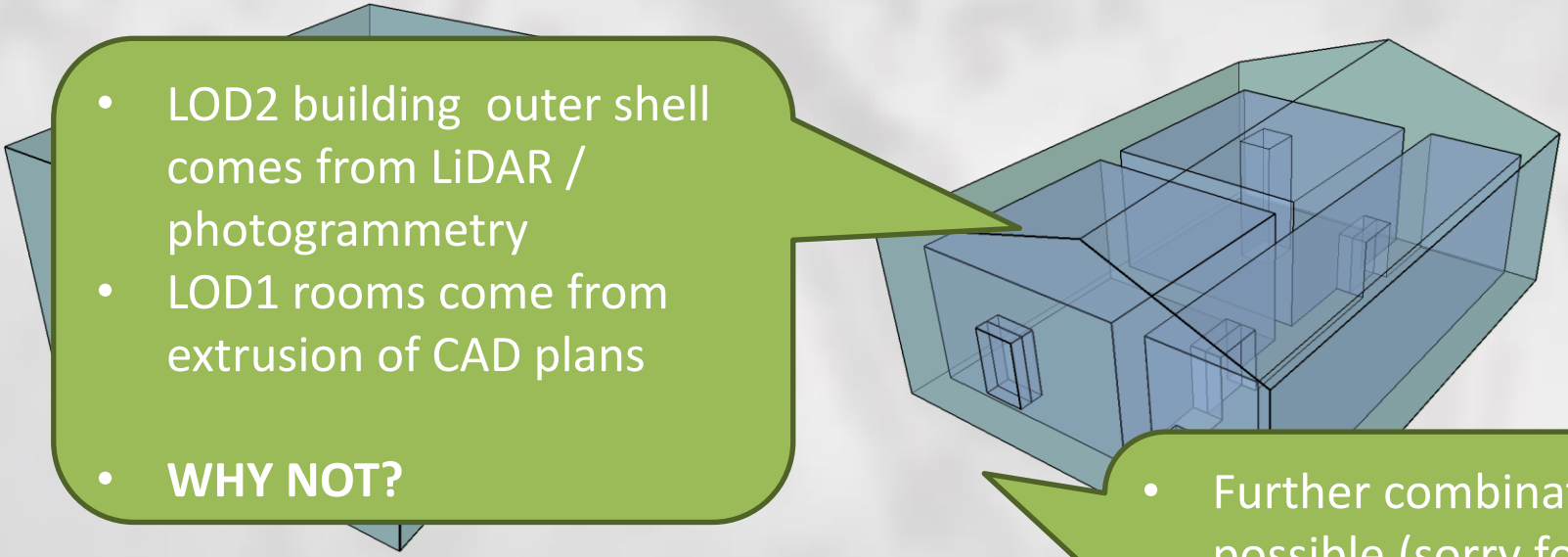
For example, projected onto floor plane

Combining multiple LODs

virtualcitySYSTEMS

The Digital City Experts. ERSTELLEN VERWALTEN VERÖFFENTLICHEN

- LOD2 building outer shell comes from LiDAR / photogrammetry
- LOD1 rooms come from extrusion of CAD plans
- **WHY NOT?**



LOD1 Rooms
LOD1 Building (shell)

LOD1 Ro
LOD2 Bu

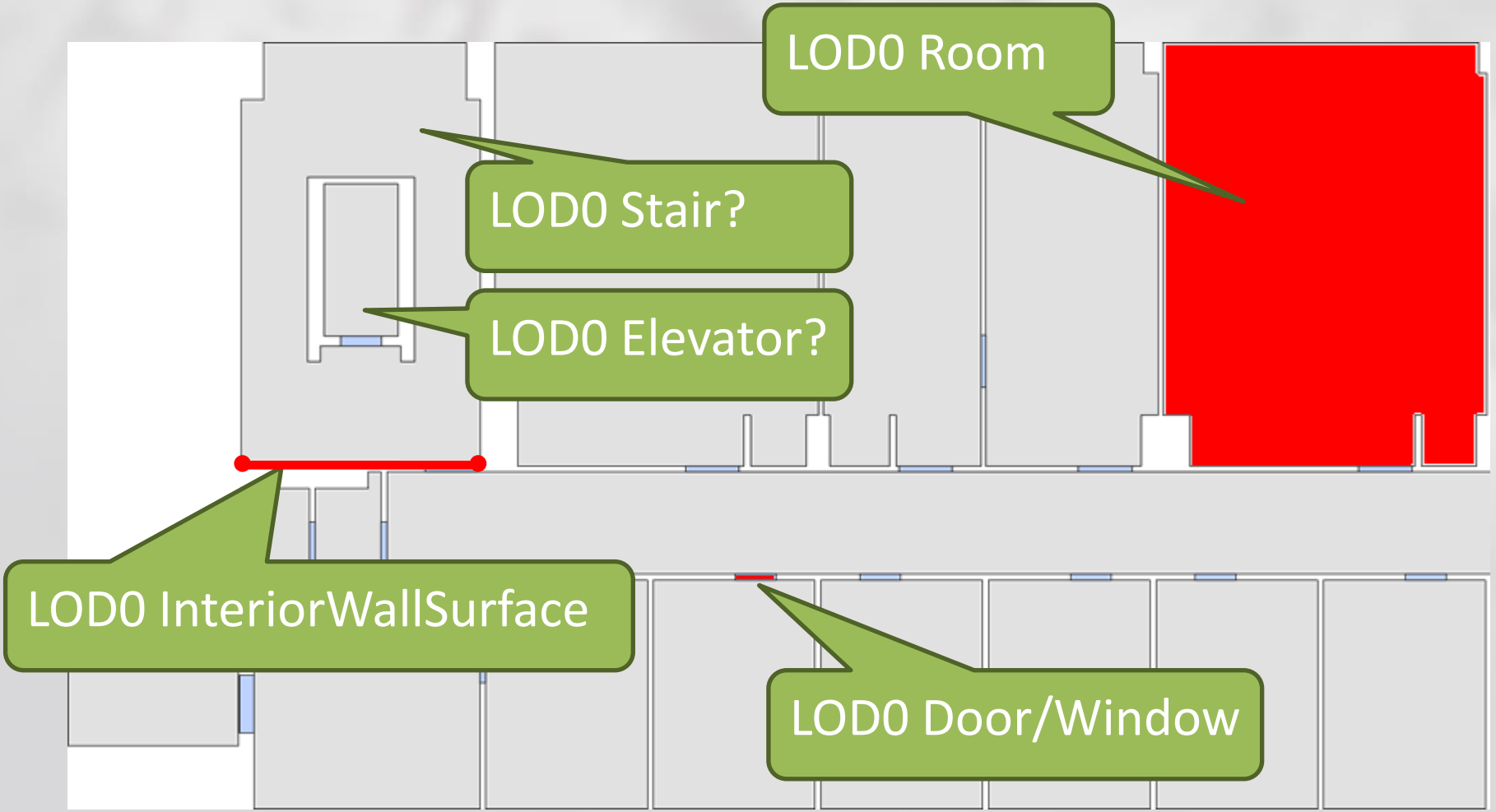
- Further combinations possible (sorry for not providing images…)
- The same idea applies to **all modules**

- Reasonable combinations will follow from ***use cases / applications***
- The CityGML standard **should not restrict** combinations a priori

Realizing floor plans through LOD0 features

virtualcitySYSTEMS

The Digital City Experts. ERSTELLEN VERWALTEN VERÖFFENTLICHEN

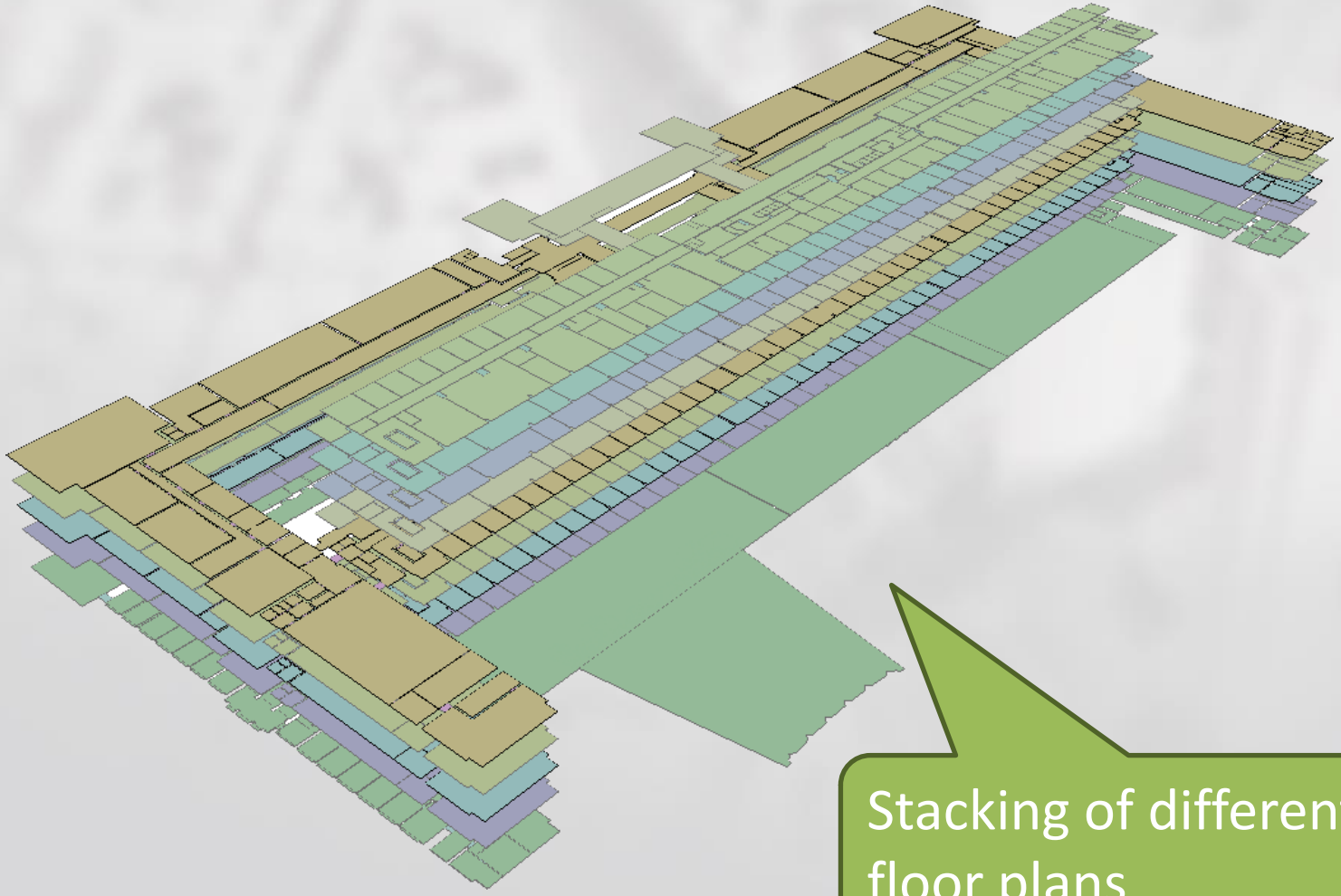


Realizing floor plans through LOD0 features

virtualcitySYSTEMS

The Digital City Experts.

ERSTELLEN | VERWALTEN | VERÖFFENTLICHEN



Stacking of different LOD0 floor plans

Relation to “old LOD” representation

- **How to realize the current LOD representations based on the new concept?**

- Restrict the CityGML data model to the application needs
  - **Idea:** Simply disallow specific LOD representations for some feature types or even complete feature types
  - **Technical how:** Profiling is the ISO way to restrict application schemas
  - Profiles can be validated and are supported by COTS CityGML software (in contrast to ADEs)

- Example “old LOD2” for buildings:
  - Only allow feature types Building, BuildingPart, WallSurface, GroundSurface, RoofSurface and ClosureSurface
  - Only allow LOD2 representation for these features

Properties of proposed LOD concept

- LODs focus on geometric description only
  - Semantic depth naturally follows from the number and type of the modelled features
  - It **cannot be mandated** to always model all possible feature types
- No (academic) differentiation into “exterior/interior” or “geometric/semantic” LODs
  - Benefit: LOD4 in its current form is no longer required → current issues with this LOD are resolved
- Combinations between LODs possible
  - E.g., LOD3 building shell with LOD1 rooms or vice versa
  - Whatever is required for an application / use case
  - CityGML makes no a priori assumptions

Properties of proposed LOD concept

- Clear link to existing concept
  - No new terminology
  - Same understanding of LOD0 – LOD3 (geometrically)
  - Clear mapping from “old LODs” (e.g., using profiles)
    → **Existing CityGML data does not become invalid**
- All current use cases plus additional use cases outlined in this presentation can be covered
- **SIMPLE, SIMPLE, SIMPLE**
  - Just **one change** to the current LOD concept: *Cancel the usage restrictions on feature types in LODs*

Further discussion

The base LOD concept should be accompanied by:

- Metadata for LODs
  - Geometry accuracy / precision; vertical / horizontal reference
  - Semantic richness
  - Suitability for a specific application
  - Data quality?
- Functional dependency between LODs
  - Can lower LODs be computed from higher LODs?
  - IMHO, not really a necessary requirement
- UML modelling and encoding rules for LODs in ADEs (!)
  - Currently, an ADE parser cannot understand LODs of ADE features in an ADE-enriched instance document