ifp

University of Stuttgart

Towards Crowd-Sourced 3D Indoor Reconstruction Based on Grammar Support

Susanne Becker

3DGI 2016 – 3D-Geoinformation in Aktion
Olten, 29th September 2016

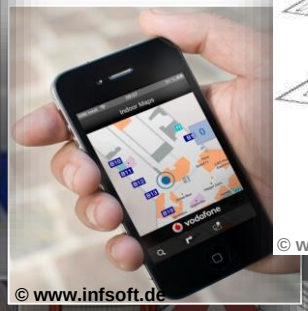
ESF Chancen fördern
EUROPÄISCHER SOZIALFONDS IN BADEN-WÜRTTEMBERG

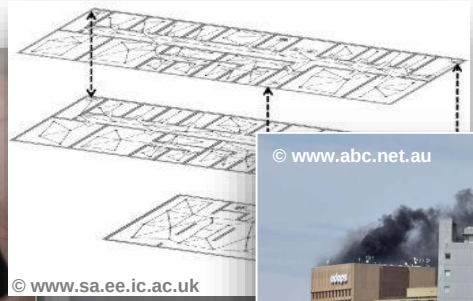
Baden-Württemberg
MINISTERIUM FÜR WISSENSCHAFT, FORSCHUNG UND KUNST

3D Indoor Models

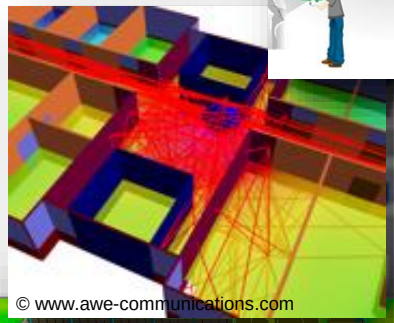
© www.gov.hk

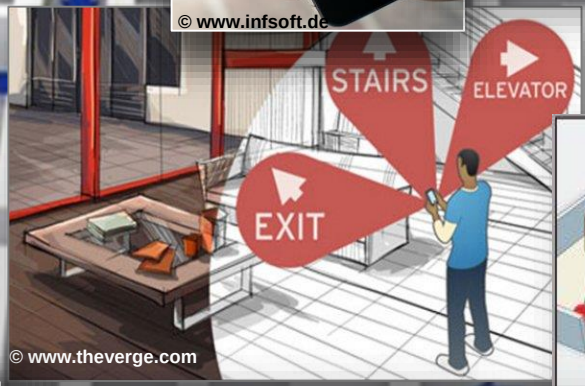
• • •


© www.infsoft.de


© www.sa.ee.ic.ac.uk

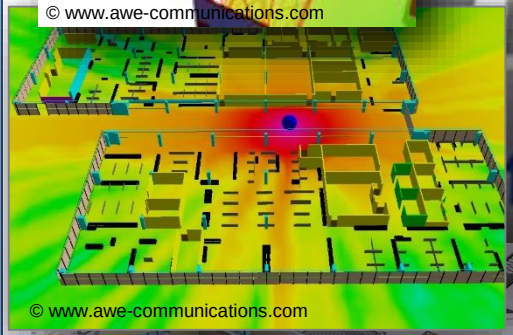

© www.abc.net.au

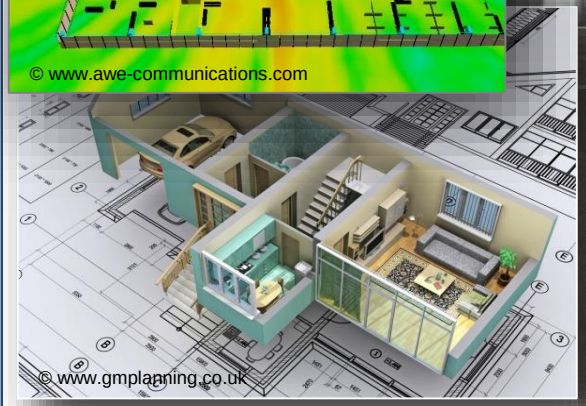

© www.awe-communications.com



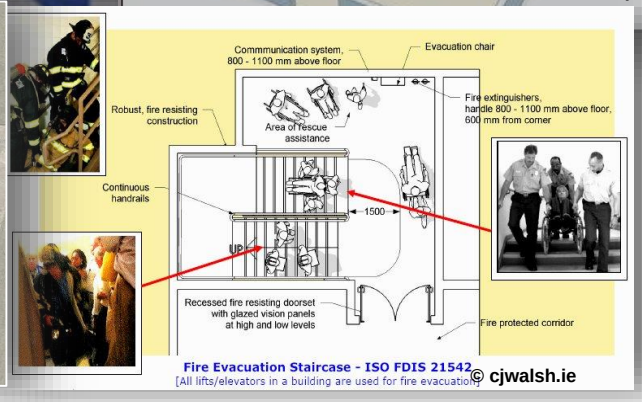
© www.theverge.com

© www.evacmap.com


© www.awe-communications.com


© www.gmplanning.co.uk

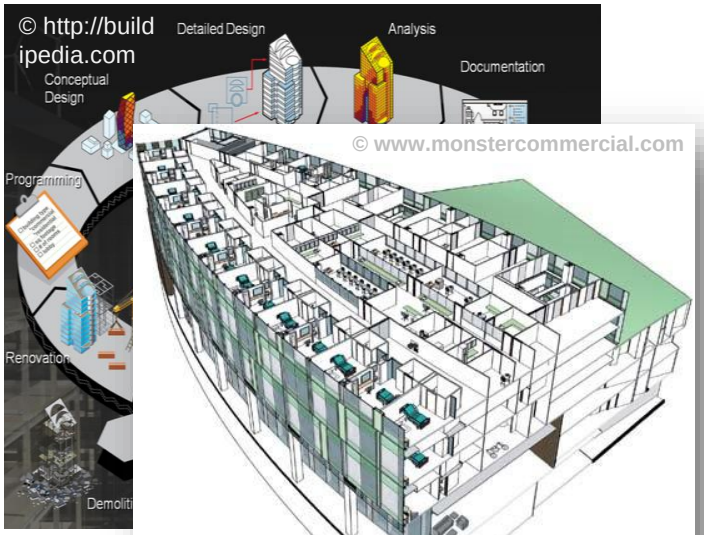

© www. cg.cis.upenn.edu



© cjwalsh.ie

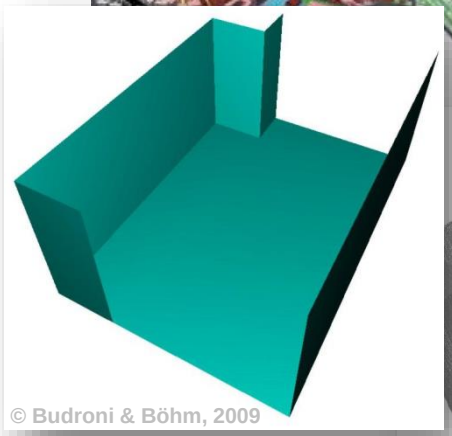
How to Get 3D Indoor Models?

- BIM world



  - **high** geometric and semantic **detail**
  - **manual creation**

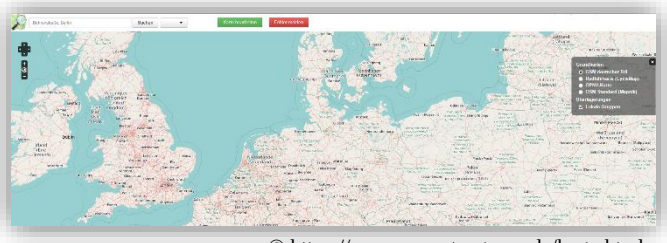
- 3D GIS world



  - mostly pure geometry models with **limited geometric detail**
  - focus on **automatic derivation** from observation data
  - **High costs** in money, time and expertise for **data acquisition**

Crowd-Based Mapping

- OpenStreetMap (since 2004)
  - Crowd-based data acquisition for creating **2D maps of the world**



© https://www.openstreetmap.de/karte.html

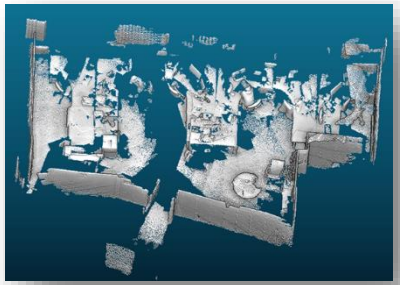
- DFG-Project Com'N'Sense (since 2013)
  - **Idea: Private users automatically** create **2D and 3D indoor models** with their smartphones **while passing through** public buildings *(Peter et al., 2013; Philipp et al., 2014)*
  - **Data:** noisy, erroneous, incomplete
  - **Goal: robust automatic** approach for 3D indoor reconstruction from **erroneous** or **incomplete observation**
  - **Powerful means: formal grammars**

© clipartpanda.com

© Google Project Tango

© Lenovo, Phab 2 Pro

© zd.net



© Lavinia Runceanu, 2016

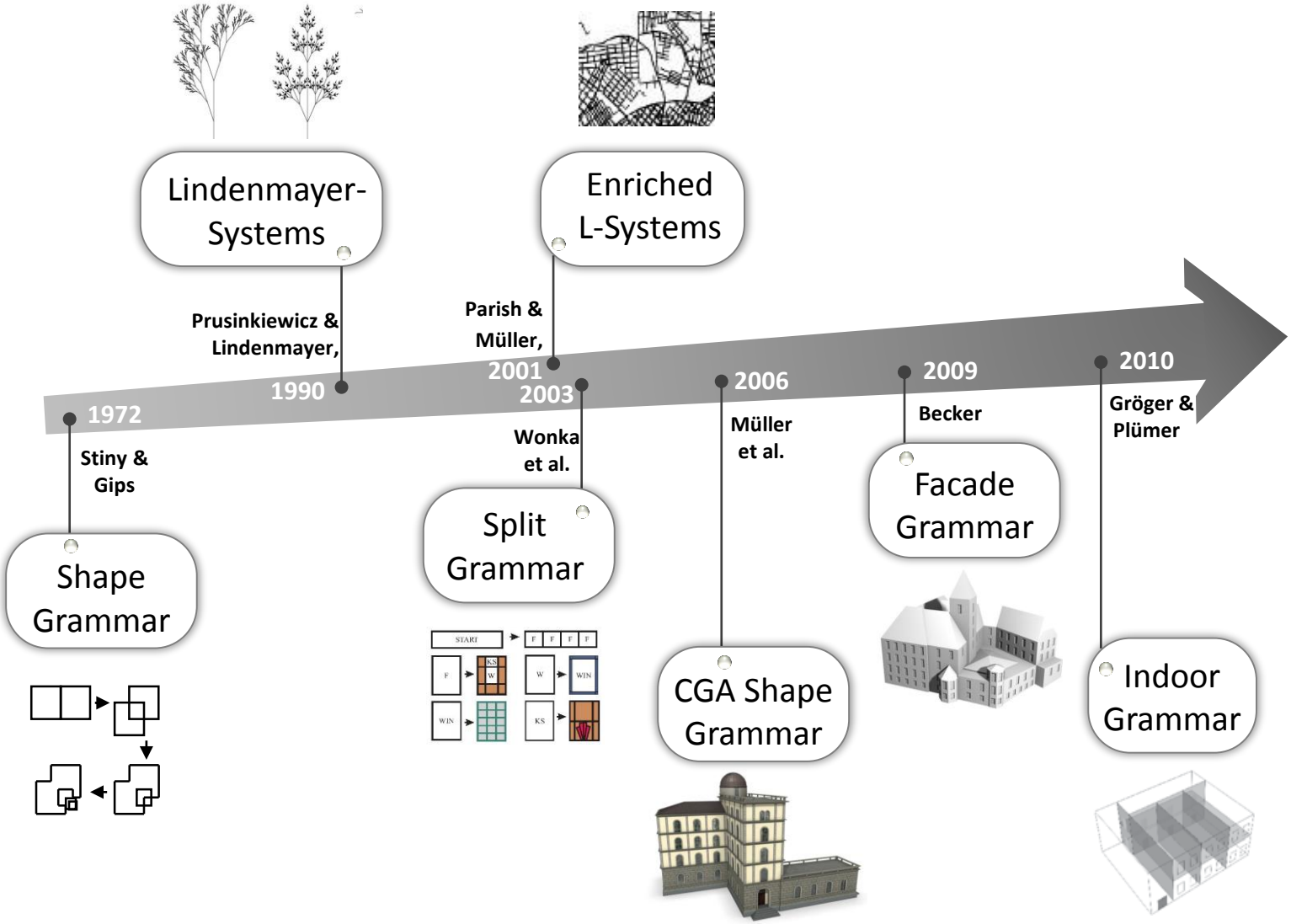
Formal Grammar

- Defines a **formal language** an amount of sequences of symbols ***(amount of words)***
  - Symbols ⇒ **alphabet**
    - *E.g.:* $V_1, V_2, \ldots, t_1, t_2, \ldots$
  - Rules for the generation of sequences of symbols ⇒ **syntax**
    - *E.g.:* $\mathbf{p_1}$: $t_1 < V_1 > t_2 \rightarrow V_2\, t_1$: prob
      $\mathbf{p_2}$: $t_2 < V_2 > t_3 \rightarrow t_2$: prob
      ⋮
- Exemplary grammar:
  - Symbols:
    - $V, 0, 1$
  - Rules:
    - p_1: $V \rightarrow V\,0$
    - p_2: $V \rightarrow 1$

⇒ Formal language L = { 1, 10, 100, … }

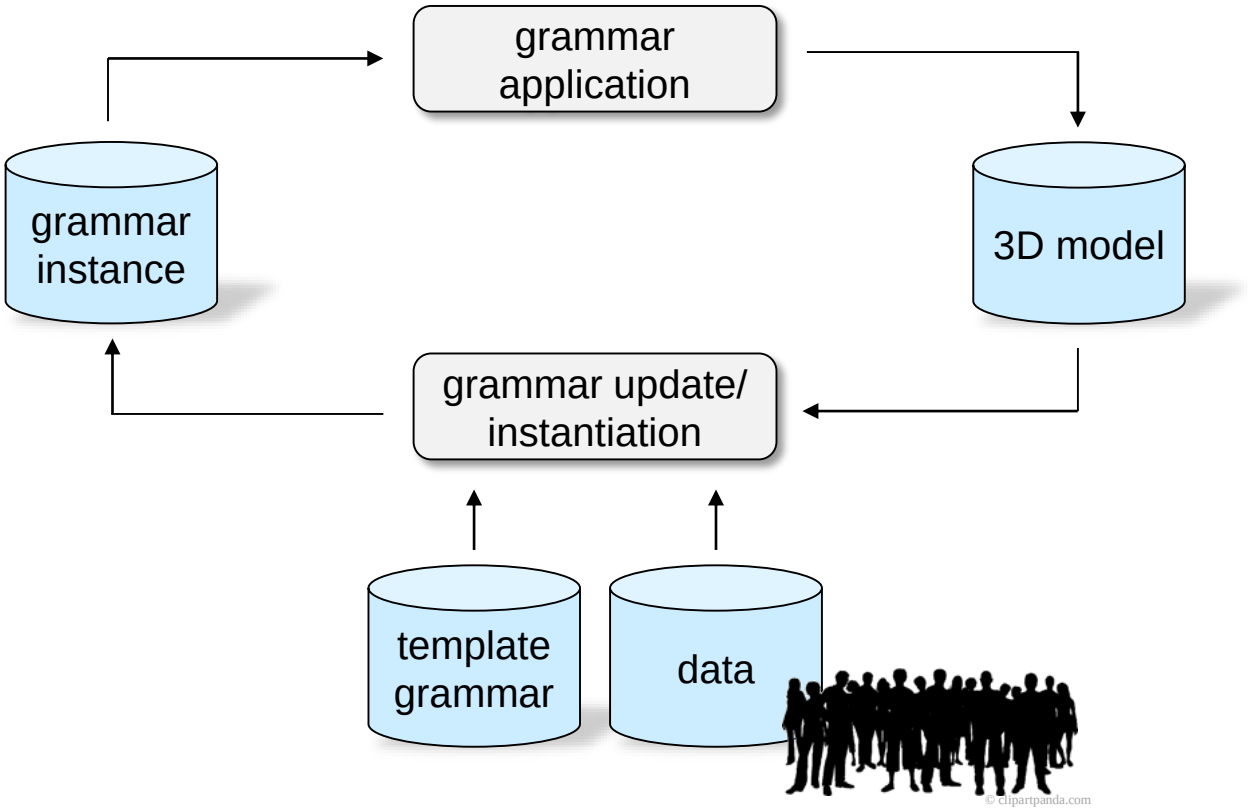
Formal Grammars

for modeling geometric structures



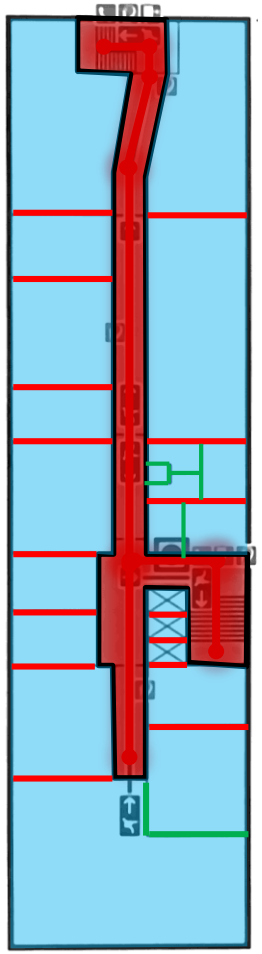
Iterative Grammar-Based Approach

- Iterative process of *grammar application* and *grammar update*



3D Indoor Grammar
Design Decisions

- Focus
  - Office buildings
  - Public buildings (e.g. schools, hotels, hospitals etc.)
- Characteristics of such buildings
  - Buildings are traversed by a *system of hallways*.
  - The *system of hallways* divides each floor into ***hallway-spaces*** and ***non-hallway-spaces***.
  - *Non-hallway-spaces* can be further divided into *smaller room units* mostly arranged in a linear *sequence* parallel to the adjacent hallway.
- Grammar concept
  - **Hallway system** *(linear structures)*
    ⇒ L-system
  - **Room configurations** *(spatial partitioning)*
    ⇒ split-grammar

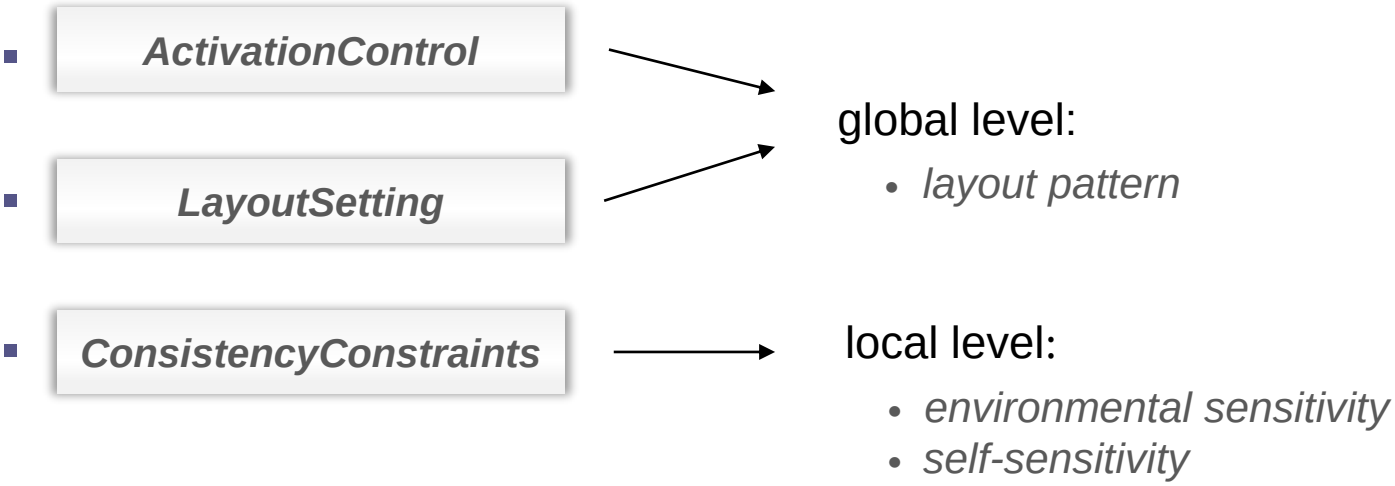


floorplan of ifp

3D Indoor Grammar

L-System for Modeling Hallways

- $G^{hallways} = (V, \omega, P)$
  - V : set of attributed symbols (modules),
  - ω : axiom (initial hallway segment)
  - P : production rules
  - related to the **enriched L-system** for modeling 2D streets (Parish & Müller, 2001)
  - **Idea**: organize the setting of **attributes**, **probabilities** and the **constraints** induced by the geometric environment through **external functions**:



3D Indoor Grammar

L-System for Modeling Hallways

- Production rules

- ω: $\mathbf{R}(ACTIVE)\mathbf{?I}(\theta_{init}, UNASSIGNED)$
- p1: $\mathbf{R}(mode) > \mathbf{?I}(\theta, state) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\{$ ***LayoutSetting*** $(mode, \theta)$ sets $\theta_p[0\text{-}4]\} \rightarrow +(\theta.angle)\mathbf{F}(\theta.len)$
  $\mathbf{B^h}(ACTIVE, \theta_p[1])\ \mathbf{B^h}(ACTIVE, \theta_p[2])\ \mathbf{B^v}(INACTIVE, \theta_p[3])$
  $\mathbf{B^v}(INACTIVE, \theta_p[4])\ \mathbf{R}(ACTIVE)\mathbf{?I}(\theta_p[0], UNASSIGNED)$
- p2: $\mathbf{R}(mode) > \mathbf{?I}(\theta, state) : state==FAILED \rightarrow \varepsilon$
- p3: $\mathbf{B^h}(mode, \theta) : mode==ACTIVE \rightarrow [\mathbf{R}(\text{mode})\mathbf{?I}(\theta, UNASSIGNED)]$
- p4: $\mathbf{?I}(\theta, state) : state==UNASSIGNED$
  $\{$ ***ConsistencyConstraints*** (θ) adjusts $state, \theta\} \rightarrow \mathbf{?I}(\theta, state)$
- p5: $\mathbf{?I}(\theta, state) : state!=UNASSIGNED \rightarrow \varepsilon$
- p6: $\mathbf{B^v}(mode, \theta) : mode==INACTIVE$
  $\{$ ***ActivationControl*** sets $mode\} \rightarrow \mathbf{B^v}(mode, \theta)$
- p7: $\mathbf{B^v}(mode, \theta) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\rightarrow [\mathbf{Q}(\text{mode})\mathbf{?I}(\theta, UNASSIGNED)]$
- p8: $\mathbf{Q}(mode) > \mathbf{?I}(\theta, state) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\{$ ***LayoutSetting*** $(mode, \theta)$ sets $\theta_p[0\text{-}3]\} \rightarrow +(\theta.angle)\mathbf{U}(\theta.len)$
  $\mathbf{B^h}(ACTIVE, \theta_p[1])\ \mathbf{B^h}(ACTIVE, \theta_p[2])\ \mathbf{B^v}(INACTIVE, \theta_p[3])$
  $\mathbf{R}(ACTIVE)\mathbf{?I}(\theta_p[0], UNASSIGNED)$
- p9: $\mathbf{Q}(mode) > \mathbf{?I}(\theta, state) : state==FAILED \rightarrow \varepsilon$

→ **horizontal** growth process

→ **vertical** growth process

3D Indoor Grammar

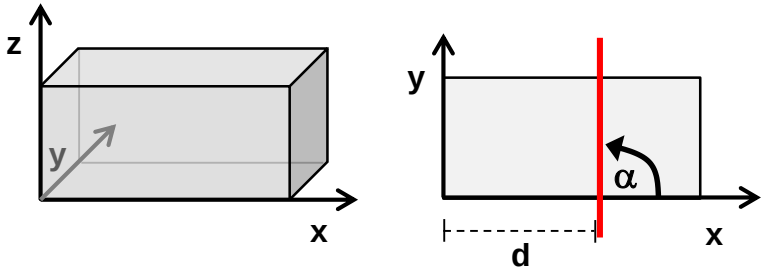
Split-Grammar for Modeling Rooms

- $G^{rooms} = (N,T,S,R)$
  - $N = \{Space\}$
  - $T = \{\ldots, space_i, space_j, \ldots\}$
  - $S = Space$
  - $R = \{Split, Merge, Instantiation\}$
    - $R_i^{SingleSplit} : Space \rightarrow Split^{Space}(\boldsymbol{n}_i, d_i)$

      with $Split^{Space}(\boldsymbol{n}_i, d_i) := Split_i^{Space} = Space^{\,l}\ Space^{\,r}$
    - $R_i^{RepeatSplit} : Space \rightarrow Split_i^{Space^{\,r}} \circ \cdots \circ Split_i^{Space^{\,r}} \circ Split_i^{Space}$
    - $R_{ij\ldots k}^{StringSplit} : Space \rightarrow Split_k^{Space^{\,r}} \circ \cdots \circ Split_j^{Space^{\,r}} \circ Split_i^{Space}$
    - $R_{ab\ldots c}^{MultiSplit} : Space \rightarrow Split_c^{*} \circ \cdots \circ Split_b^{*} \circ Split_a^{Space}$

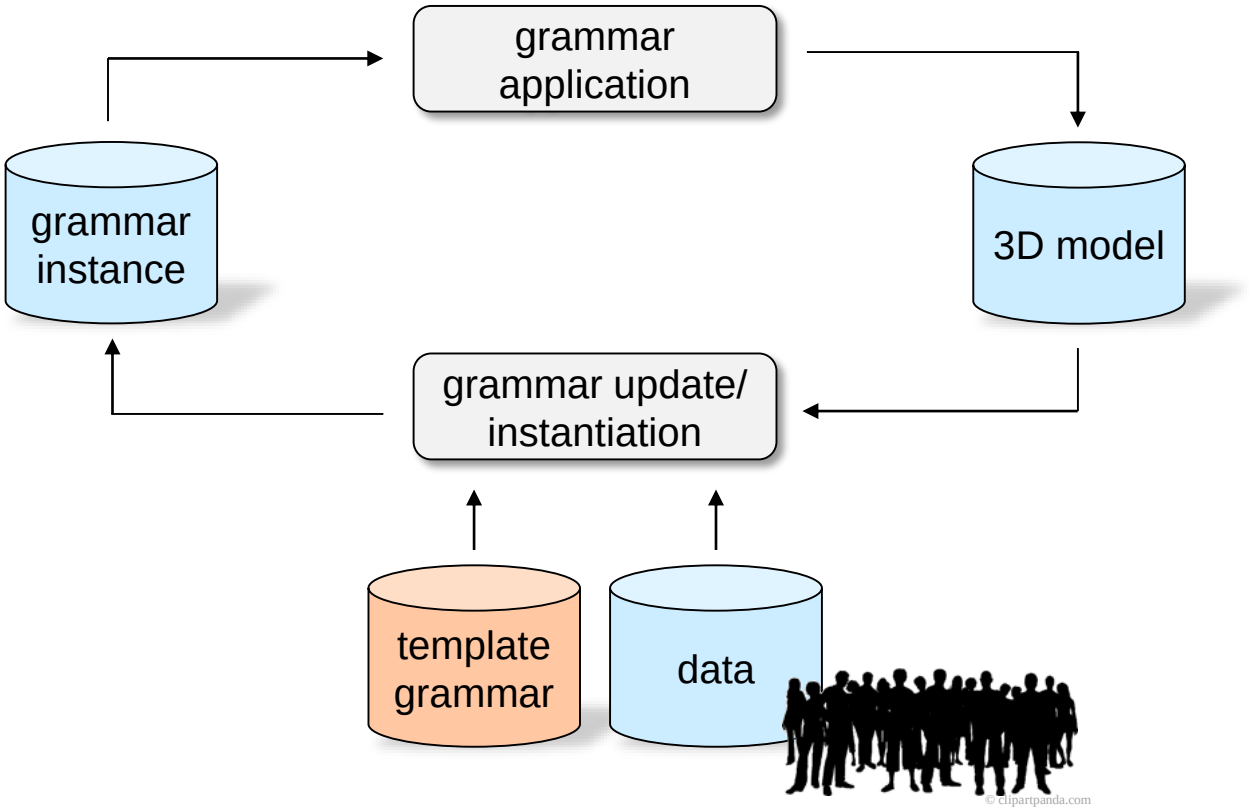
      with $* = Space \in$ previously generated Spaces
    - $R^{Merge} : Space^{\,l}\,Space^{\,r} \rightarrow Merge^{Space^{\,l},Space^{\,r}}$

      with $Merge^{Space^{\,l},Space^{\,r}} = Space^{\,l} \cup Space^{\,r}$
    - $R^{Instantiation} : Space \rightarrow space$



Iterative Grammar-Based Approach

- Iterative process of *grammar application* and *grammar update*



Instantiation of Individual Grammars

- $G^{indoor} = (G^{hallways}, G^{rooms})$
  - $G^{hallways}$

- ω: $\mathbf{R}(ACTIVE)?\mathbf{I}(\theta_{init}, UNASSIGNED)$
- p1: $\mathbf{R}(mode) > ?\mathbf{I}(\theta, state) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\{\textbf{\textit{LayoutSetting}}(mode, \theta)$ sets $\theta_p[0\text{-}4]\} \rightarrow +(\theta.angle)\mathbf{F}(\theta.len)$
  $\mathbf{B}^h(ACTIVE, \theta_p[1])\ \mathbf{B}^h(ACTIVE, \theta_p[2])\ \mathbf{B}^v(INACTIVE, \theta_p[3])$
  $\mathbf{B}^v(INACTIVE, \theta_p[4])\ \mathbf{R}(ACTIVE)?\mathbf{I}(\theta_p[0], UNASSIGNED)$
- p2: $\mathbf{R}(mode) > ?\mathbf{I}(\theta, state) : state==FAILED \rightarrow \varepsilon$
- p3: $\mathbf{B}^h(mode, \theta) : mode==ACTIVE \rightarrow [\mathbf{R}(\text{mode})?\mathbf{I}(\theta, UNASSIGNED)]$
- p4: $?\mathbf{I}(\theta, state) : state==UNASSIGNED$
  $\{\textbf{\textit{ConsistencyConstraints}}(\theta)$ adjusts $state, \theta\} \rightarrow ?\mathbf{I}(\theta, state)$
- p5: $?\mathbf{I}(\theta, state) : state!=UNASSIGNED \rightarrow \varepsilon$
- p6: $\mathbf{B}^v(mode, \theta) : mode==INACTIVE$
  $\{\textbf{\textit{ActivationControl}}$ sets $mode\} \rightarrow \mathbf{B}^v(mode, \theta)$
- p7: $\mathbf{B}^v(mode, \theta) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\rightarrow [\mathbf{Q}(\text{mode})?\mathbf{I}(\theta, UNASSIGNED)]$
- p8: $\mathbf{Q}(mode) > ?\mathbf{I}(\theta, state) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\{\textbf{\textit{LayoutSetting}}(mode, \theta)$ sets $\theta_p[0\text{-}3]\} \rightarrow +(\theta.angle)\mathbf{U}(\theta.len)$
  $\mathbf{B}^h(ACTIVE, \theta_p[1])\ \mathbf{B}^h(ACTIVE, \theta_p[2])\ \mathbf{B}^v(INACTIVE, \theta_p[3])$
  $\mathbf{R}(ACTIVE)?\mathbf{I}(\theta_p[0], UNASSIGNED)$
- p9: $\mathbf{Q}(mode) > ?\mathbf{I}(\theta, state) : state==FAILED \rightarrow \varepsilon$

  - G^{rooms}

- $R_i^{\text{SingleSplit}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_i, d_i)$
  with $Split^{Space}(\boldsymbol{n}_i, d_i) := Split_i^{Space} = Space^{\,l}\ Space^{\,r}$
- $R_i^{\text{RepeatSplit}} : Space \rightarrow Split_i^{Space^r} \circ \cdots \circ Split_i^{Space^r} \circ Split_i^{Space}$
- $R_{ij\ldots k}^{\text{StringSplit}} : Space \rightarrow Split_k^{Space^r} \circ \cdots \circ Split_j^{Space^r} \circ Split_i^{Space}$
- $R_{ab\ldots c}^{\text{MultiSplit}} : Space \rightarrow Split_c^{*} \circ \cdots \circ Split_b^{*} \circ Split_a^{Space}$
  with $* = Space \in$ previously generated Spaces
- $R^{\text{Merge}} : Space^{\,l}\ Space^{\,r} \rightarrow Merge^{Space^l, Space^r}$
  with $Merge^{Space^l, Space^r} = Space^{\,l} \cup Space^{\,r}$
- $R^{\text{Instantiation}} : Space \rightarrow space$

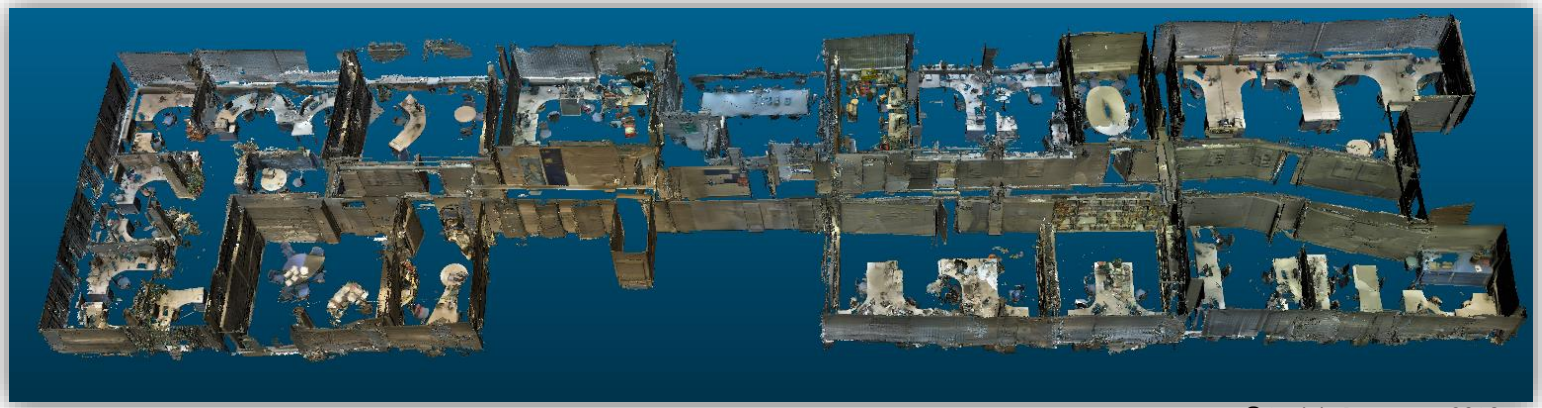
  - rules can stay fix

Instantiation of the L-System

Inverse Procedural Modeling

- To determine:
  - **layout parameters:** *hallway lengths, orientations, widths, probabilities*
  - **control image:** *2D probability distribution for vertical growth*

- Various observation data can be used as data source, e.g.:
  - traces from foot-mounted MEMS/IMU systems *(Philipp et al., 2014)*
  - 3D point clouds *(current work)*



Instantiation of Individual Grammars

- $G^{indoor} = (G^{hallways}, G^{rooms})$
  - $G^{hallways}$

- ω: $\mathbf{R}(ACTIVE)?\mathbf{I}(\theta_{init}, UNASSIGNED)$
- p1: $\mathbf{R}(mode) > ?\mathbf{I}(\theta, state) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\{\textbf{\textit{LayoutSetting}}(mode, \theta) \text{ sets } \theta_p[0\text{-}4]\} \rightarrow +(\theta.angle)\mathbf{F}(\theta.len)$
  $\mathbf{B}^h(ACTIVE, \theta_p[1])\ \mathbf{B}^h(ACTIVE, \theta_p[2])\ \mathbf{B}^v(INACTIVE, \theta_p[3])$
  $\mathbf{B}^v(INACTIVE, \theta_p[4])\ \mathbf{R}(ACTIVE)?\mathbf{I}(\theta_p[0], UNASSIGNED)$
- p2: $\mathbf{R}(mode) > ?\mathbf{I}(\theta, state) : state==FAILED \rightarrow \varepsilon$
- p3: $\mathbf{B}^h(mode, \theta) : mode==ACTIVE \rightarrow [\mathbf{R}(\text{mode})?\mathbf{I}(\theta, UNASSIGNED)]$
- p4: $?\mathbf{I}(\theta, state) : state==UNASSIGNED$
  $\{\textbf{\textit{ConsistencyConstraints}}(\theta) \text{ adjusts } state, \theta\} \rightarrow ?\mathbf{I}(\theta, state)$
- p5: $?\mathbf{I}(\theta, state) : state!=UNASSIGNED \rightarrow \varepsilon$
- p6: $\mathbf{B}^v(mode, \theta) : mode==INACTIVE$
  $\{\textbf{\textit{ActivationControl}} \text{ sets } mode\} \rightarrow \mathbf{B}^v(mode, \theta)$
- p7: $\mathbf{B}^v(mode, \theta) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\rightarrow [\mathbf{Q}(\text{mode})?\mathbf{I}(\theta, UNASSIGNED)]$
- p8: $\mathbf{Q}(mode) > ?\mathbf{I}(\theta, state) : state==SUCCEED \;\&\&\; mode==ACTIVE$
  $\{\textbf{\textit{LayoutSetting}}(mode, \theta) \text{ sets } \theta_p[0\text{-}3]\} \rightarrow +(\theta.angle)\mathbf{U}(\theta.len)$
  $\mathbf{B}^h(ACTIVE, \theta_p[1])\ \mathbf{B}^h(ACTIVE, \theta_p[2])\ \mathbf{B}^v(INACTIVE, \theta_p[3])$
  $\mathbf{R}(ACTIVE)?\mathbf{I}(\theta_p[0], UNASSIGNED)$
- p9: $\mathbf{Q}(mode) > ?\mathbf{I}(\theta, state) : state==FAILED \rightarrow \varepsilon$

  - rules can stay fix

  - G^{rooms}

- $R_i^{\text{SingleSplit}} : Space \rightarrow Split^{Space}(n_i, d_i)$
  with $Split^{Space}(n_i, d_i) := Split_i^{Space} = Space^{\,l}\ Space^{\,r}$
- $R_i^{\text{RepeatSplit}} : Space \rightarrow Split_i^{Space^r} \circ \cdots \circ Split_i^{Space^r} \circ Split_i^{Space}$
- $R_{ij\ldots k}^{\text{StringSplit}} : Space \rightarrow Split_k^{Space^r} \circ \cdots \circ Split_j^{Space^r} \circ Split_i^{Space}$
- $R_{ab\ldots c}^{\text{MultiSplit}} : Space \rightarrow Split_c^{*} \circ \cdots \circ Split_b^{*} \circ Split_a^{Space}$
  with $* = Space \in$ previously generated Spaces
- $R^{\text{Merge}} : Space^{\,l}\ Space^{\,r} \rightarrow Merge^{Space^l, Space^r}$
  with $Merge^{Space^l, Space^r} = Space^{\,l} \cup Space^{\,r}$
- $R^{\text{Instantiation}} : Space \rightarrow space$

  - rules have to be set for the split of

    → the 3D building shell into floors

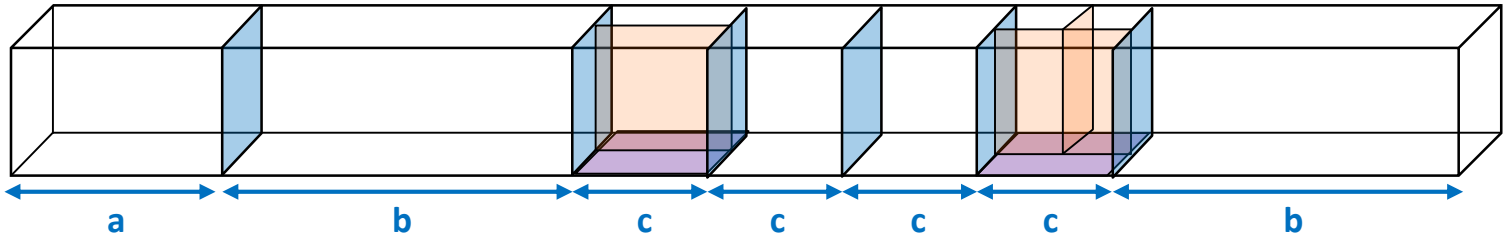
    → the floors into hallways and non-hallways

    → the non-hallways into rooms

Instantiation of the Split Grammar

Inverse Procedural Modeling

- Modeling room layouts



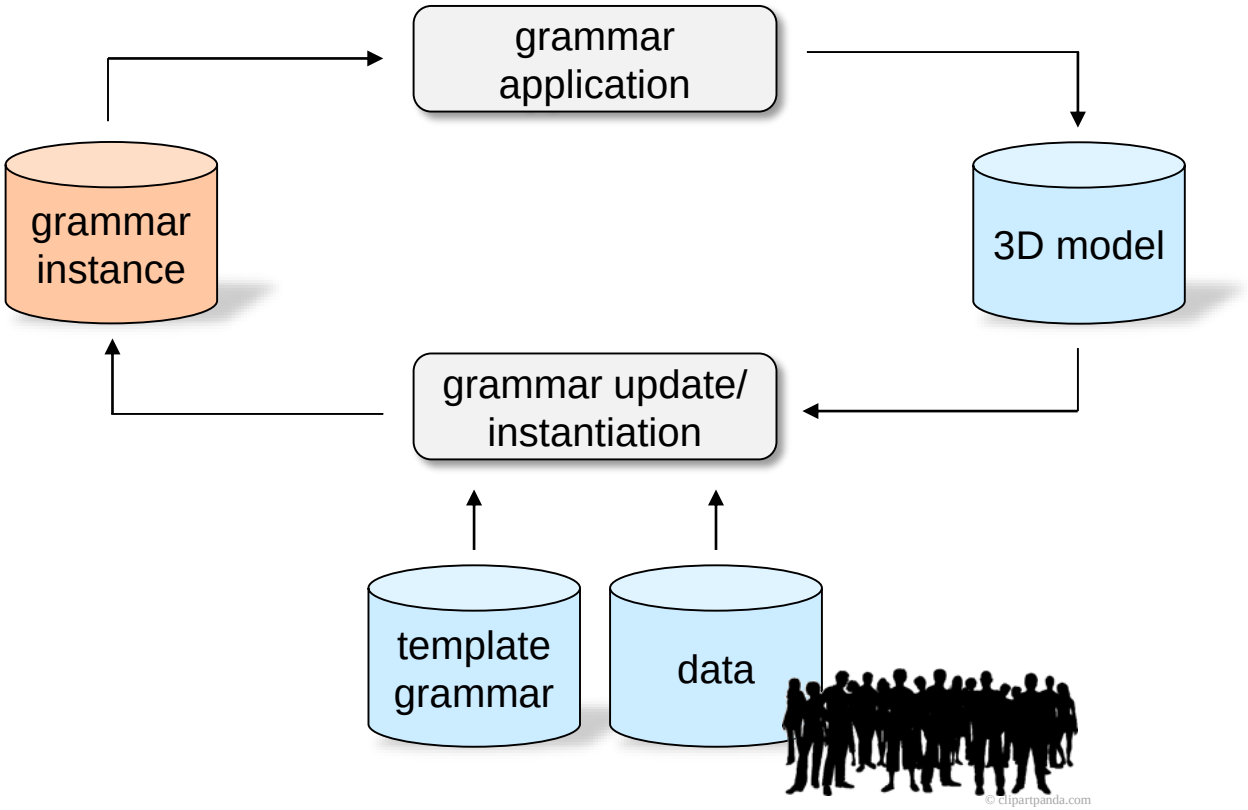
Split rules:

- $R_a^{Single} : Space \rightarrow Split^{Space}(\boldsymbol{n}_a | d_a)$
- $R_b^{Single} : Space \rightarrow Split^{Space}(\boldsymbol{n}_b | d_b)$
- $R_c^{Single} : Space \rightarrow Split^{Space}(\boldsymbol{n}_c | d_c)$
- $R_d^{Single} : Space \rightarrow Split^{Space}(\boldsymbol{n}_d | d_d)$
- $R_e^{Single} : Space \rightarrow Split^{Space}(\boldsymbol{n}_e | d_e)$

- $R_1^{Multi} : Space \rightarrow Split_d{}^{Space}$
- $R_2^{Multi} : Space \rightarrow Split_e{}^{Space^r} \circ Split_d{}^{Space}$
- $R_c^{Repeat} : Space \rightarrow Split_c{}^{Space^r} \circ \ldots \ldots \circ Split_c{}^{Space^r} \circ Split_c{}^{Space}$

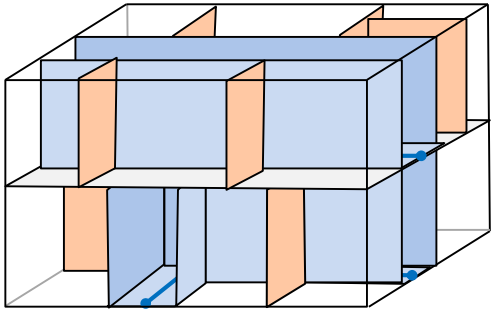
Iterative Grammar-Based Approach

- Iterative process of *grammar application* and *grammar update*



Grammar Application
Procedural Modeling

- Production process:
  - Automatic generation of *realistic* hypotheses about 3D indoor geometries

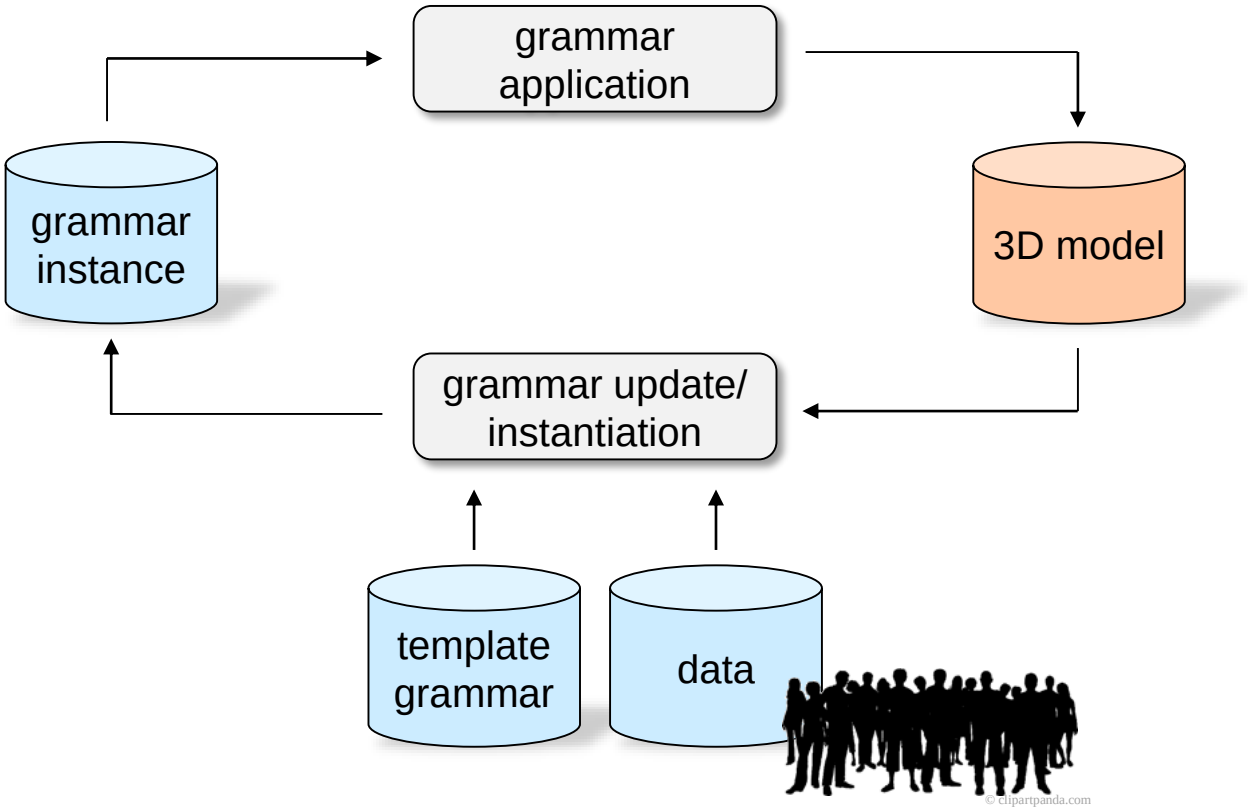


- L-System:
  - Rule application in sequential order
  - Termination when no branch can be further developed

- Split grammar:
  - *SingleSplits* for generating floors and **hallways**
  - Constraint-augmented random walk on a Markov Random Field for generating **room configurations**

Iterative Grammar-Based Approach

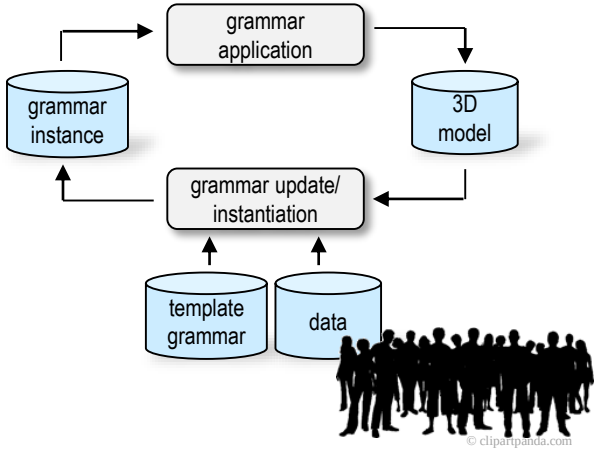
- Iterative process of *grammar application* and *grammar update*



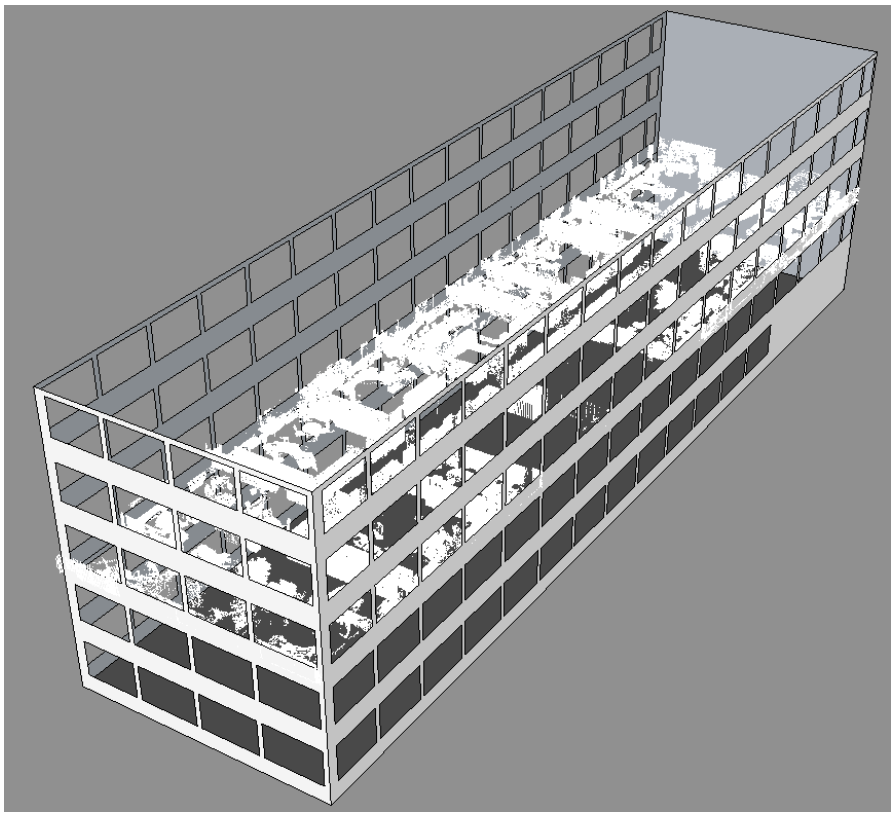
Iterative Learning and Verification Process

Results

- Seamless transition from LOD3 to LOD4



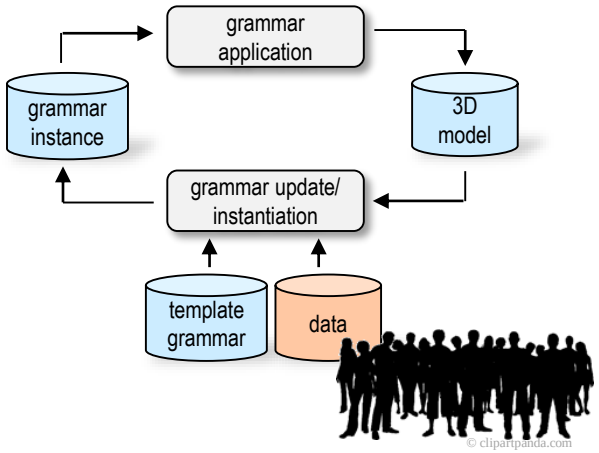
- Available data:



- **LOD3 model**
- **2D traces** in the 4th floor
- **3D point cloud** in the 4th floor

Iterative Learning and Verification Process
Results

- Seamless transition from LOD3 to LOD4



- Grammar instantiation:

Initial L-System:

- **lengths:** l_1, l_2, l_3
- **widths:** w_1, w_2
- **orientations:** α_1, α_2, α_3
- **probabilities**
- **control image**

Initial split grammar:

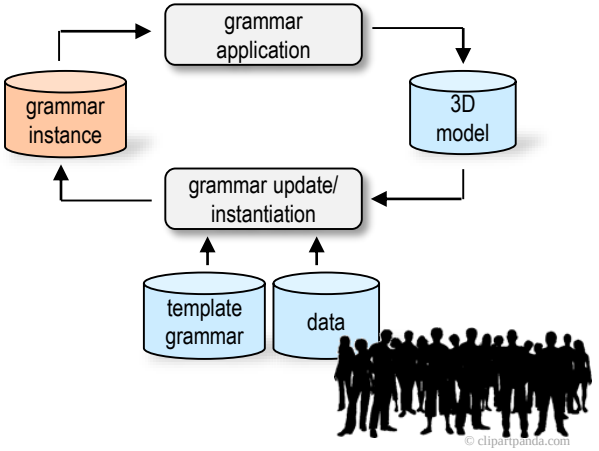
- $\mathbf{R_a^{Single}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_a \,|\, d_a)$
- $\mathbf{R_f^{Single}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_f \,|\, d_f)$

- Derivation of **initial L-System** from 2D traces
- Derivation of **initial split grammar** from LOD3 model
  - smallest room width $\gtrsim$ smallest window width
  - floor height $\approx$ vertical distance between two vertically arranged windows

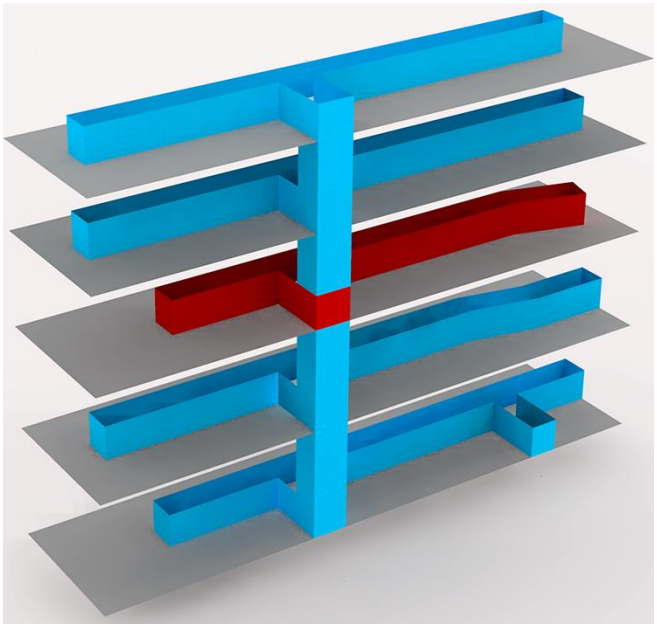
University of Stuttgart

Iterative Learning and Verification Process
Results

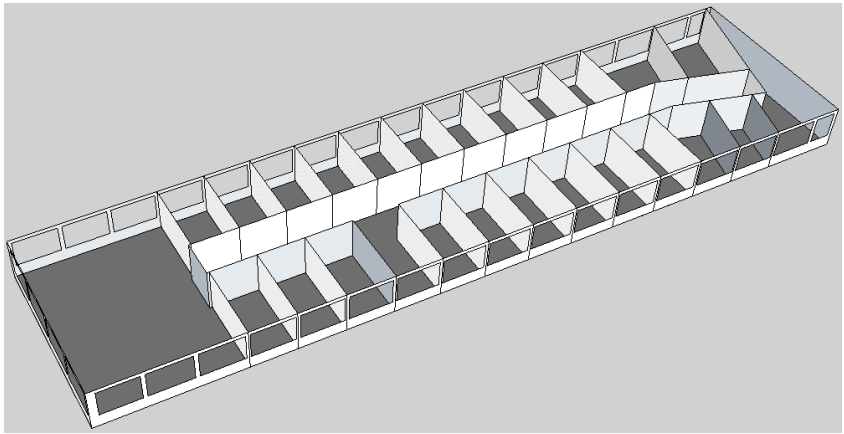
- Seamless transition from LOD3 to LOD4



- Grammar application:
  - **Initial split grammar** to generate **floors**
  - **Initial L-System** to generate **hallway spaces** and **non-hallway spaces**

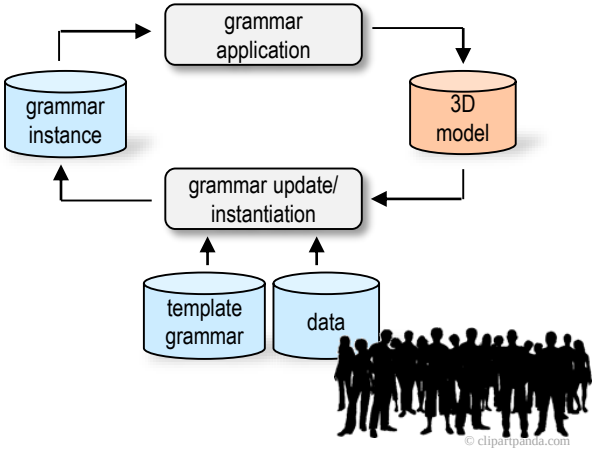


- **Initial split grammar** to generate **rooms** in the 4th floor



Iterative Learning and Verification Process
Results

- Seamless transition from LOD3 to LOD4

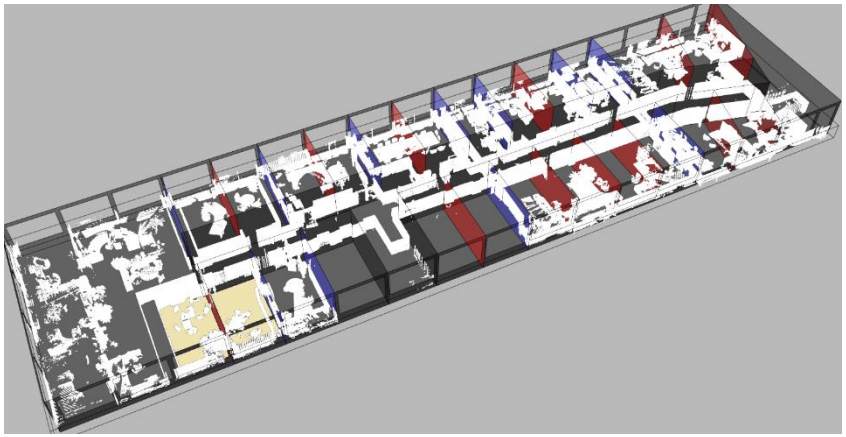


- Grammar update:
  - **Enhanced split grammar**

Enhanced split grammar:

- $\mathbf{R_a}^{\mathbf{Single}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_a | d_a)$
- $\mathbf{R_b}^{\mathbf{Single}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_b | d_b)$
- $\mathbf{R_c}^{\mathbf{Single}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_c | d_c)$
- $\mathbf{R_d}^{\mathbf{Single}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_d | d_d)$
- $\mathbf{R_f}^{\mathbf{Single}} : Space \rightarrow Split^{Space}(\boldsymbol{n}_f | d_f)$
- $\mathbf{R_{bc}}^{\mathbf{String}} : Space \rightarrow Split_c{}^{Space^r} \circ \ldots \ldots \circ Split_b{}^{Space}$
- $\mathbf{R_{cb}}^{\mathbf{String}} : Space \rightarrow Split_b{}^{Space^r} \circ \ldots \ldots \circ Split_c{}^{Space}$
- probabilities

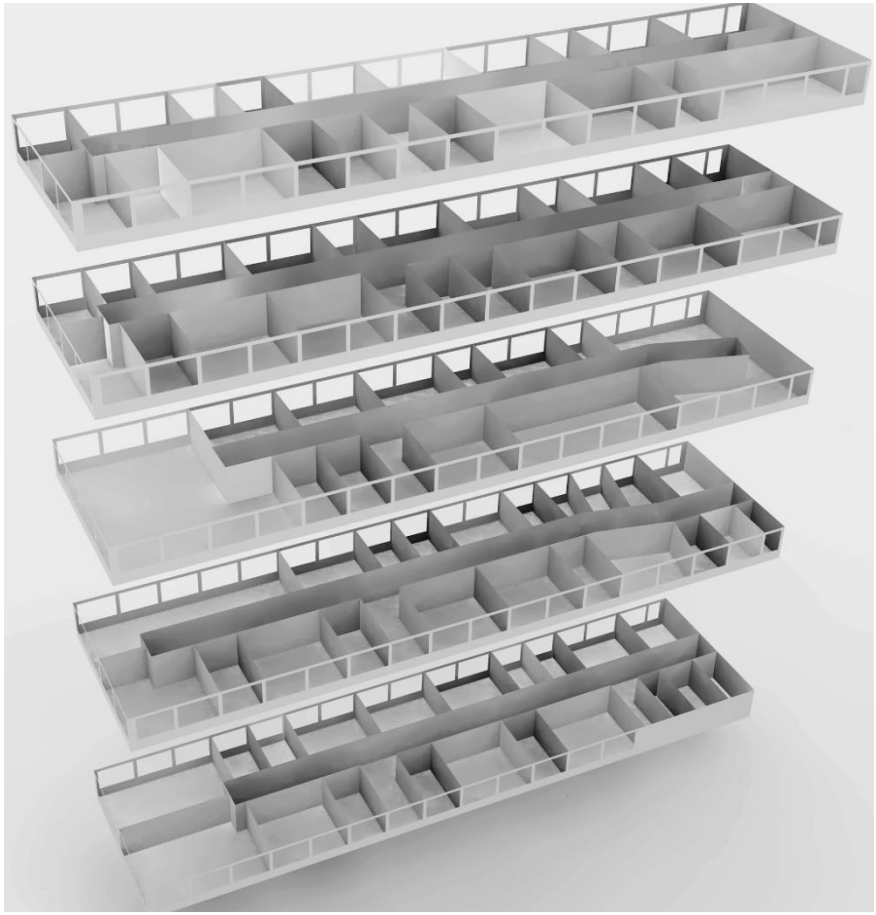
- Verification of 3D model

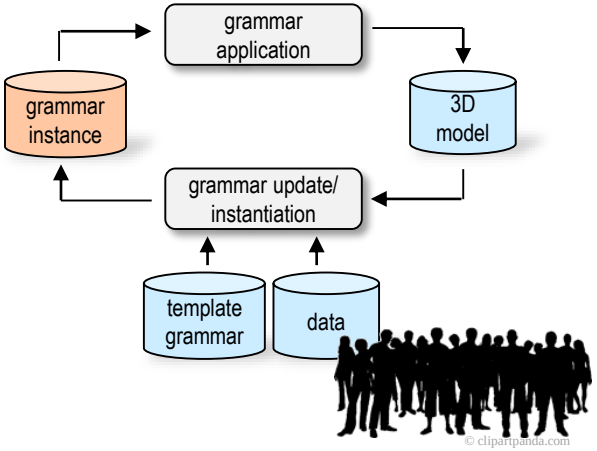


Iterative Learning and Verification Process

Results

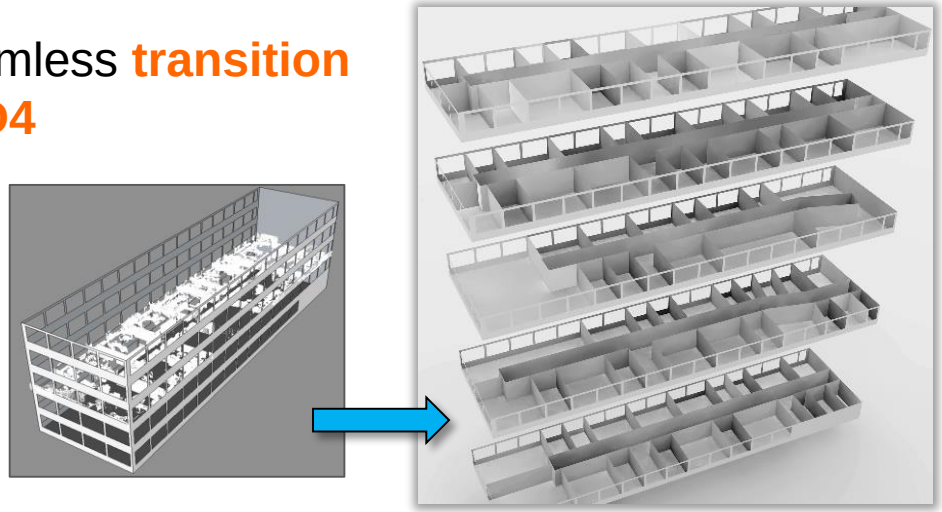
- Seamless transition from LOD3 to LOD4
- Grammar application:





Conclusions

- **3D indoor grammar** to support the reconstruction of building interiors from crowd-sourced sensor data
- **Individual grammars** can be **derived automatically** from observation data
- Grammar can be integrated in continuous **update and enhancement loop**
- **Robust** and **flexible** grammar-based “hypothesis and testing”-approaches
- Approach for a seamless **transition from LOD3 to LOD4**



References

- **Becker, S., 2009**. Generation and application of rules for quality dependent façade reconstruction. ISPRS J. Photogrammetry and Remote Sensing, 64, pp. 640-653.
- **Gröger, G. and Plümer, L., 2010**. Derivation of 3D indoor models by grammars for route planning. Photogrammetrie-Fernerkundung-Geoinformation, 2010(3), pp. 193-210.
- **Müller, P., Wonka, P., Haegler, S., Ulmer, A. and van Gool, L., 2006**. Procedural modeling of buildings. ACM Trans. Graph., 25(3), pp. 614-623.
- **Parish, Y.I.H. and Müller, P., 2001**. Procedural modeling of cities. In Proc. 28. Ann. Conf. Comp. Graph. and Interactive Techniques. SIGGRAPH '01, pp. 301-308.
- **Peter, M., Becker, S. and Fritsch, D., 2013**. Grammar Supported Indoor Mapping. In Proceedings of the 26th International Cartographic Conference, 1-18. Dresden, 2013.
- **Philipp, D., Baier, P., Dibak, C., Dürr, F., Rothermel, K., Becker, S., Peter, M. and Fritsch, D., 2014**. MapGENIE: Grammar-enhanced Indoor Map Construction from Crowd-sourced Data. Proceedings of the 12th IEEE International Conference on Pervasive Computing and Communications (PerCom 2014), Budapest, Hungary, March 24-28, 2014, pp. 1-9.
- **Prusinkiewicz, P. and Lindenmayer, A., 1990**. The algorithmic beauty of plants, Springer New York.
- **Stiny, G. und Gips, J., 1972**. Shape grammars and the generative specification of painting and sculpture. Information Processing, 71:1460–1465.
- **Wonka, P., Wimmer, M., Sillion, F. and Ribarsky, W., 2003**. Instant architecture. In ACM SIGGRAPH 2003 Papers, pp. 669-677.