DRUM – Distributed Transactional Building Information Management

Seppo Törmä, Jyrki Oraskari, Nam Vu Hoang
Distributed Systems Group
Department of Computer Science and Engineering
School of Science, Aalto University

Aalto University
School of Science

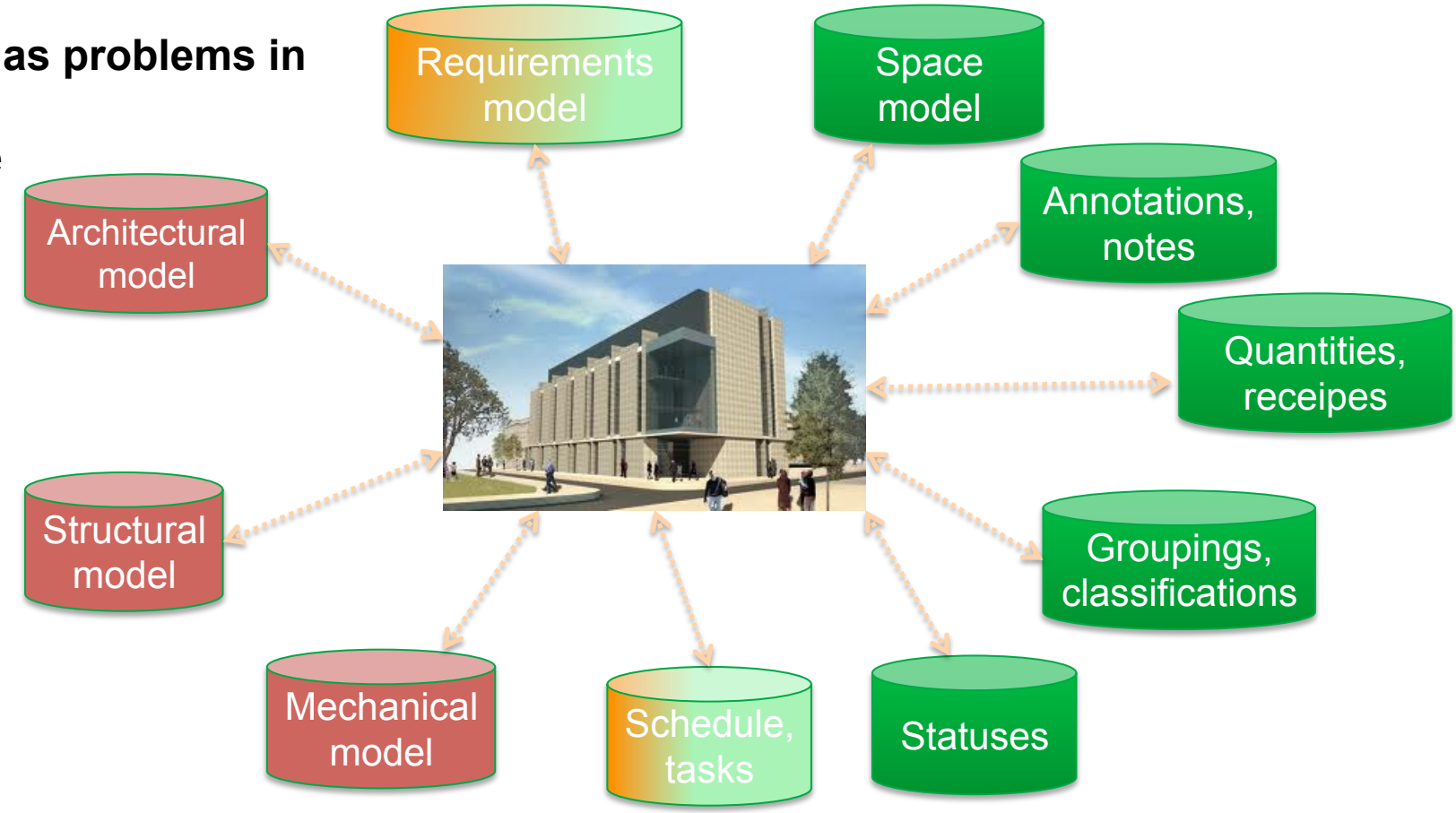
Contents

- Short version
  - Motivation
  - Core ideas
- Long version
  - Basic concept and design goals
  - Interlinking of partial models
  - The approach of linked project data
  - Research problems

Multiple sources of building information

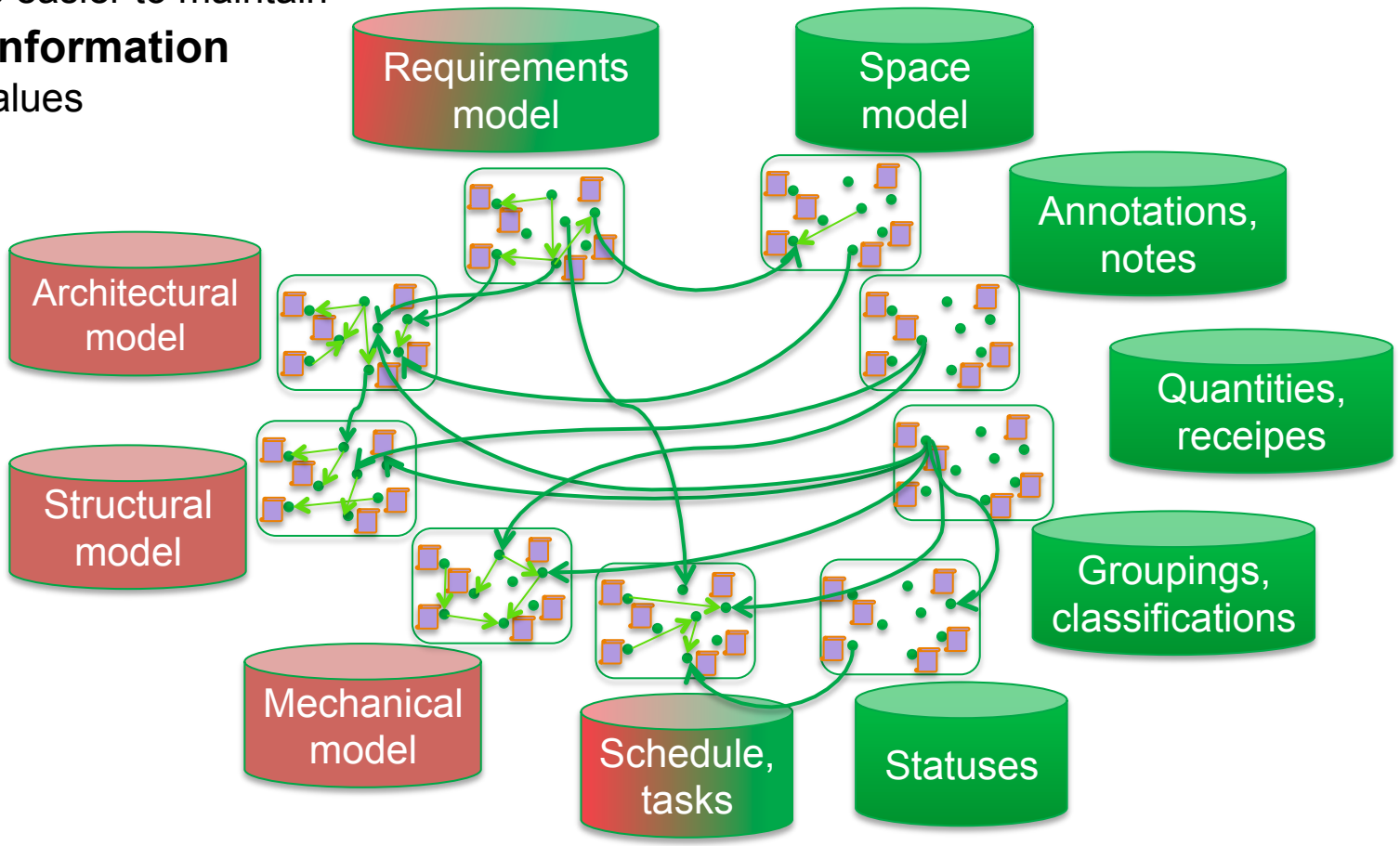
Different aspects of the same building
→Models need to be in agreement
Disagreement and conflicts result in
-redesign
-replanning
-rework
Manifest as problems in
-cost
-schedule
-quality

How to manage the relations of models?
- change management: detection, notification, change requests, …
Connecting a BIM to external information
- annotations, status, groupings, media, …



Interlinking the models

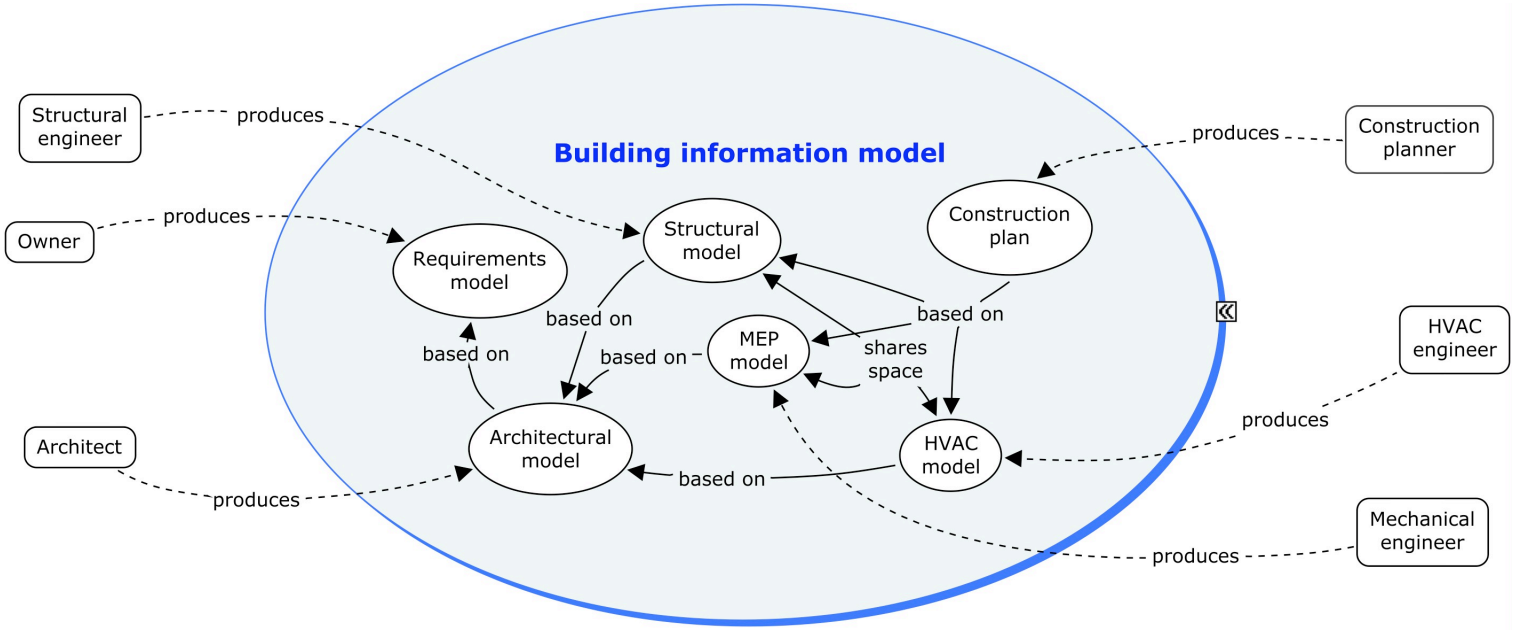
1. **Identify the IDs** (GUIDs, URIs, …)
2. **Link the IDs across models**
3. **Link the IDs within models**
   - fewer external links are needed
   - internal links easier to maintain
4. **Add other information**
   - attributes, values
   - geometry
   - text
   - …
5. **Provide an interface to access links**
   - find out links to other models
6. **Detect changes and their impact**
   - compute a diff between model versions
   - find out the links of the entities in the diff



Basic concepts and design goals

Partial models

- Discipline models (Rosenman), Writeable application views (Eastman), Aspect models (van Nederveen)



- Information producers → partial models
- Information consumers → views

Partial models

- Building information is created as partial models
  - Each partial model belongs to a particular discipline
  - Once created, partial models can be shared with others
- DRUM approach: Building information should also be maintained as partial models
  - Shared models should not be modified by parties that do not have discipline-specific expertize and tools
  - Incentives: Responsibility and control
  - Proper conversion roundtrip (native –> IFC –> native) cannot be guaranteed which means that model maintenance must be done in the native format

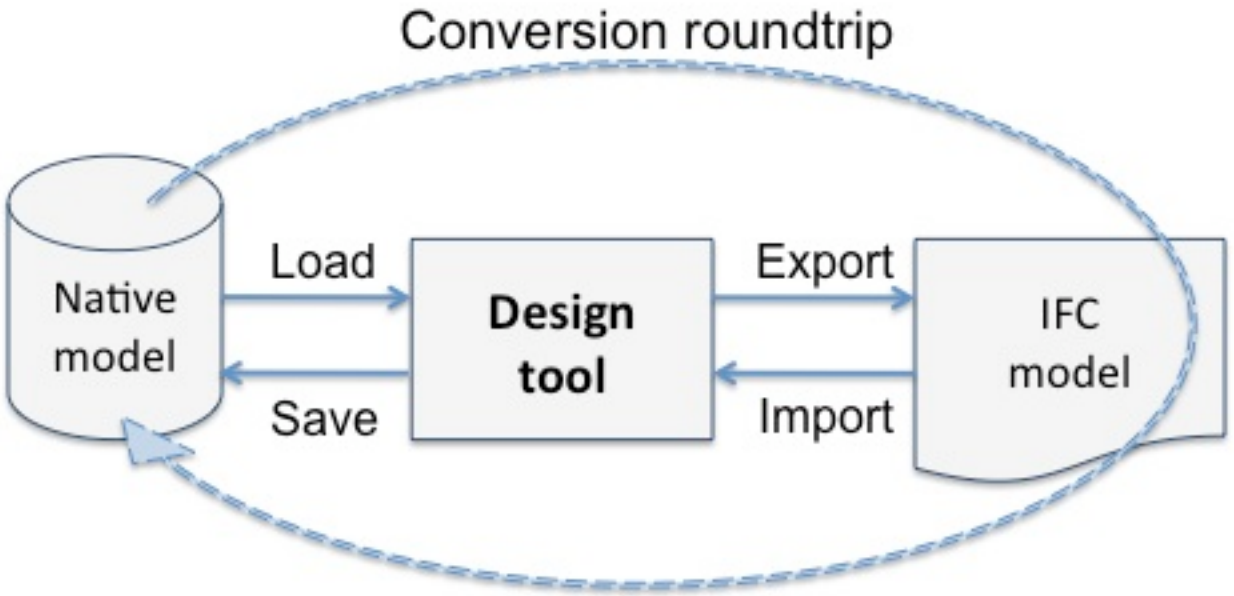
Partial models – Characteristics

- Characteristics
  - Produced by one party (or a team of tightly collaborating parties)
  - Produced using a same design tool
  - Require discipline-specific expertize
  - Have a well-understood use (or users)
  - Often relate to some well-defined phase
  - Have dense internal relations vs. relatively sparse relations with external entities
- Changes and evolution
  - Each of the models will face changes throughout a project
  - Changes propagate between models
- Partial models are
  - co-existing
  - co-evolving

Other issues affecting BIM solutions

- Forces of fragmentation
  - Lots of parties with widely differing capabilities
  - Short-term relations between companies
  - Different contractual structures
  - Contractual responsibilities, fear of disputes
  - Need to protect expertize
- Connections of BIM to relevant external data sources
  - Local building codes, zoning plans
  - Infrastructure models (LandXML)
  - Documents, spreadsheets, project plans, messages, …
  - Photographs and videos
  - Social networks for informal communication and collaboration
  - EPC/RFID material tracking systems

No proper conversion roundtrip



Importing an exported model looses information

• cannot be done in practice (not repetitively in any case)

Changes have to be done to the native model

• changes require support from the design tool

IFC models are read-only

Design goals

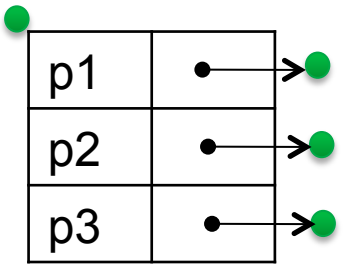
Require flexible representations

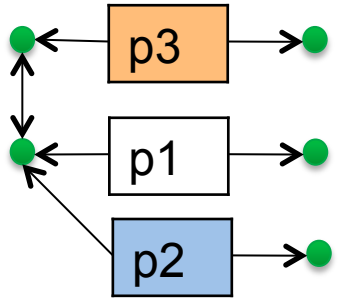
Require interlinking between models

1. Integrated information consumption
   - Each consumer has an integrated view to the information about same entities by different producers
2. Efficient information dissemination
   - Each consumer receives all the information by different producers relevant to its activities in a timely manner
3. Changes exclusively by producers
   - The producer of information has the exclusive right to make changes to it
4. Changes coordinated by producers
   - Changes are coordinated between all producers of affected information
5. Loose coupling
   - Adapts to the loosely coupled project consortiums in a same way across projects
6. Flexible deployment
   - Easy to adopt and deploy by parties at different levels of technical competence.
7. Extensible coverage
   - Supports the interlinking of the diverse kinds of information entities faced in a project.

Flexible distributed representations

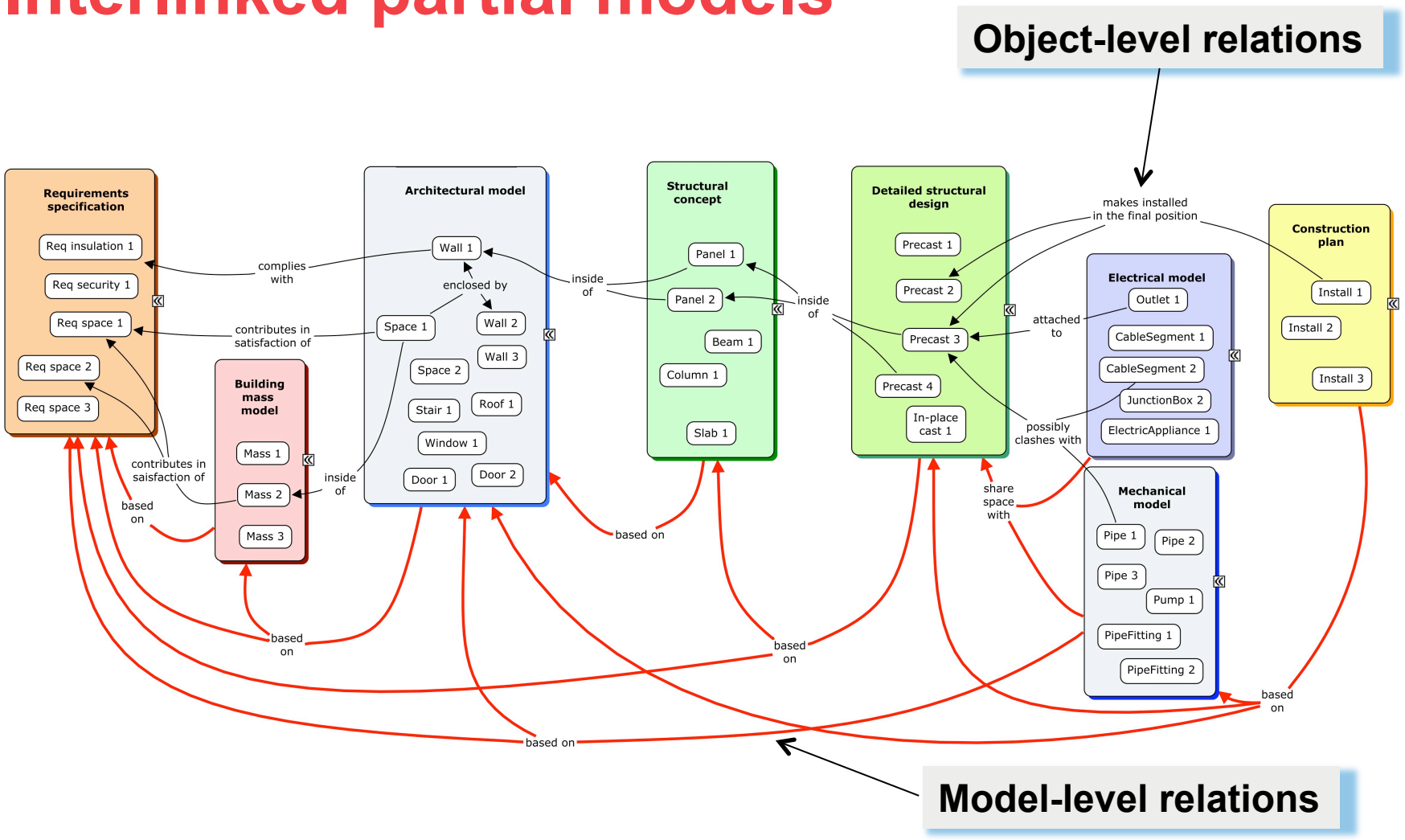
- Questions
  - How can other parties augment partial models created by others? E.g., with analysis information (energy, …)
  - How can information created by different parties be merged together?
- Object/entity granularity: too large and centralized
  - Example: IFC
  - Augmentation requires modification of objects
  - Who can add properties or property sets?
- Relation granularity: allows distributed representation
  - Example: RDF
  - Models can be augmented without modifying information produced by others
  - Properties can be added (1) just by referring to the identity of an object, or (2) referring to another identity and telling that it is same as the previous



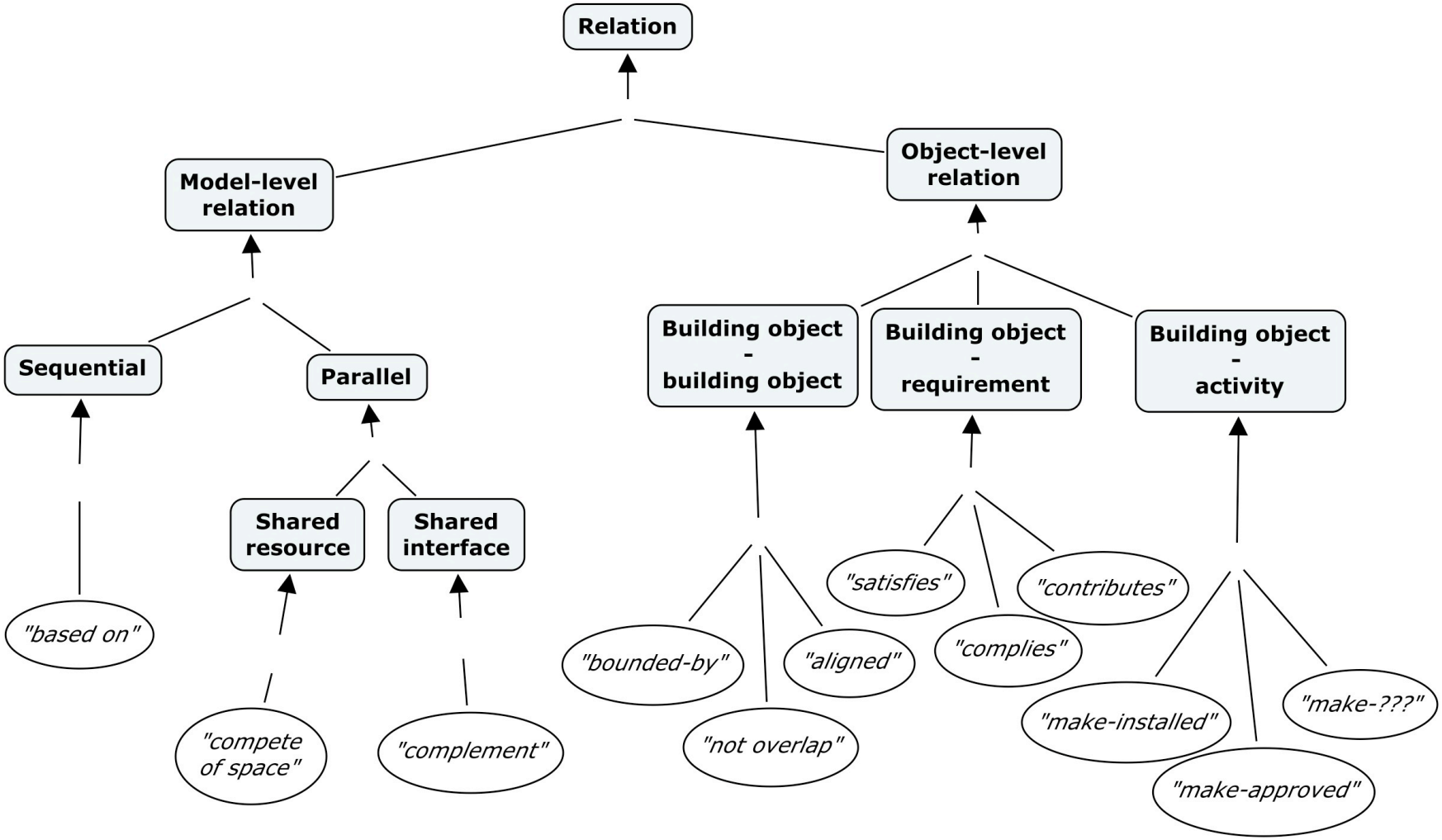


Interlinking of partial models

Interlinked partial models



Types of relations



Model-level relations

- Sequential: one model is based on another
  - Model m_2 is an elaboration of model m_1.
    - Model m_2 is more detailed or concrete with respect to m_1.
    - Each object in m_2 is bounded by some object in m_1.
    - Architectural – structural
- Parallel: competing or complementing
  - Model m_2 compete of same space with model m_1.
    - An indirect relation through a shared space that must be allocated to their building objects.
    - The over-allocation of the space results in spatial clashes.
    - No objects in m_2 must overlap with any object in m_1.
    - MEP model – HVAC model
  - Model m_2 represents adjacent zones with model m_1.
    - An indirect dependency through a shared interface
    - The entities at both sides should match: structures and openings should be aligned.
    - Every object in the interface of m_2 is aligned with corresponding objects in m_1.

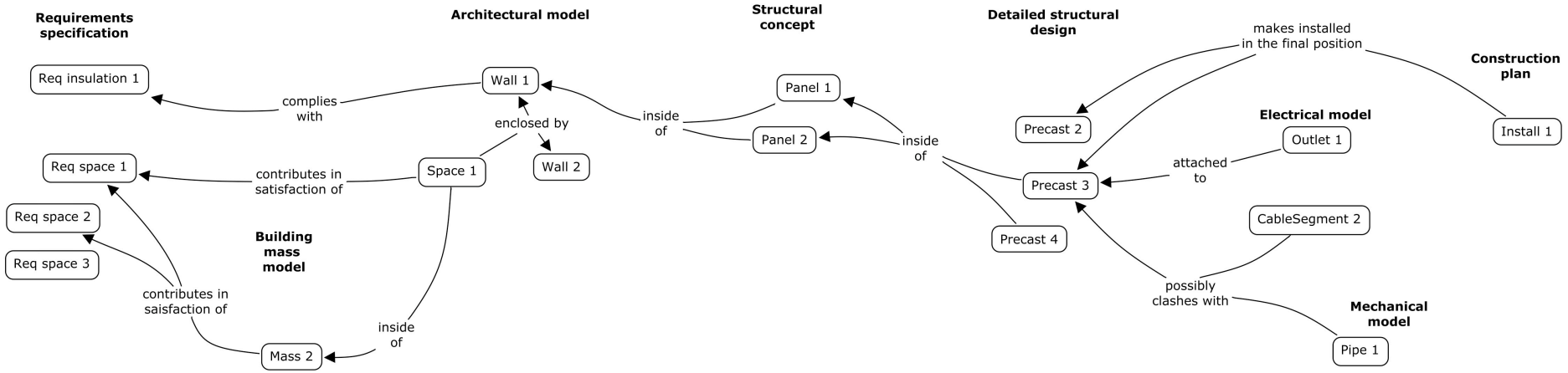
Object-level relations

- Building object b_1 – building object b_2
  - *bounds*(b_1, b_2) *aligned*(b_1, b_2) *overlaps*(b_1, b_2)
- Building object b_1 – requirement r_2
  - *satisfies*(b_1, r_2) *complies*(b_1, r_2) *contributes*(b_1, r_2) *breaks*(b_1, r_2)
- Building object b_1 – activity a_2
  - there are many activities for each entity
    - …, *move*(b_1, l_3), *install*(b_1), *approve*(b_1), …
  - relations of activities and entities are mediated by the states of entities
  - some activities can have a unique entity that is the object of the activity
    - but some do not: several objects (assembly), two objects (fastening), …
  - one activity is often performed to a group of entities
    - design, procurement, transportation, approval, ownership-transfer, …
  - a_2 = *installation-activity with*
    - *precondition*(a_2, *location*(b_1, l_3))
    - *postcondition*(a_2, *installed*(b_1))

Detecting relations

- Time of detection
  - As relation is established
  - As relations are queried
  - As relation is broken
- Manner of detection
  - Manual – Designer registers a relation
  - Semi-automatic – System proposes relations
  - Automatic – Relations are registered automatically
- Storage of relations
  - Internal or external to models?
  - Storage causes problems when changes happen
    - Hanging, missing, and wrong links

Linking in change management



- The network of all the objects in which a change can propagate
  - A change to requirements can flow all the way to the construction plan
- Change propagation inside a model (through internal relations)
  - Managed by the BIM tool and a designer
- Change propagation between models
  - Managed by DRUM

Levels of change management

1. *Reactive change propagation across models*: The other parties are notified about a change so that they can restore the consistency.
2. *Proactive change management protocols*: Collaborative protocols that make it possible to take into account the views of different parties affected by a change. There are different possible protocols based on change proposals, counterproposals, and so on.
3. *Transactional change management protocols*: Protocols that take the advantage of the distributed versioning capabilities of the participating models.

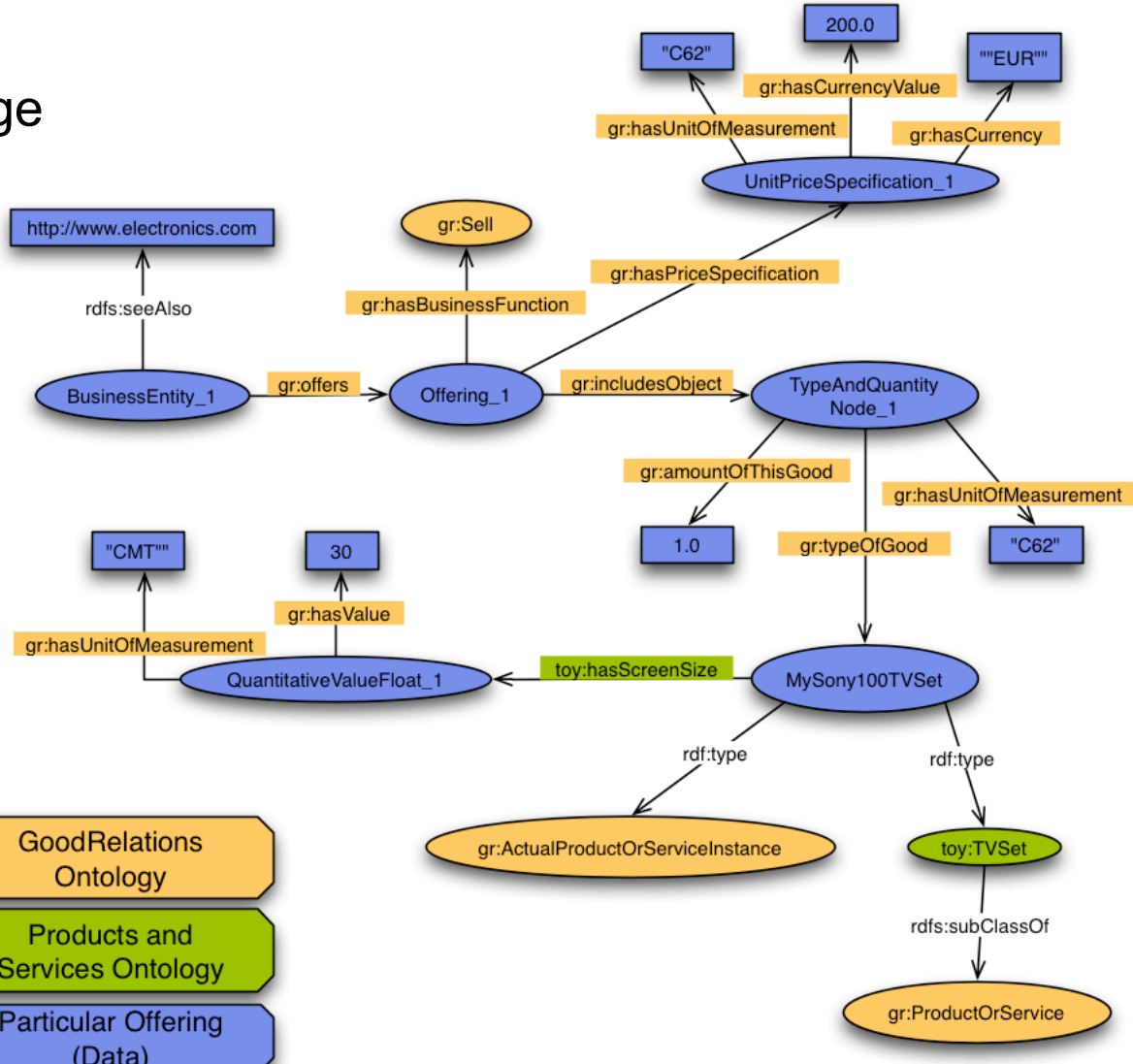
The approach of linked project data

Linked data

- A research area with practical solutions and tools to management of interrelated co-existing datasets
  - Compatible with many design goals above
- Many advantages
  - Lots of research and development activity
  - Large community
  - Numerous open source tools
  - Integrates well with familiar Web technologies
  - Connects well with other types of data
- Integration with Web helps to reposition IFC-based model sharing
  - IFC model as a hub that links together all other data available about the building and project

RDF – Resource Description Framework

- Graph-based
  - flexible, easy to merge
  - extensible
- Can use multiple schemas
  - ontologies or shared vocabularies
- Exposes objects
  - Each entity has an URI
- Linking
  - Across datasets
  - To other data



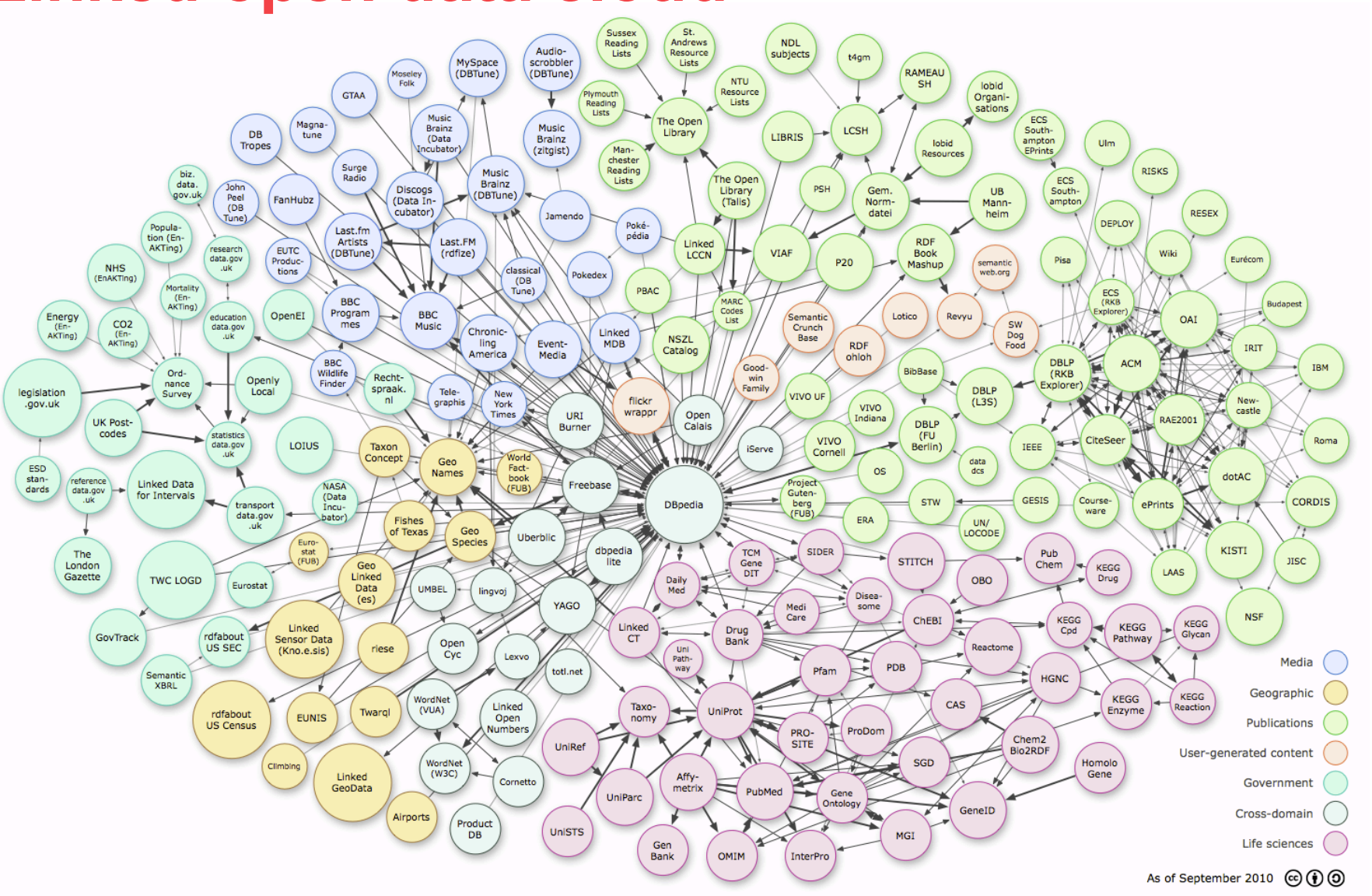
Linked data – Principles

- RDF data has not been browsable
  - Linked data principles aim to fix this problem
- Principles (Berners-Lee, 2006)
  1. Use URIs as names for things
  2. Use HTTP URIs so that people can look up those names
  3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL)
  4. Include links to other URIs, so that they can discover more things
- Other conventions
  - Mint URIs only in your own domain ➔ multiple URIs for an object
  - Use *owl:sameAs* to tell that two URIs mean the same object ➔ avoid problems of centralized identity management
  - Store your links to your own linkset in your own domain ➔ multiple linksets between datasets

Linked data – Other specifications

- Define schema (shared vocabularies) with ontology languages
  - OWL, RDFS
- XML Namespaces
  - Keep different terminologies separate from each other
- SPARQL – A SQL-like query language for RDF
  -
```
SELECT ?lat ?long WHERE {
    userXXX hasLocation ?location .
    ?location latitude ?lat .
    ?location longitude ?long .
}
```
- Emerging support for advanced reasoning
  - Rule frameworks: N3Logic, SPIN
- Linked Open Data Cloud – http://lod-cloud.net

Linked open data cloud



Aalto University
School of Science

"Linking Open Data cloud diagram, by Richard Cyganiak and Anja Jentzsch. http://lod-cloud.net/"

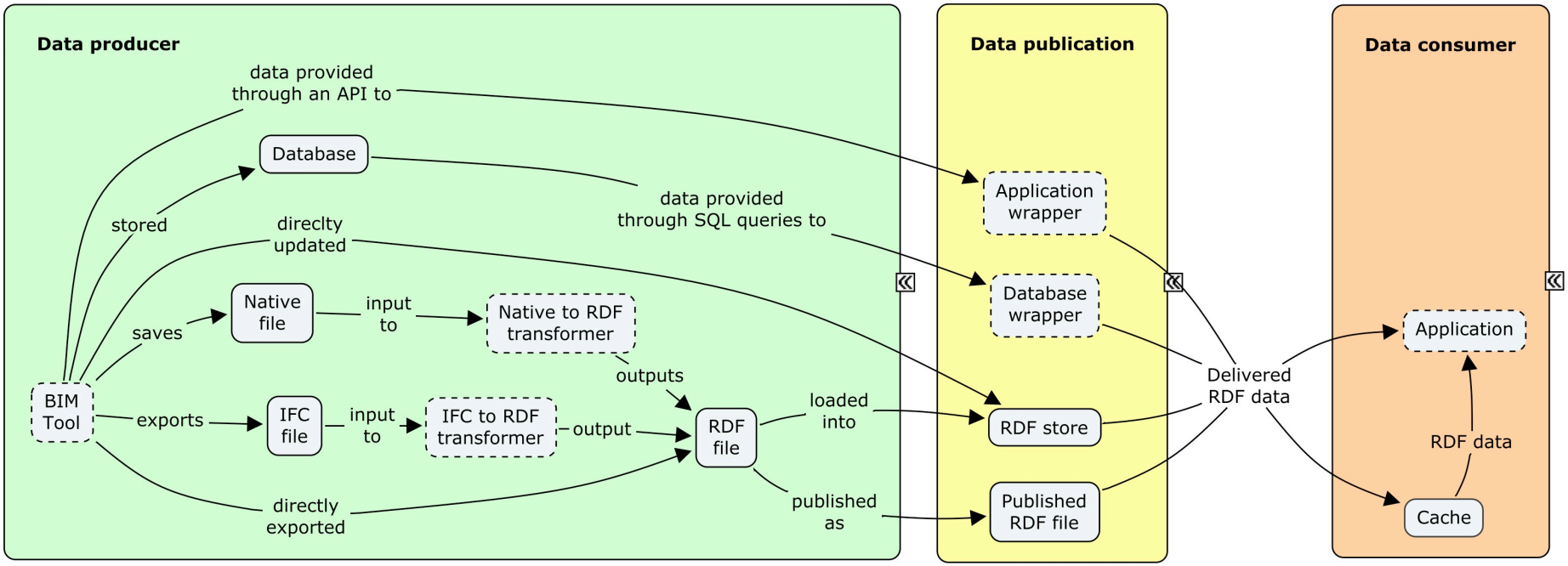
URI design

- Two possible schemes to provide a URI to an object identified with a GUID:
  - http://<company>/<project>/<guid>
  - http://<company>/<project>#<guid>
- URI can contain information about
  - Owner
  - Project
  - Dataset
- Examples
  - http://srv.fi/Musiikkitalo/1zm6Otovn1BOu4DngSQ1bi
  - http://ark-lpr.fi/Helsingin_Musiikkikeskus/0BqNaWRAvCrBzUXC5Y7SBM
- URI identity matching
  - http://ark-lpr.fi/Helsingin_Musiikkikeskus/0BqNaWRAvCrBzUXC5Y7SBM owl:sameAs http://srv.fi/Musiikkitalo/1zm6Otovn1BOu4DngSQ1bi;

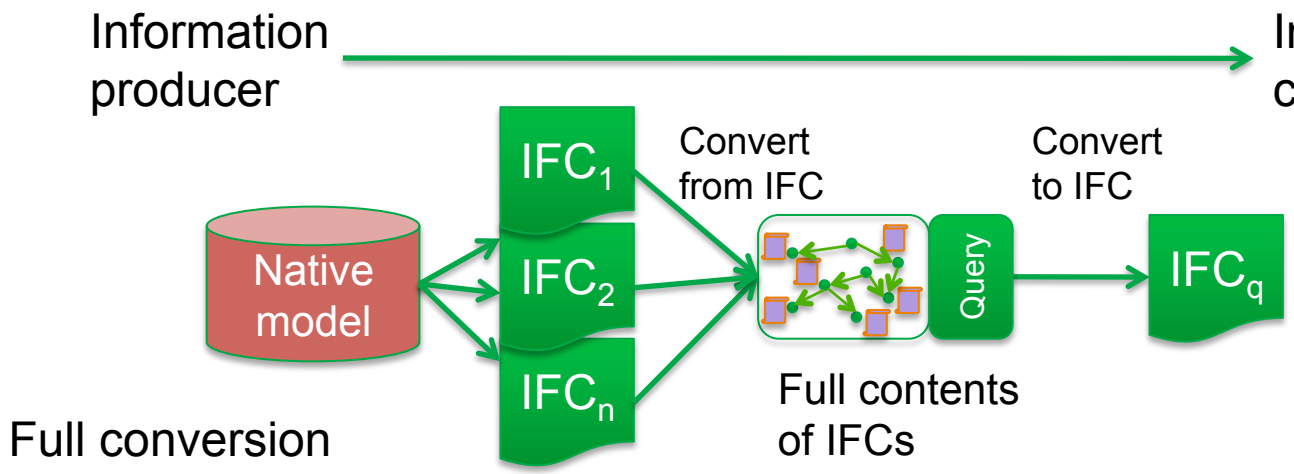
Linked data – Adaptation to BIM

- *Schema*: The IFC schema is translated from EXPRESS to OWL.
- *Identities*: The GUIDs in IFC models are translated into URIs.
- *Representation*: The IFC models are translated into RDF.
- *Publication*: Model is stored in an RDF store with SPARQL endpoint in the Web domain of the owner.
- *Dereferencing*: URIs that identify real-world entities will return RDF description when URI is looked up. Requires the implementation of the URI Lookup Interface.
- *Metadata*: Each model is described using a dataset description language (in XRD, POWDER-S, or voiD) to manage the collection of models belonging to a same project.
- *Linksets*: The links between different partial models are described using RDF triples. Each party that creates links also publishes them in their own linksets. There can be multiple linksets for each pair of models.
- *Security*: The security and authentication issues are solved using standard Web methods (Http Authentication and/or SSL/TSL).

Alternatives for publishing RDF data

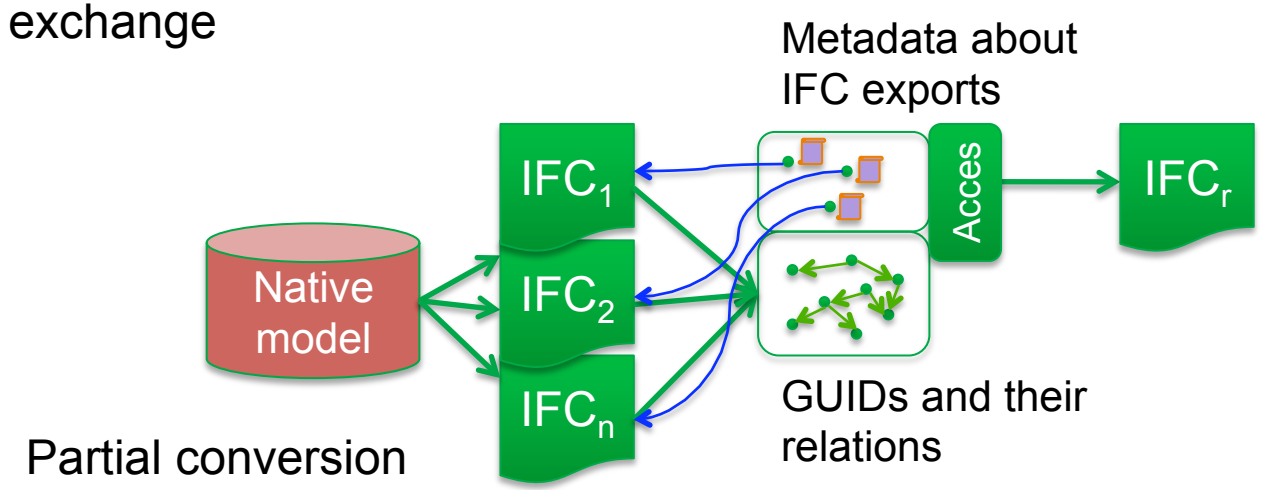


Exchange increments or complete files



Full conversion with increment exchange

Collaborative, increment-level, flexible, applicable if no fear of contractual disputes (same organization, IDP)



Partial conversion with file exchange

Controlled, file-level, static contents, better from the perspective of contractual and legal issues, emphasizes the responsibility of producers (DBB)

Link discovery – Tools for RDF models

- Apporaches
  - Key-based methods
    - same GUID, ISBN, …
  - Similarity-based methods
    - For places: string-similarity of names, closeness of locations, similar population count, …
- Advanced performance improvement techniques
  - from record linking field
  - pruning the pairs of entities that are compared (blocking)
- Automatic tools
  - SILK Link Discovery Framework
  - LIMES Link Discovery Framework
- Do not work well with simple IFC->RDF translation
  - Too much computation in each comparison
  - The absolute geometry is needed (e.g., axis representation)

Link discovery – BIM specific tools

- Solibri Model Checker
  - Cross-model checking can produce links
  - E.g., architectural model and structural model
- Tekla Structures
  - The architectural design can be taken as a background of structural design
  - Design work can produce links between architectural entities (a wall) and structural entities (a precast concrete element)
- Tekla BIMSight
  - Can visualize multiple models together
  - Link detection based on overlapping geometries

Dataset description

- Areas of metadata
  - General metadata: creator, time of modification, …
  - Structural metadata: patterns for resource URIs, ontologies used in the dataset, statistics about the size of the dataset, partitioning of a dataset, …
  - Access metadata: RDF data dumps, SPARQL endpoints, URI lookup endpoints, …
- Different languages
  - *XRD – Extensible Resource Descriptor* (by OASIS)
  - *POWDER – Protocol for Web Description Resources* (by W3C)
  - *voiD – Vocabulary of Interlinked Datasets* (by LOD community)
- Scope management in BIM
  - How to specify what datasets belong to same project?
  - How to specify the relations of the datasets? (based on, shares-space, …)

Dataset dynamics

- Frequency and extent of changes is dataset dependent
  - From sensor datasets to archival datasets
- In BIM there is a very particular type of dataset dynamics
  - The changes are partly planned and partly unexpected
  - There is a continuous strive for consistency between datasets
  - Changes propagate through the links from one dataset to another
- Link maintenance
  - Broken links – detection and removal or fixing
  - Detect the recreation of recently removed entities
    - Heuristic similarity
    - Popitsch, Haslhofer: DSNotify

Linked data - Evaluation

1.	Integrated information consumption	++
2.	Efficient information dissemination	+
3.	Changes exclusively by producer	+
4.	Changes coordinated by producers	+/-
5.	Loose coupling	++
6.	Flexible deployment	+
7.	Extensible coverage	++

BIM as Linked data – Research problems

- *IFC-RDF Bridge* – How to efficeintly convert from IFC to RDF and back. How the model should be represented in RDF to support linking?
- *Information scope management* – How to represent and manage information about what parties, datasets, and linksets belong to a project? How to implement in a distributed fashion?
- *Event notifications* – How to notify interested parties when information is modified? How to specify the interest of parties? How to detect changes?
- *Link type modeling* – Analysis and modeling of different types of links.
- *Link discovery* – Developing good methods and heuristics to discover links. How different solutions can work together? (links supplied by BIM tools, RDF link discovery tools, and manual discovery)
- *Change management* – When one of the interlinked datasets changes, compatible changes in related datasets are required. What kinds of change management protocols should be used? How transactions are handled?