Managing Geosemantic Diversity: Repositories & Patterns

Torsten Hahmann

National Center for Geographic Information & Analysis
School of Computing and Information Science
University of Maine

June 2, 2014

Outline

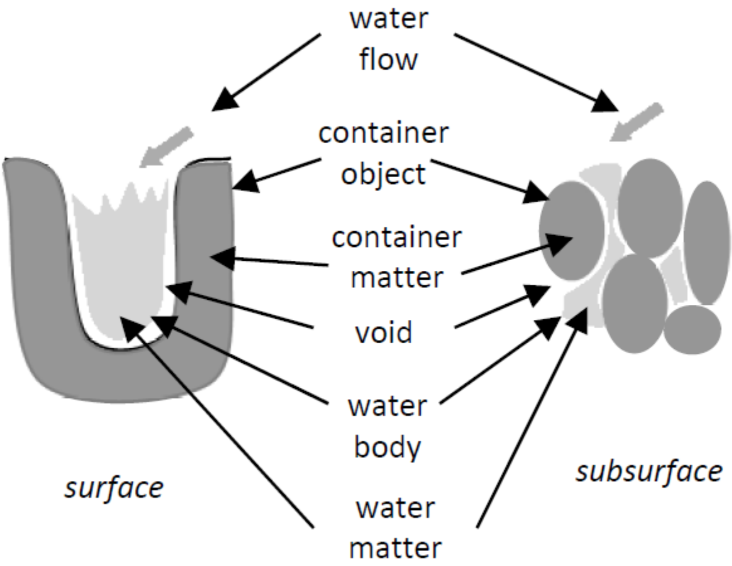
- Introduction of the Hydro Container Schema and, more generally, the Physical Container Schema
- Example 1 of a formal pattern: Physical Containment
- Example 2 of a formal pattern: Voids
- Other formal patterns in the Hydro Container Schema
- *Patterns in the Ontology Repository Formalism*

Developing a Hydro Ontology

Joint project with Boyan Brodaric (Geological Survey of Canada) and Michael Gruninger (University of Toronto) since 2012

- E-science flavor – **goal** is the precise modelling of a domain
- Formal treatise of key classes of entities and relations
- Can be thought of as developing a set of interrelated Ontology Design Patterns, more specifically: **Content Patterns (CP)**
- Formal approach using an expressive ("heavyweight") ontological language (Common Logic, similar to first-order logic)
  - Emphasis: Capturing structural knowledge, not data points
  - Purpose 1: Facilitates precise ontological analysis
  - Purpose 2: Ontology can be computationally verification
  - Result 1: **Formal, verified ontology** that can easily be reused
  - Result 2: Formalized, reusable pieces of ontology: **CPs**
- Drills much deeper than the usual **CPs** as **Conceptual Models**

Hydro Container Schema – Participating Entities

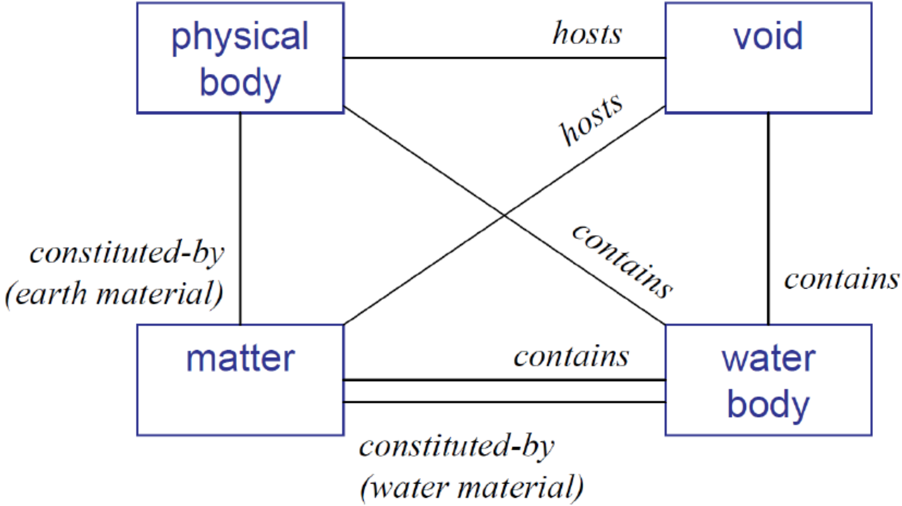


Hydro Container Schema specializes Physical Container Schema

Kinds of participating entities

- **Material Endurants**: physical objects and physical matter
- **Physical Objects**: water bodies and rock bodies
- **Physical Matter**: watter matter and rock matter
- **Physical Spaces**: voids
- **Physical Processes**: Water Flows

Hydro Container Schema – Key Relationships



Physical Container Schema

Kinds of relations between physical container schema participants

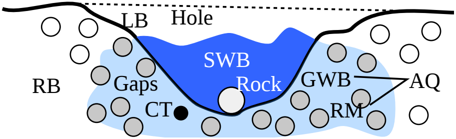
- **Containment**: among material endurants (objects, matter)
  - Also involving voids
- **Hosting**: between material endurants (objects, matter) and voids
- **Constituency**: between objects and matter
- **Dependency**: between/among material endurants and voids
- **Granularity**: between/among material endurants and voids
- **Flowing**: detailed version of Surface Hydrography Pattern (but including surface and subsurface and interconnecting water flows)

Approach

Idea: Use *formal ontological analysis* to better understand all involved entities and relations

- Formalize in an expressive logical language (Common Logic, which is similar to first-order logic but with a machine-readable encoding and some additional features for convenience)
  - Talk about *foreign*: even most computer scientists are afraid to deal with full-fledged logic; usually more comfortable with class diagrams
- Differentiate between different variants of each kind of entity and relations and thereby build a hierarchy of *formally distinguished* **subclasses** and **subrelations/subproperties**

Example 1: Kinds of Physical Containment Relations



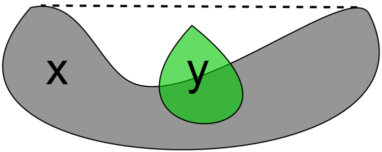
- The rock body contains rock matter
- The rock body contains a depression (a hole)
- The lake bed contains the lake
- The rock body contains (the same) lake
- The lake contains a rock
- The lake contains water
- The rock matter contains minuscule gaps
- The aquifer contains rock matter and water
- The aquifer and the ground water body contain a contaminant
- ...

What Exactly is Physical Containment?

Hahmann & Brodaric: Kinds of physical containment, Proc. of COSIT 2013

Containment, generally:

- Container schema (Kuhn, 2007): Binary relation between a container and a containee (inside or not); containee can enter and exit



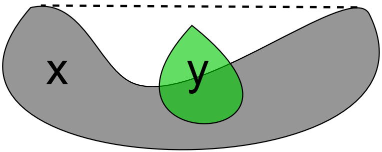
Spatially: $PO(y, x) \wedge P(y, \mathrm{ch}(x))$

What Exactly is Physical Containment?

Hahmann & Brodaric: Kinds of physical containment, Proc. of COSIT 2013

Containment, generally:

- Container schema (Kuhn, 2007): Binary relation between a container and a containee (inside or not); containee can enter and exit



Spatially: $PO(y, x) \wedge P(y, \mathrm{ch}(x))$

With physical typing:

$$mat(x) \rightarrow mat(y)$$
$$mat(x) \rightarrow fully\text{-}phys\text{-}contains(x, y)$$
$$V(x) \rightarrow \neg fully\text{-}phys\text{-}contains(x, y)$$

Containment, physically:

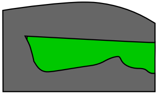
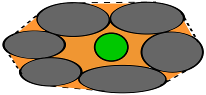
- Container and containee are physical endurants: either material endurants or voids (hosted by material endurants)

$\Rightarrow$ need to account for physical constraints

Distinctions between Full Containment Relations

Hahmann & Brodaric: Kinds of physical containment, Proc. of COSIT 2013

- Physical dependency between container and containee

dependent  vs.  detachable

Dependent containment: Without the containment relation at least one of the participating endurants would not exist in its present form

Detachable (non-dependent) containment: Each participating endurant can exist in its present form when separated

Distinctions between Full Containment Relations

Hahmann & Brodaric: Kinds of physical containment, Proc. of COSIT 2013

- Physical dependency between container and containee
- (Im)materiality of container
- (Im)materiality of containee
- Further distinctions: Parthood, Location, Enclosure, etc.

⇒ helps formally characterize the **most general variant of physical containment**, which is essentially a **CP** of physical containment that specializes the general container CP

⇒ identifies interaction between **physical containment pattern** and **physical (inter)dependence pattern**

Distinctions between Full Containment Relations

Hahmann & Brodaric: Kinds of physical containment, Proc. of COSIT 2013

- Physical dependency between container and containee
- (Im)materiality of container
- (Im)materiality of containee
- Further distinctions: Parthood, Location, Enclosure, etc.

⇒ helps formally characterize the **most general variant of physical containment**, which is essentially a **CP** of physical containment that specializes the general container CP

⇒ identifies interaction between **physical containment pattern** and **physical (inter)dependence pattern**

Distinctions between Full Containment Relations

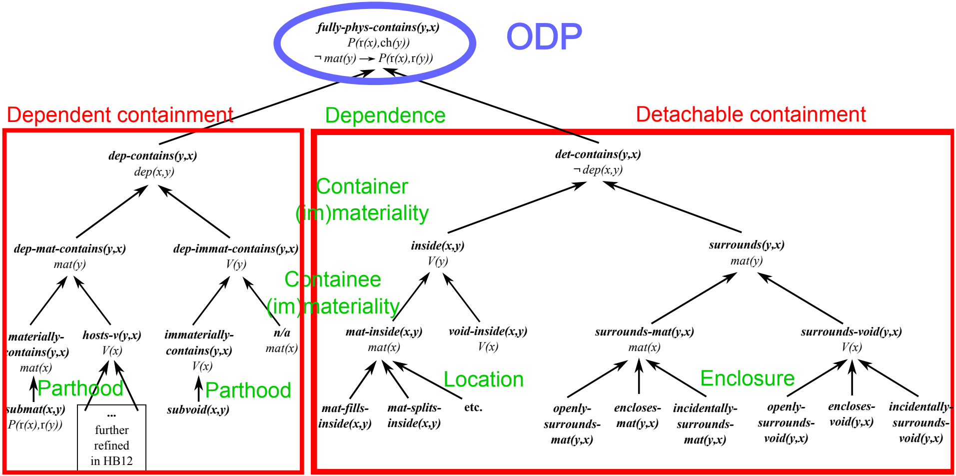
Hahmann & Brodaric: Kinds of physical containment, Proc. of COSIT 2013

- Physical dependency between container and containee
- (Im)materiality of container
- (Im)materiality of containee
- Further distinctions: Parthood, Location, Enclosure, etc.

⇒ helps formally characterize the **most general variant of physical containment**, which is essentially a **CP** of physical containment that specializes the general container CP

⇒ identifies interaction between **physical containment pattern** and **physical (inter)dependence pattern**

Physical Containment Relation as an ODP



$$\textit{fully-phys-contains}(x,y) \leftrightarrow \quad PED(x) \wedge PED(y) \wedge P(\mathrm{r}(x), \mathrm{ch}(y))$$

$$\wedge \left[\neg mat(y) \rightarrow P(\mathrm{r}(x), \mathrm{r}(y))\right]$$

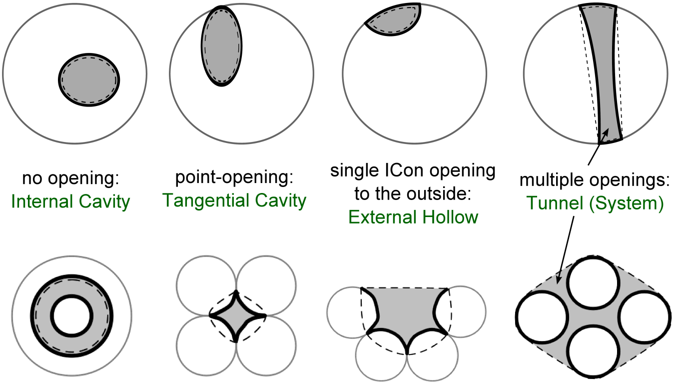
Example 2: Kinds of Voids

Casati & Varzi: Holes, 1994;

Hahmann & Brodaric: The Void in Hydro Ontology, Proc. of FOIS 2012

Holes vs. Gaps: based on whether the host is internally self-connected

Cavities vs. Tunnels vs. Depressions: based on the void's opening (to the outside or to other voids)



Voids – Formalization in Common Logic (Excerpt)

(V-D) $V(x) \leftrightarrow \exists y[\mathit{hosts\text{-}v}(y,x)]$ (all voids are hosted)

(V-A1′) $\mathit{hosts\text{-}v}(y,x) \rightarrow \mathit{PED}(y) \wedge \neg V(y) \wedge V(x) \wedge P(\mathrm{r}(x), \mathrm{ch}(y)) \wedge C_{\mathrm{S}}(\mathrm{r}(x), \mathrm{r}(y)) \wedge \mathit{PO}(\mathrm{r}(x), \mathrm{r}(y))$

(hosting a void relation)

(V3) $\mathit{hosts}(x,y) \wedge V(y) \rightarrow \neg V(x)$ (voids cannot host voids)

(V6) $\mathit{hosts\text{-}v}(x,y) \wedge \mathit{hosts\text{-}v}(x,z) \wedge \mathit{PO}(y,z) \rightarrow y \subseteq z \vee z \subseteq y$

(V11) $\mathit{hosts\text{-}v}(x,v) \rightarrow \mathrm{op}(x,v) = \mathrm{r}(v) \cdot (\mathrm{r}(x) + \mathrm{r}(v))'$

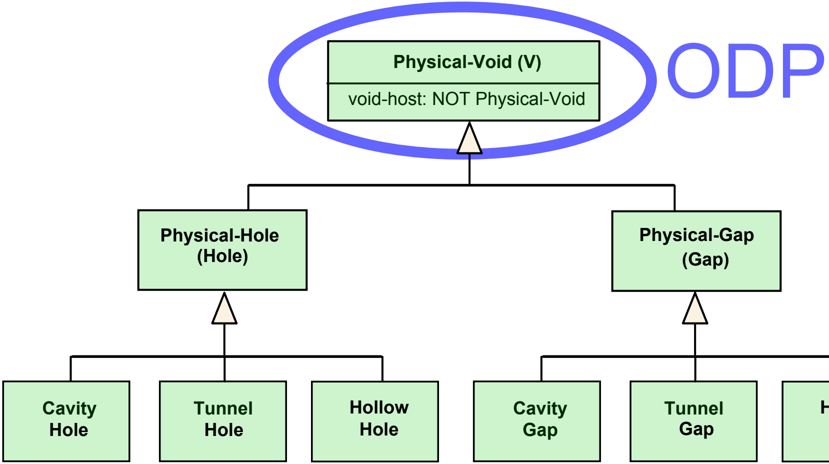
(the opening of a void: boundary not shared with its host)

(V_{S}-D) $V_{\mathrm{S}}(y) \equiv \mathit{ICon}(y) \wedge \exists x[\mathit{hosts\text{-}v}(x,y)]$ (simple void)

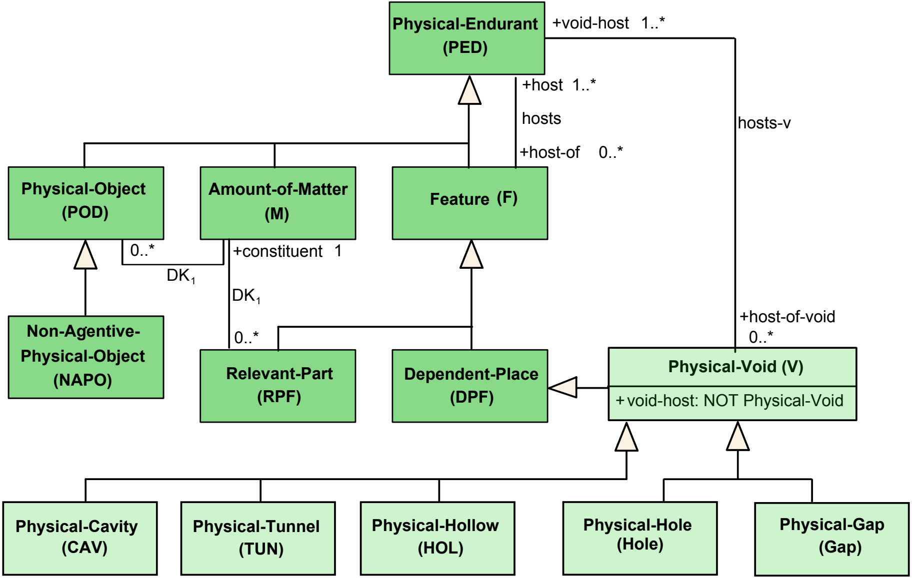
(V_{C}-D) $V_{\mathrm{C}}(y) \equiv \neg \mathit{ICon}(y) \wedge \exists x[\mathit{hosts\text{-}v}(x,y)]$ (complex void)

(V-A28) $V_{\mathrm{S}}(x) \rightarrow \exists y[\mathit{hosts\text{-}v}(y,x) \wedge \mathit{ICon}((\mathrm{r}(x) + \mathrm{r}(y))')]$

Concept of a Void as an ODP



Voids in the DOLCE Classification of Physical Endurants



Formalizations and Analysis of Other Relations in the Physical Container Schema: Interdependence

Physical Interdependence

Hahmann et al.: Interdependence among material objects and voids, Proc. of FOIS 2014 (to appear)

- A spatial and material characterization of when two material endurants an/or voids are physically interdependent: cannot separated without at least one of them changing in its form
- How physical interdependence specializes general ontological dependence

Formalizations and Analysis of Other Relations in the Physical Container Schema: Constitution and Granularity

Granularity and constitution among material endurants and voids

Hahmann & Brodaric: Voids and material constitution across physical granularities, Proc. of FOIS 2014 (to appear)

- Formal framework for comparing granularities:
  - a rock vs.
  - an amount of rock matter vs.
  - a atomic/molecular structure vs.
  - a subatomic structure
- The role of voids in identifying differences in granularity
- Constitution as dependent containment with change in granularity

Outline

- Introduction of the Hydro Container Schema and, more generally, the Physical Container Schema
- Example 1 of a formal pattern: Physical Containment
- Example 2 of a formal pattern: Voids
- Other formal patterns in the Hydro Container Schema
- Patterns in the Ontology Repository Formalism
  - The formalizations give rise to hierarchies of ontology modules
  - These hierarchies form an ontology repository
  - We can extract formalized CPs from this repository

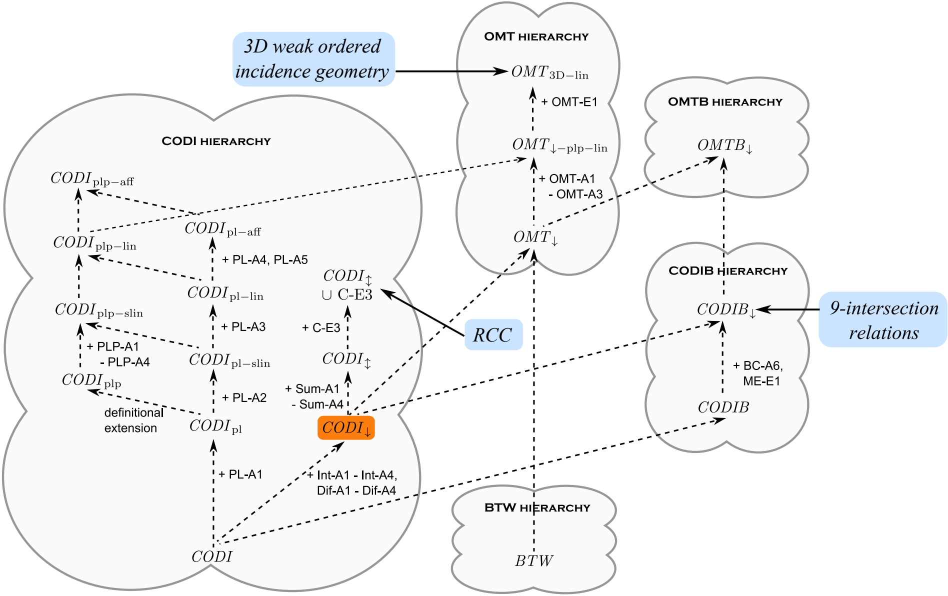
Using COLORE to identify Content Patterns

COLORE: COmmon Logic Ontology REpository: `colore.oor.net`

- Over 1300 ontology modules (all being ontologies themselves)
- Set of interrelated **hierarchies of ontologies** that use a common set of undefinable vocabulary terms
  - Acknowledges that there are multiple interpretations of the same terms (built-in variability)
  - Hierarchies **ordered** by vocabulary extensions (introducing new terms)
  - Within hierarchies, ontologies **ordered** by axiomatic extensions (introducing new assumptions)
  - Distinction between mathematical and generic (content) hierarchies
  - **Ordering**: think of as mappings
- Discussed in details in a series of papers by *Gruninger et al.*

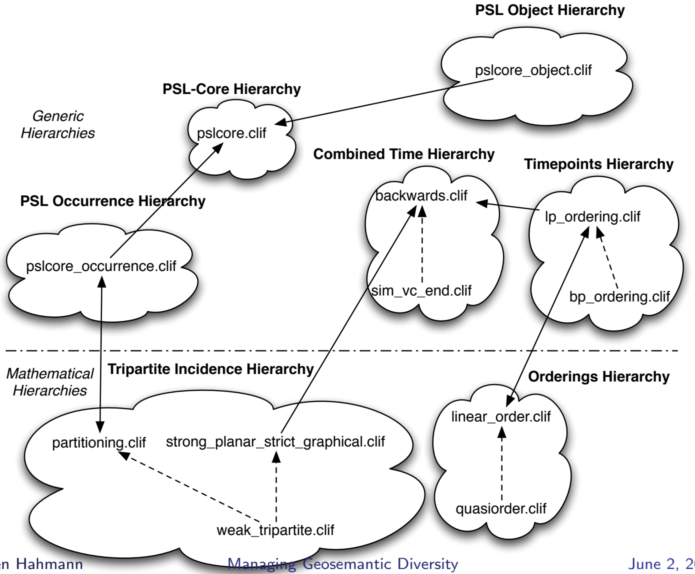
  A Sideways Look at Upper Ontologies, Proc. of FOIS 2014 (to appear);
  Modular First-Order Ontologies via Repositories, Applied Ontology 7(2), 2012;
  Ontology verification with repositories, Proc. of FOIS 2010;
  Exploiting Modularity for Ontology Verification, Proc. of WoMo 2011

Ontology Repository: COLORE



Using COLORE to identify Content Patterns

Hahmann: Ontology Repositories – A Treasure Trove for Content Ontology Design Patterns, submitted to WoMO 2014



Using COLORE to identify Content Patterns

Hahmann: Ontology Repositories – A Treasure Trove for Content Ontology Design Patterns, submitted to WoMO 2014

Corollary

The root theory of every irreducible generic hierarchy in an ontology repository is a formalized CP.

Corollary

All operations on CPs (import, clone, specialization, generalization, composition, expansion) are logical relationships in the repository.

Corollary

Every theory in a mathematical hierarchy in an ontology repository is a knowledge pattern.

Clark et al.: Knowledge Patterns, Handbook of Ontologies, 2003

Takeaway & Future Directions

- Want to demonstrate how the entire pipeline can work: conceptual model, modularization, expressive formalization, integration (using the repository), verification, and vocabulary extraction
- Need to **Fully Verify** the integrated hydro ontology that combines the different pieces/patterns that have been verified individually
- Need to find ways to **Extract a Lightweight Vocabulary** of the hydro ontology, e.g., in OWL or RDF (or both)
- Vocabulary terms in the formal ontology are not **linguistic terms**: need mapping mechanism to deal with linguistic ambiguity

Takeaway & Future Directions

More objective & scientific way to create **Reusable** Content Patterns

- Driver for e-Science initiatives addressing interoperability issues
- Getting away from ad-hoc ontology & pattern development
- Requires close collaboration between ontology & domain experts
- Helps to really understand domain concepts and their interaction
- Currently labor-intensive
- Want to automate as many steps as possible
- Need better visualization (forget about Protege)
  $\Rightarrow$ Work with Autodesk Research (*AutoCAD, Maya*)