Modeling the Transitional Organization

Christian Kaul

DDVUG-Herbsttagung, 24.10.2019

Who Is This Guy?

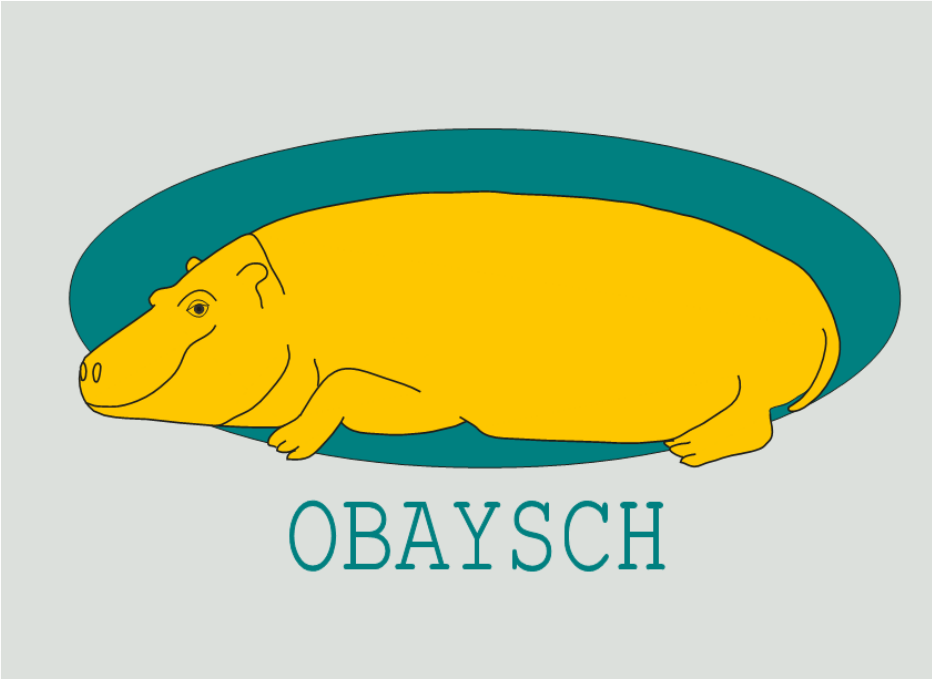
Christian Kaul

https://www.linkedin.com/in/christian-kaul/

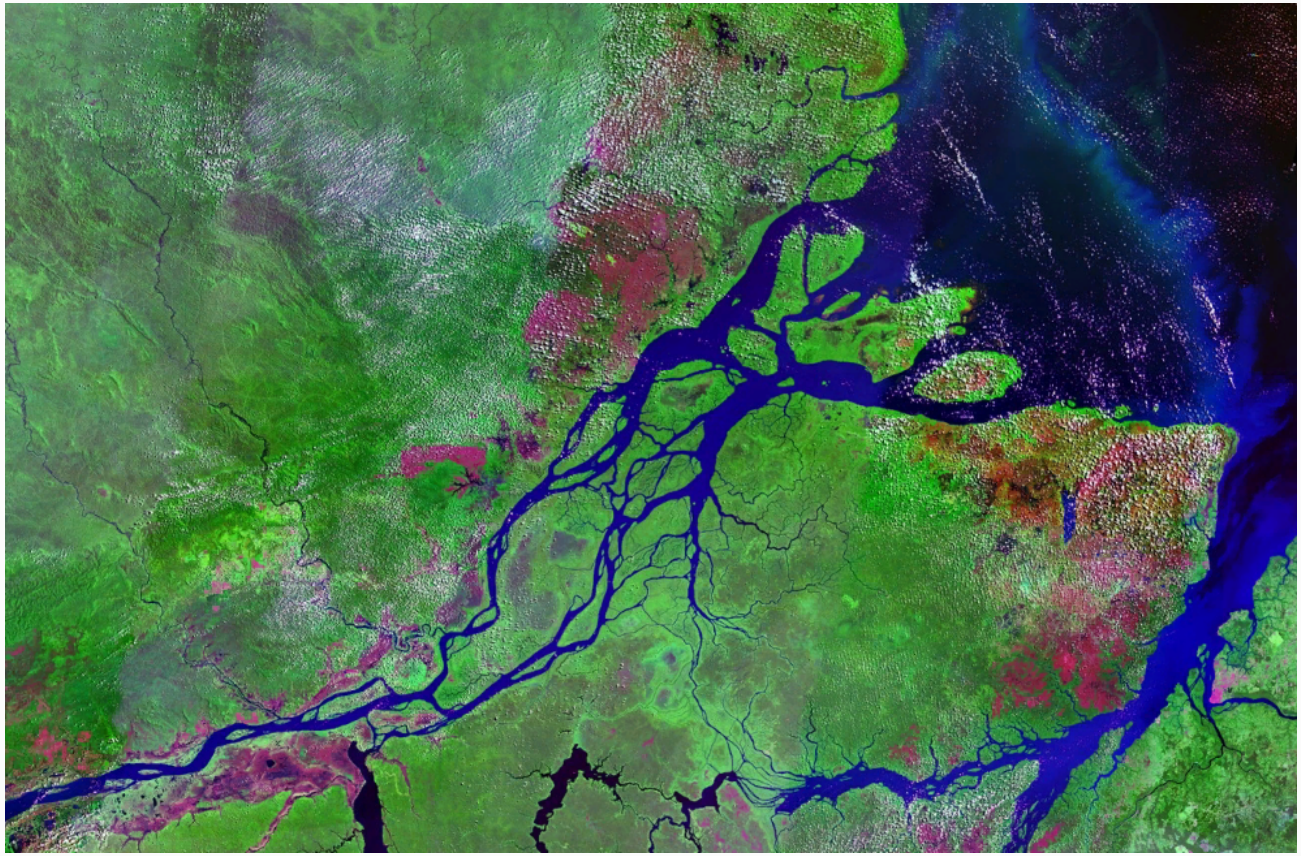
Data Modeling Aficionado & Creator at Obaysch

Munich, Germany

Ask me about the Knowledge Gap Spring 2020 data modeling conference!



What Is This All About?



Source: https://commons.wikimedia.org/wiki/File:Mouths_of_amazon_geocover_1990.png

And Where Do We Go From Here?

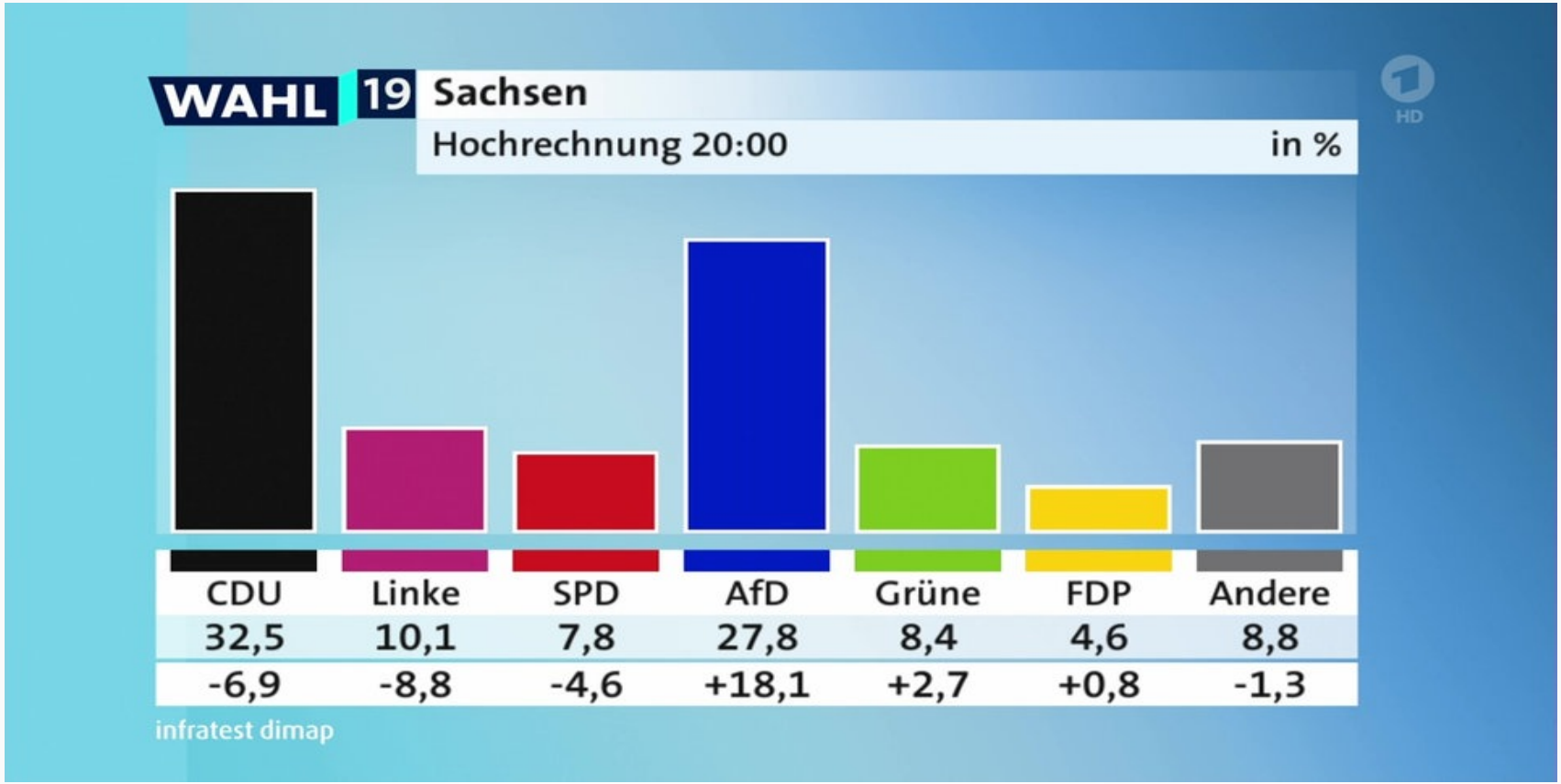
- The Current State of Transitional Modeling
  - Posits & Assertions
  - Schema by Design
  - Bareclad
- The Transitional Organization (TransOrg)
  - Conceptual Model
  - Logical Design
  - Physical Implementations

The Current State of Transitional Modeling

Elevator Pitch: Transitional Modeling

- Invented by Lars Rönnbäck (of Anchor Modeling fame).
- Can deal with conflicting, unreliable, varying information.
- Has a very simple notation.
- Data Vault, Anchor, 3NF can be considered special cases.

And Now … It‘s Election Day!



Source: https://www.tagesschau.de/inland/sachsen-landtagswahl-111.html

Appearances

- An **appearance** is a pair that consists of
  - a **unique identifier** for something and
  - a **role** that describes the context in which this thing appears.

- (**1990**, *name*)
- (**1990**, *party*)
- (**2019**, *level*)
- (**2019**, *election*)
- (**1975**, *name*)
- (**1975**, *candidate*)
- (**100**, *result*)

Dereferencing Sets

- A **dereferencing set** is a set of one or more appearances that belong together.
- Sets with only one appearance stand for attributes of a thing.
- Sets with more than one appearance stand for relationships between things.

- {(**1990**, *name*)}
- {(**2019**, *level*)}
- {(**1975**, *name*)}
- {(**1990**, *party*), (**1975**, *candidate*), (**2019**, *election*)}
- {(**1990**, *party*), (**2019**, *election*), (**100**, *result*)}

Posits

- A **posit** is a triple that consists of
  - a dereferencing set,
  - some **value** and
  - a time point, the **appearance time** of this dereferencing set with this value.

- [{(**1990**, *name*)}, **CDU Landesverband Sachsen**, 1990-03-03]
- [{(**1975**, *name*)}, **Michael Kretschmer**, 1975-05-07]
- [{(**1990**, *party*), (**2019**, *election*), (**1975**, *candidate*)}, **nominated**, 2019-01-07]

Assertions

- An **assertion** is a meta-posit that includes
  - the unique identifier of the posit being asserted,
  - the unique identifier of the **asserter**,
  - a **confidence value** in the range [−1, 1] and
  - the **assertion time**.

- [{(**1950**, *name*)}, **ARD**, 1950-06-09]
- **P325:** [{(**100**, *share of votes*)}, **32.5**, 2019-09-01 18:00]
- **P321:** [{(**100**, *share of votes*)}, **32.1**, 2019-09-01 18:00]
- [{(**P325**, *posit*), (**1950**, *asserter*)}, **0.9**, 2019-09-01 20:00]
- [{(**P325**, *posit*), (**1950**, *asserter*)}, **0**, 2019-09-02 00:30]
- [{(**P321**, *posit*), (**1950**, *asserter*)}, **1**, 2019-09-02 00:30]

Elevator Pitch: Schema by Design

- Only impose theoretical minimum of structure at write time.
- Provide enough metainformation to create models.
- Choose appropriate model at read time.

Classes

- A **class** is a thing that can be used to classify other things.
- At read time, the classes will become the entities of the model the consumer uses.

- [{(**1**, *name*)}, **State-Level Politician**, 2019-10-24]
- [{(**2**, *name*)}, **State-Level Party Organization**, 2019-10-24]
- [{(**3**, *name*)}, **State-Level Election**, 2019-10-24]
- [{(**8**, *name*)}, **Political Actor**, 2019-10-24]
- [{(**9**, *name*)}, **Political Event**, 2019-10-24]

Classifiers

- A **classifier** is a special kind of posit that assigns a thing to a certain class.
- At read time, the classifiers will be used to populate the entities the consumer uses.

- **P751:** [{(**1975**, *has*), (**1**, *class*)}, **classified**, 2019-10-24]
- **P902:** [{(**1990**, *has*), (**2**, *class*)}, **classified**, 2019-10-24]
- **P193:** [{(**2019**, *has*), (**3**, *class*)}, **classified**, 2019-10-24]
- **P758:** [{(**1975**, *has*), (**8**, *class*)}, **classified**, 2019-10-24]
- **P908:** [{(**1990**, *has*), (**8**, *class*)}, **classified**, 2019-10-24]
- **P199:** [{(**2019**, *has*), (**9**, *class*)}, **classified**, 2019-10-24]

Assertions of Classifiers

- An **assertion of a classifier** is an assertion whose posit is a classifier.
- These assertions represent the modeling decisions of different modelers.

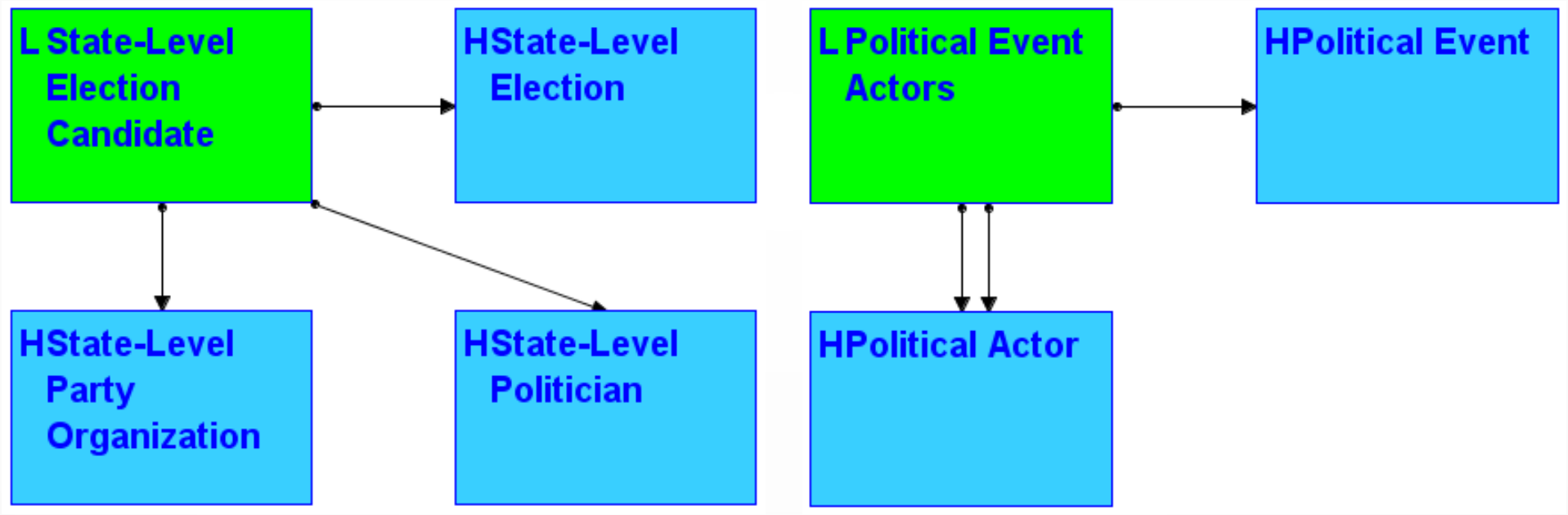
- [{(**42**, *name*)}, **Hans Hultgren**, A long time ago]
- [{(**P751**, *posit*), (**42**, *asserter*)}, **1**, 2019-10-24]
- [{(**P902**, *posit*), (**42**, *asserter*)}, **1**, 2019-10-24]
- [{(**P193**, *posit*), (**42**, *asserter*)}, **1**, 2019-10-24]
- [{(**47**, *name*)}, **David C. Hay**, A long time ago]
- [{(**P758**, *posit*), (**47**, *asserter*)}, **1**, 2019-10-24]
- [{(**P908**, *posit*), (**47**, *asserter*)}, **1**, 2019-10-24]
- [{(**P199**, *posit*), (**47**, *asserter*)}, **1**, 2019-10-24]

Models

- A **model** is a body of information in which each thing has been classified using assertions of classifiers.
- Different models show the same posits in a different structure.

- [{(**M42**, *name*)}, **Hans Hultgren's Election Model**, 2019-10-24]
- [{(**M47**, *name*)}, **David C. Hay's Election Model**, 2019-10-24]
- [{(**1990**, *party*), (**2019**, *election*), (**1975**, *candidate*)}, **nominated**, 2019-01-07]

Same Data, Different Models



And How Do I Store All This?

- Lars Rönnbäck is working on **Bareclad**, an open-source database based on transitional modeling and implemented in Rust.
- If you want to contribute: https://github.com/Roenbaeck/bareclad.

{ bareclad™ }

The Transitional Organization (TransOrg)

Organizational Pain Points

- Disconnect between business model, IT systems (both operative and analytic) and organizational structure.
- Data distributed over a high number of non-integrated IT systems.
- Frequent data migrations necessary (e. g. because of vendor-imposed, application-specific data models).
- Often manual interfaces between incompatible applications.
- GDPR compliance almost impossible (personal data is all over the place, you are probably not even aware of some of the locations).

Organizational Wish List

- Close alignment between business model, IT systems and organizational structure.
- Manageable number of IT systems integrated by a common model.
- No technically-driven data migrations.
- No incompatible applications that require manual interfaces.
- Easy GDPR compliance (each piece of personal data can be found, corrected and deleted in one dedicated location).

What If Your Whole Organization Was Generated From a Model?

Then You Would Have A Transitional Organization.

Levels of TransOrg

- **Conceptual Model:**
  - Narrative lines (temporal, possibly conflicting stories people tell each other)
  - Uses transitional modeling
- **Logical Design:**
  - Concepts, connections, details (derived from narrative lines)
  - Uses 5NF or 6NF pattern
- **Physical Implementations:**
  - Organizational chart (small teams, 1:1 with concepts)
  - Physical data stores (derived from logical design using shape functions)

Conceptual Model

- Consists of **narrative lines** (temporal, possibly conflicting stories people tell each other about how things happen in the organization).
- Includes a **common meta-model** (agreed-upon classifications for things that are important for the organization).
- Uses transitional modeling (posits and assertions).

Logical Design

- Consists of **concepts** (the important things), **connections** (the relationships between them) and **details** (their attributes).
- Derived from narrative lines using the common meta-model.
- Uses a 5NF (Vault-style) or a 6NF (Anchor-style) pattern.

Physical Implementation: Org Chart

- Organizational chart derived from logical design.
- Small cross-functional teams, 1:1 with concepts.
- Each team responsible for the details of their concept.
- Defined interfaces between teams, 1:1 with connections.
- Teams jointly responsible for their common connections, usually with one team in the lead.

Physical Implementation: Data Store

- Physical data stores derived from logical design using **shape functions** (mathematical functions that can represent shapes in the logical design as well as in the physical implementation).
- Each team responsible for storing data about instances of their concept, its details and the connections for which they are in the lead in one and only one defined location.
- All data stores interoperable because of the common logical design.
- Applications created on top of these data stores, no application-specific data models or data stores.

Just Model Your Reality

- Regularly document representative examples of what happens in your organization using narrative lines.
- If you want to make structural changes to the organization, build consensus to alter the transitional (meta-)model accordingly.
- Generate the new logical design of your organization from the altered conceptual model.
- Generate the new org chart and changes in the data stores from the altered logical design.

What About the Wish List?

- Close alignment between business model, IT systems and organizational structure. ✓
- Manageable number of IT systems integrated by a common model. ✓
- No technically-driven data migrations. ✓
- No incompatible applications that require manual interfaces. ✓
- Easy GDPR compliance (each piece of personal data can be found, corrected and deleted in one dedicated location). ✓

Still Interested in This Stuff?

- Lars Rönnbäck‘s presentations and articles:
  www.anchormodeling.com/publications/
- My articles (on Transitional Modeling, TransOrg and other topics):
  www.linkedin.com/in/christian-kaul/detail/recent-activity/posts/
- A related, similarly grandiose initiative from Tim Berners-Lee:
  https://solid.mit.edu
- Dave McComb‘s Data-Centric Manifesto:
  http://datacentricmanifesto.org