data modeling zone

Becoming a Model-Driven Organization

in Four Simple Steps

Who Is This Guy?

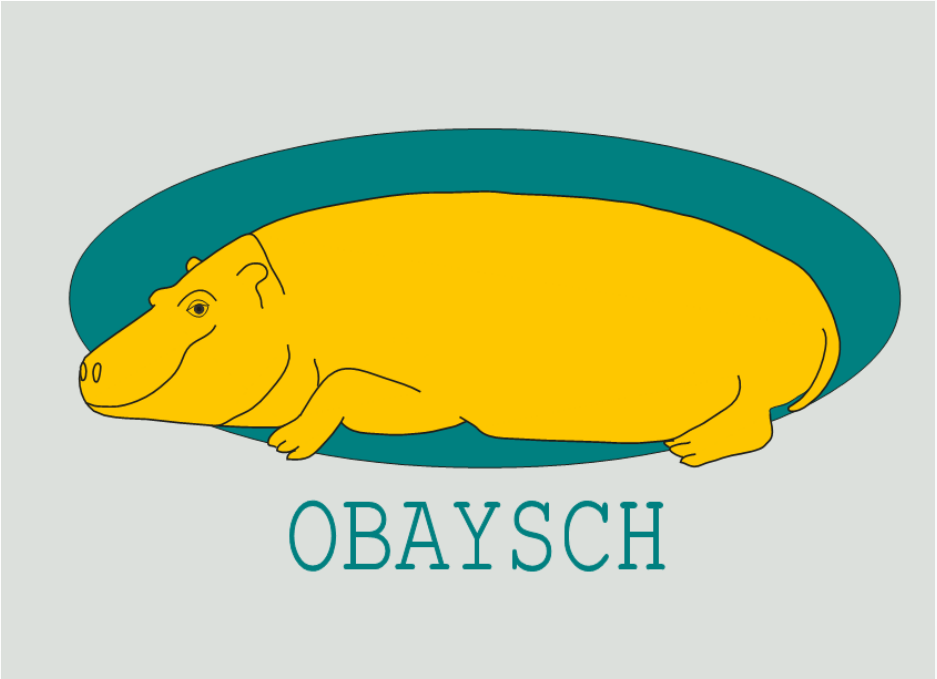
Christian Kaul

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

Data Modeling Aficionado & Creator at Obaysch

Munich, Germany

Ask me about the Knowledge Gap 2022 data modeling & data architecture conference!



What Is This All About?

You never change things by fighting the existing reality. To change something, build a new model that makes the existing model obsolete.

– R. Buckminster Fuller, 1983

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).

You Want Your Wishes to Become True?

Become a Model-Driven Organization ...

. . . in Four Simple Steps

1. Interesting Instances
2. Common Classifications
3. Logical Design
4. Physical Implementation

Model-Driven Organization Levels

- **Conceptual:**
  - Interesting instances
  - Common classifications
  - Uses e. g. transitional modeling
- **Logical:**
  - Concepts, connections, details
  - Uses 6NF pattern (like anchor modeling)
- **Physical:**
  - Organizational chart
  - Physical data stores

Step 1: Interesting Instances

Interesting Instances

- Temporal, possibly conflicting stories people tell each other about how things happen in the organization.
- Think concrete examples of business processes.
- Uses e. g. **transitional modeling** (posits and assertions).

Wait a Minute …
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.

Example: Flight

Julian Herzog, CC BY 4.0, https://commons.wikimedia.org/wiki/File:Airbus_A350-941_F-WWCF_MSN002_ILA_Berlin_2016_17.jpg

Posits

- A **posit** is a triple of
  - a **dereferencing set** including one or more **appearances** consisting of a **unique identifier** and a **role**,
  - some **value** and
  - a time point, the **appearance time** of this dereferencing set with this value.

- [{(**1953**, *name*)}, **Deutsche Lufthansa**, 1954-09-17]
- [{(**1986**, *name*)}, **Christian Kaul**, 1986-01-26]
- [{(**1986**, *customer*), (**1953**, *vendor*), (**2021**, *ticket*)}, **bought**, 2021-10-27]

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**.

- **P23**: [{(**2021**, *departure gate*)}, **G23**, 2021-11-19 11:30]
- **P42**: [{(**2021**, *departure gate*)}, **G42**, 2021-11-19 11:30]
- [{(**P23**, *posit*), (**1953**, *asserter*)}, **0.8**, 2021-11-19 07:30]
- [{(**P23**, *posit*), (**1953**, *asserter*)}, **−1**, 2021-11-19 11:15]
- [{(**P42**, *posit*), (**1953**, *asserter*)}, **1**, 2021-11-19 11:15]

Step 2:
Common Classifications

Common Classifications

- Agreed-upon classifications for things that are important for the organization.
- Think different buckets for different types of instances.
- Uses a **schema-by-design** approach.

Wait a Minute …
Schema by Design?

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*)}, **Passenger**, 2021-11-18]
- [{(**2**, *name*)}, **Airline**, 2021-11-18]
- [{(**3**, *name*)}, **Plane Ticket**, 2021-11-18]
- [{(**8**, *name*)}, **Party**, 2021-11-18]
- [{(**9**, *name*)}, **Agreement**, 2021-11-18]

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.

- **P186**: [{(**1986**, *has*), (**1**, *class*)}, **classified**, 2021-11-18]
- **P253**: [{(**1953**, *has*), (**2**, *class*)}, **classified**, 2021-11-18]
- **P319**: [{(**2019**, *has*), (**3**, *class*)}, **classified**, 2021-11-18]
- **P886**: [{(**1986**, *has*), (**8**, *class*)}, **classified**, 2021-11-18]
- **P853**: [{(**1953**, *has*), (**8**, *class*)}, **classified**, 2021-11-18]
- **P919**: [{(**2019**, *has*), (**9**, *class*)}, **classified**, 2021-11-18]

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]
- [{(**P186**, *posit*), (**42**, *asserter*)}, **1**, 2021-11-18]
- [{(**P253**, *posit*), (**42**, *asserter*)}, **1**, 2021-11-18]
- [{(**P319**, *posit*), (**42**, *asserter*)}, **1**, 2021-11-18]

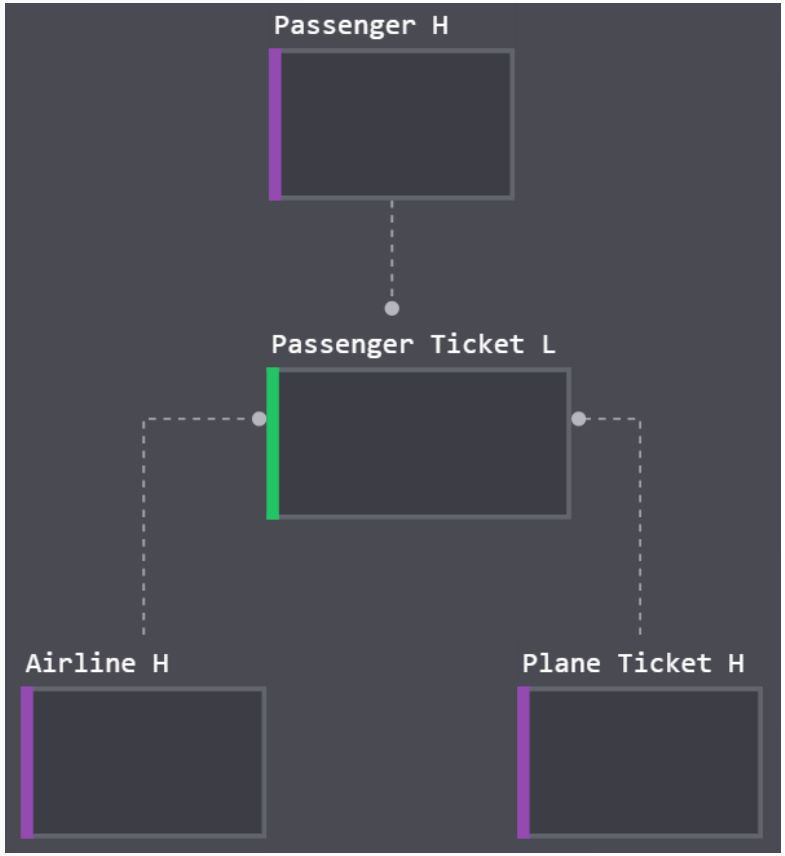
- [{(**47**, *name*)}, **David C. Hay**, A long time ago]
- [{(**P886**, *posit*), (**47**, *asserter*)}, **1**, 2021-11-18]
- [{(**P853**, *posit*), (**47**, *asserter*)}, **1**, 2021-11-18]
- [{(**P919**, *posit*), (**47**, *asserter*)}, **1**, 2021-11-18]

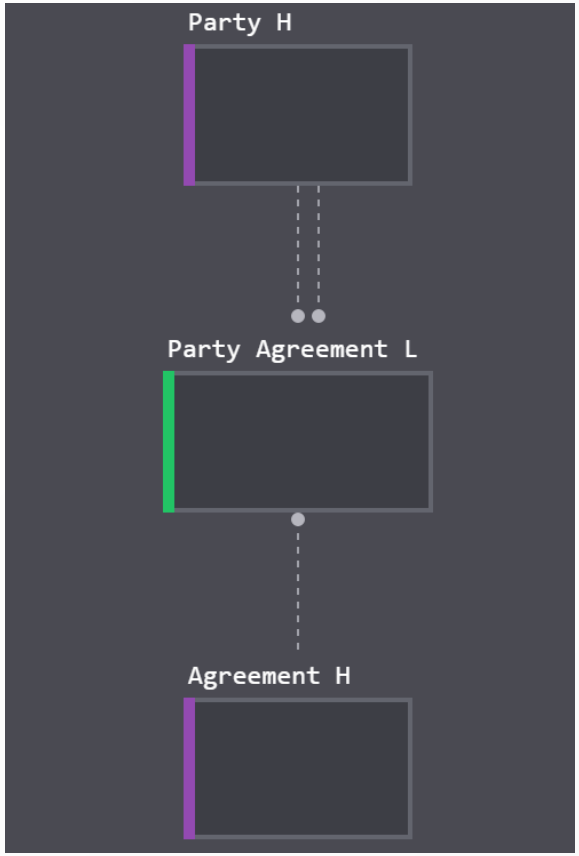
Models

- 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 Flight Model**, 2021-11-18]
- [{(**M47**, *name*)}, **David C. Hay's Flight Model**, 2021-11-18]
- [{(**1986**, *customer*), (**1953**, *vendor*), (**2021**, *ticket*)}, **bought**, 2021-11-18]

Same Data, Different Models





Step 3:
Logical Design

Logical Design

- Consists of **concepts** (the important things), **connections** (the relationships between them) and **details** (their attributes).
- Derived from interesting instances using common classifications.
- Uses a 6NF (Anchor-style) pattern.

Step 4:
Physical Implementation

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 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.

Putting It All Together

Just Model Your Reality

- Regularly document **interesting instances**, representative examples of what happens in your organization.
- If you want to make structural changes to the organization, build consensus to alter the relevant **common classifications**.
- 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). ✓

Want to Dig Deeper? Then Read This …

- Lars Rönnbäck‘s presentations and articles
- My articles (on transitional modeling and many other topics)
- Our common article on the model-driven organization
- One-day training (also available in German)

… Or Talk to Me!

Christian Kaul

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

Data Modeling Aficionado & Creator at Obaysch

Munich, Germany

Ask me about the Knowledge Gap Spring 2022 data modeling & data architecture conference!