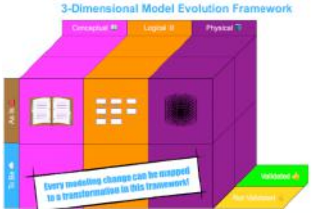
Mastering Data Modeling:

Understanding Conceptual, Logical, and Physical Models


3-Dimensional Model Evolution Framework
Conceptual
Logical
Physical
Every modeling change can be mapped to a transformation in this framework!
Validated
Not Validated

peter.aiken@anythingawesome.com +1.804.382.5957

Peter Aiken, Ph.D.

- I've been doing this a long time
- My work is recognized as useful
- Associate Professor of IS (vcu.edu)
- Institute for Defense Analyses (ida.org)
- DAMA International (dama.org)
- MIT CDO Society (iscdo.org)
- Anything Awesome (anythingawesome.com)
- Experienced w/ 500+ data management practices worldwide
- 13 books and dozens of articles
- Multi-year immersions
  - US DoD (DISA/Army/Marines/DLA)
  - Nokia
  - Deutsche Bank
  - Wells Fargo
  - Walmart
  - HUD …

$1,500,000,000.00 USD

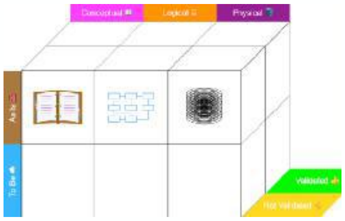
BEYOND BYTE$
DATA LITERACY
THE CDO JOURNEY
MONETIZING DATA MANAGEMENT
The Case for the Chief Data Officer
Peter Aiken and Michael Gorman
Data Reverse Engineering
Peter H. Aiken
Building Corporate Portals with XML
XML IN DATA MANAGEMENT

https://anythingawesome.com

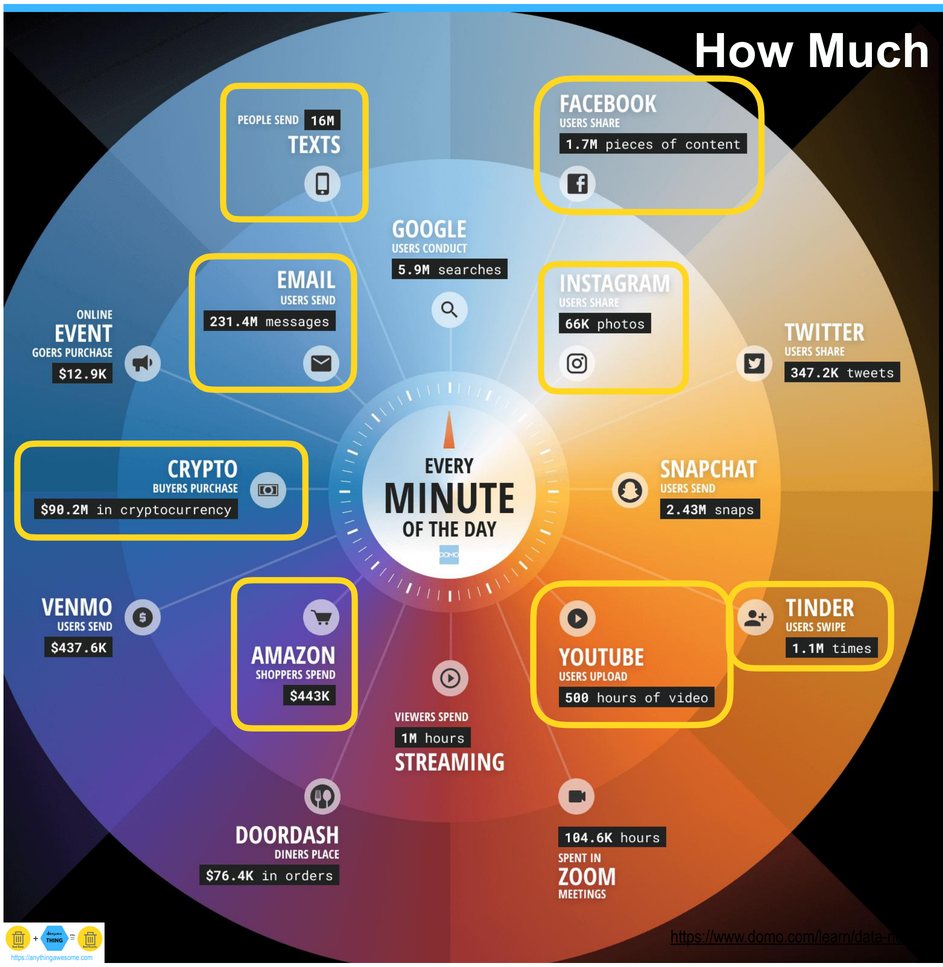
Program Overview

Conceptual
Versus
Logical
Versus
Physical
Data Modeling

- Introduction to Modeling Data
  - Motivation
  - 3 primary data model types (+ plus two characteristics)
  - Reasons for each
  - Purposeful Modeling Basics (conversions, forward/reverse engineering)
- Conceptual
  - Motivation: Architectural tradeoffs
  - Strategy and conceptual data modeling
  - Glossary/Dictionary capabilities
- Logical
  - Motivation: Simplicity (Operational and Design)
  - Motivation towards standards
  - Business meets strategy
- Physical
  - Motivation: Required documentation and/or facts
  - Become the blueprints for physical construction of the solution
  - Blueprints are used for future maintenance of the solution
- Take Aways/References/Q&A



How Much Data (by the Minute?)

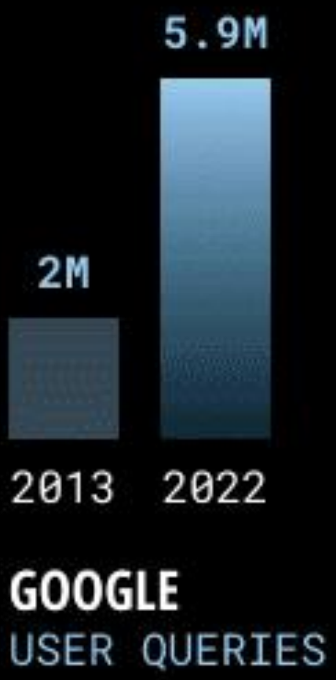

PEOPLE SEND 16M TEXTS
FACEBOOK USERS SHARE 1.7M pieces of content
GOOGLE USERS CONDUCT 5.9M searches
EMAIL USERS SEND 231.4M messages
INSTAGRAM USERS SHARE 66K photos
ONLINE EVENT GOERS PURCHASE $12.9K
TWITTER USERS SHARE 347.2K tweets
CRYPTO BUYERS PURCHASE $90.2M in cryptocurrency
EVERY MINUTE OF THE DAY
DOMO
SNAPCHAT USERS SEND 2.43M snaps
VENMO USERS SEND $437.6K
AMAZON SHOPPERS SPEND $443K
YOUTUBE USERS UPLOAD 500 hours of video
TINDER USERS SWIPE 1.1M times
VIEWERS SPEND 1M hours STREAMING
DOORDASH DINERS PLACE $76.4K in orders
104.6K hours SPENT IN ZOOM MEETINGS

For the entirety of 2022, every minute of every day:

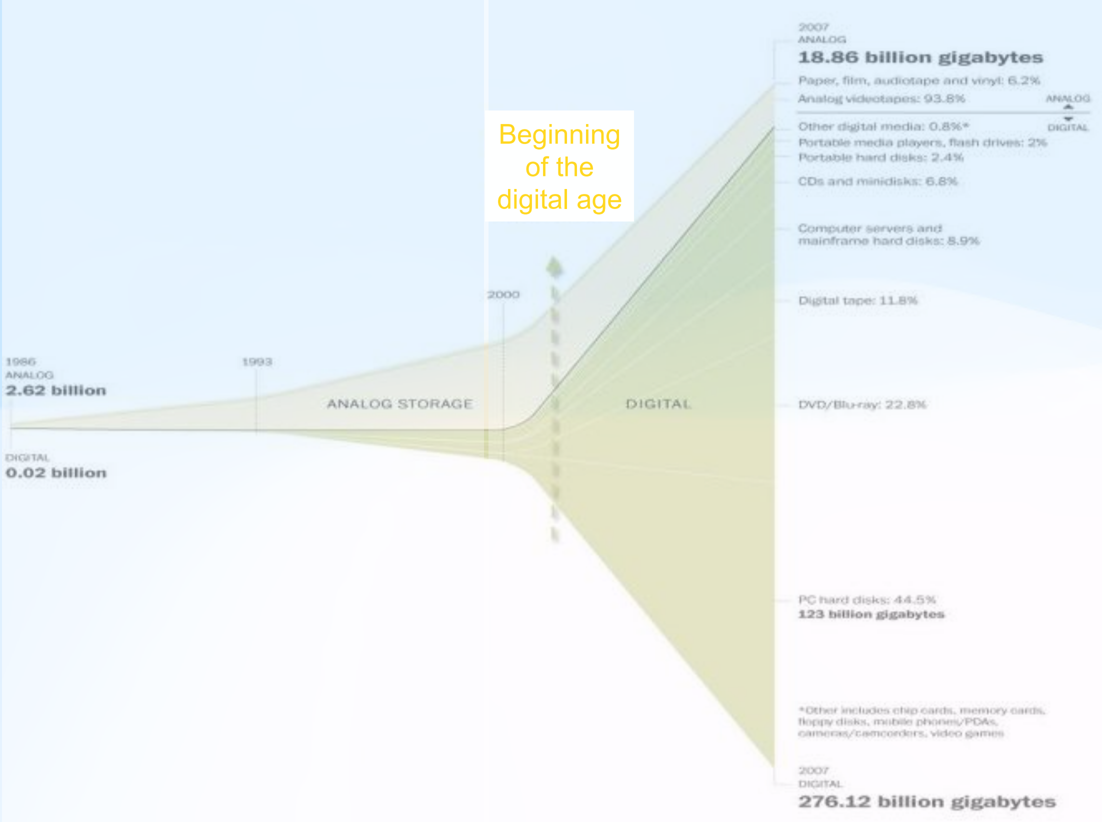
- Facebook users share 1.7 million pieces of content
- Instagram users share 66K photos
- Tinder users record 1M swipes
- YouTube users upload 500 hours of video
- Amazon shoppers spend $443K
- Crypto buyers purchase $90M+
- Email users send 231M messages
- People send 16M texts

https://www.domo.com/learn/data-n

Data Never Sleeps 1 Compared to 10

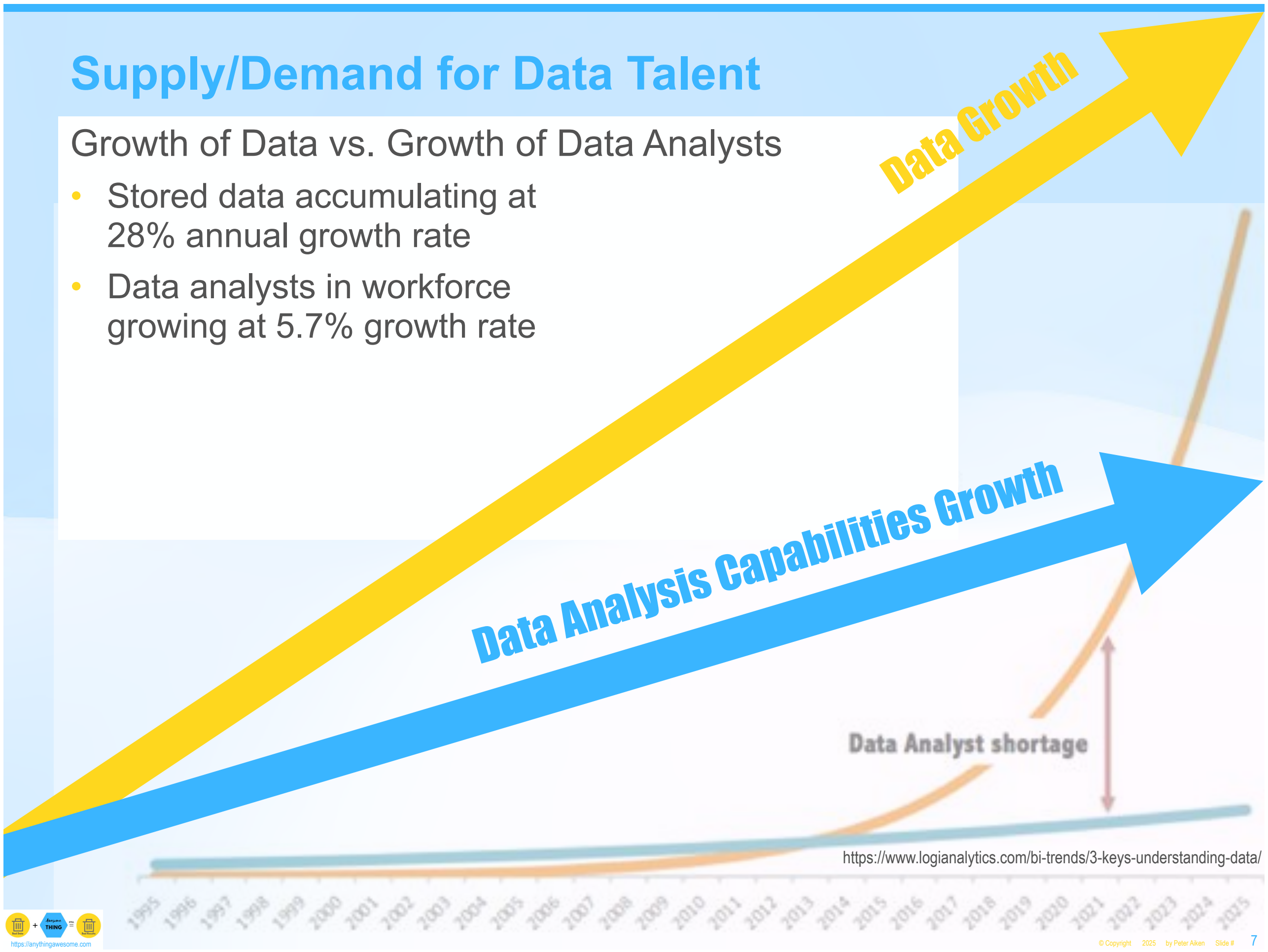


Global Information Storage Capacity



Beginning of the digital age

https://www.martinhilbert.net/worldinfocapacity-html/

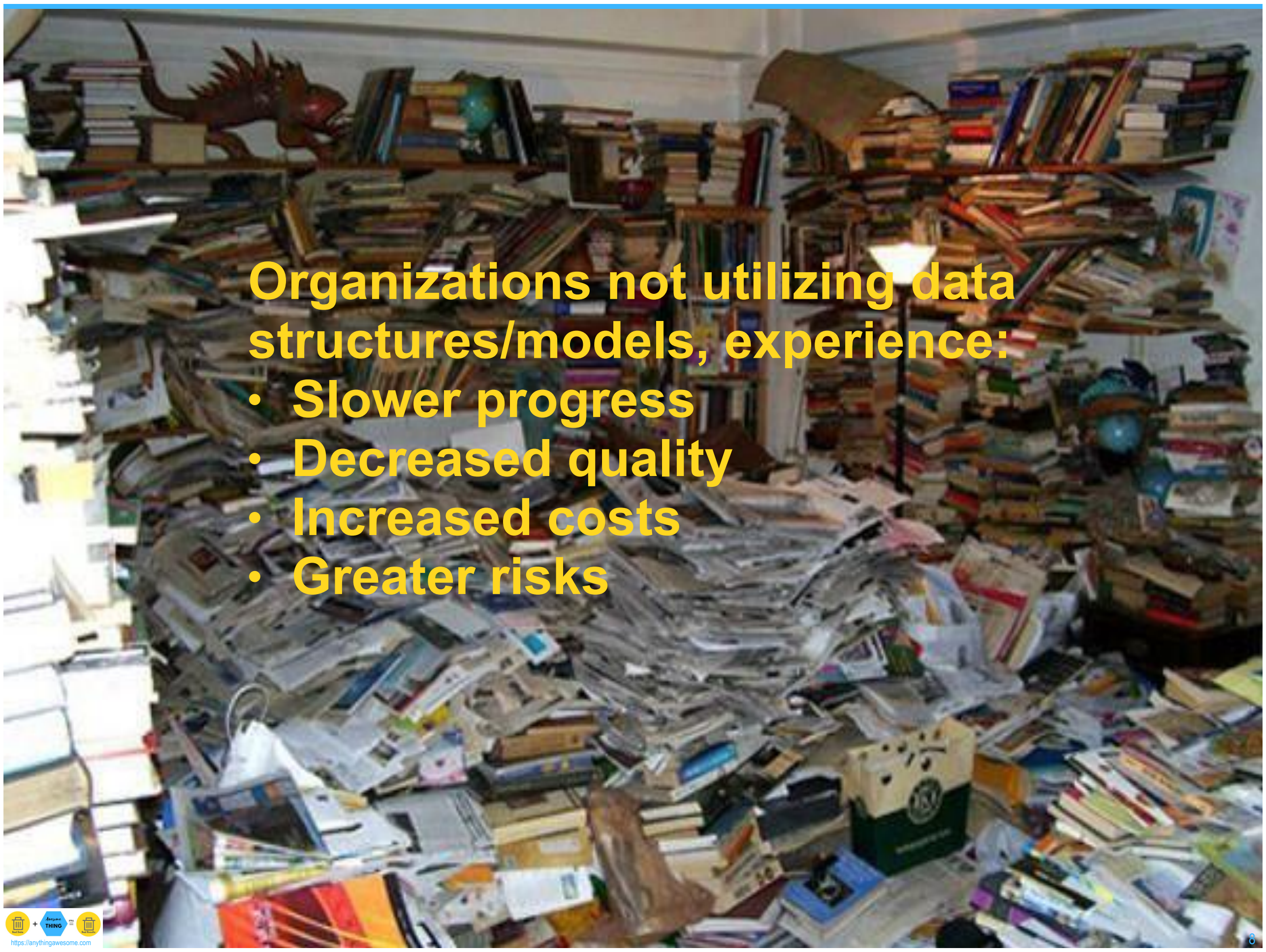


Supply/Demand for Data Talent

Growth of Data vs. Growth of Data Analysts

- Stored data accumulating at 28% annual growth rate
- Data analysts in workforce growing at 5.7% growth rate

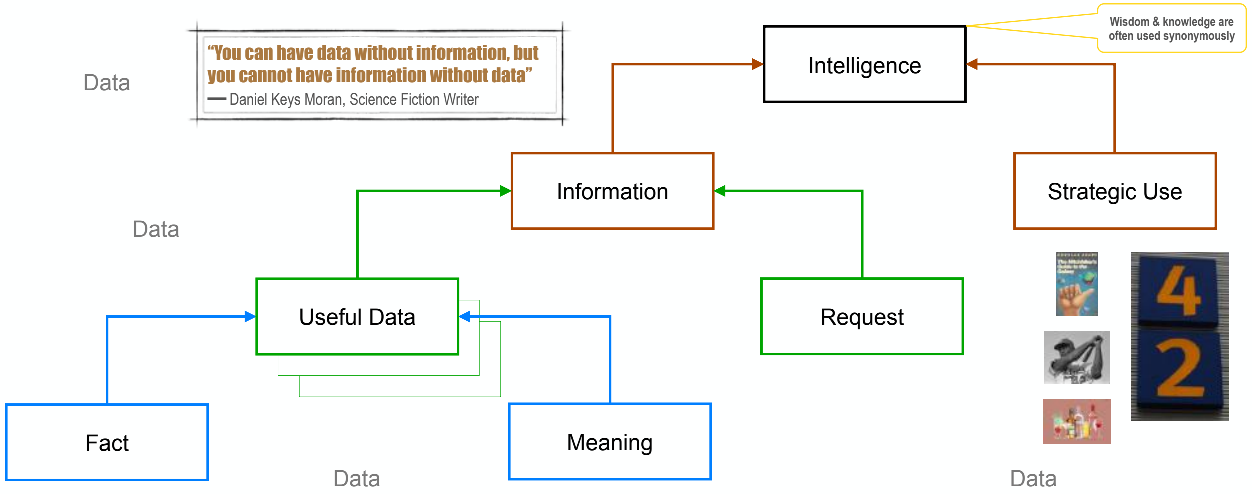
https://www.logianalytics.com/bi-trends/3-keys-understanding-data/



Organizations not utilizing data structures/models, experience:

- Slower progress
- Decreased quality
- Increased costs
- Greater risks

A Model Precisely Defining 3 Important Concepts


Data
"You can have data without information, but you cannot have information without data"
— Daniel Keys Moran, Science Fiction Writer
Intelligence
Wisdom & knowledge are often used synonymously
Data
Information
Strategic Use
Useful Data
Request
Fact
Meaning
Data
Data

1. Each FACT combines with one or more MEANINGS.
2. Each specific FACT and MEANING combination is referred to as a DATUM.
3. An INFORMATION is one or more DATA that are returned in response to a specific REQUEST
4. INFORMATION REUSE is enabled when one FACT is combined with more than one MEANING.
5. INTELLIGENCE is INFORMATION associated with its STRATEGIC USES.
6. DATA/INFORMATION must formally arranged into an ARCHITECTURE.

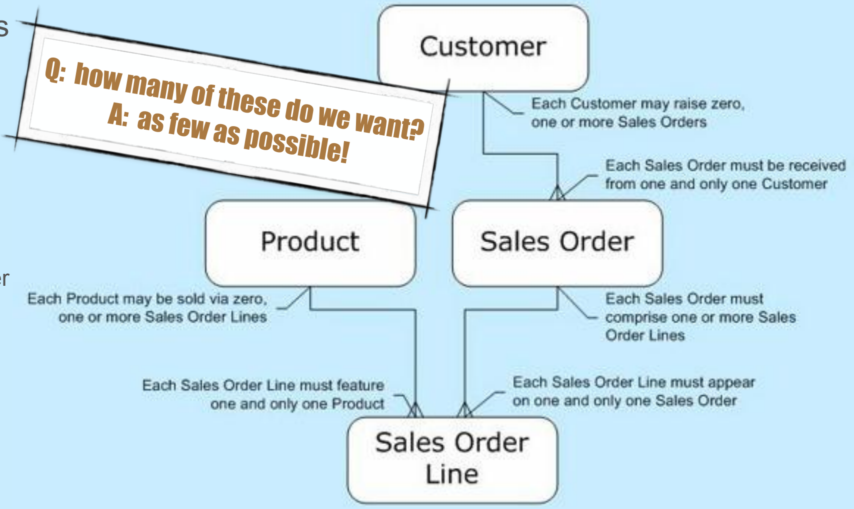
[Built on definitions from Dan Appleton. 1983]

Each Data Arrangement Is a Data Structure

"An organization of information, usually in memory, for better algorithm efficiency, such as queue, stack, linked list, heap, dictionary, and tree, or conceptual unity, such as the name and address of a person. It may include redundant information, such as length of the list or number of nodes in a subtree."

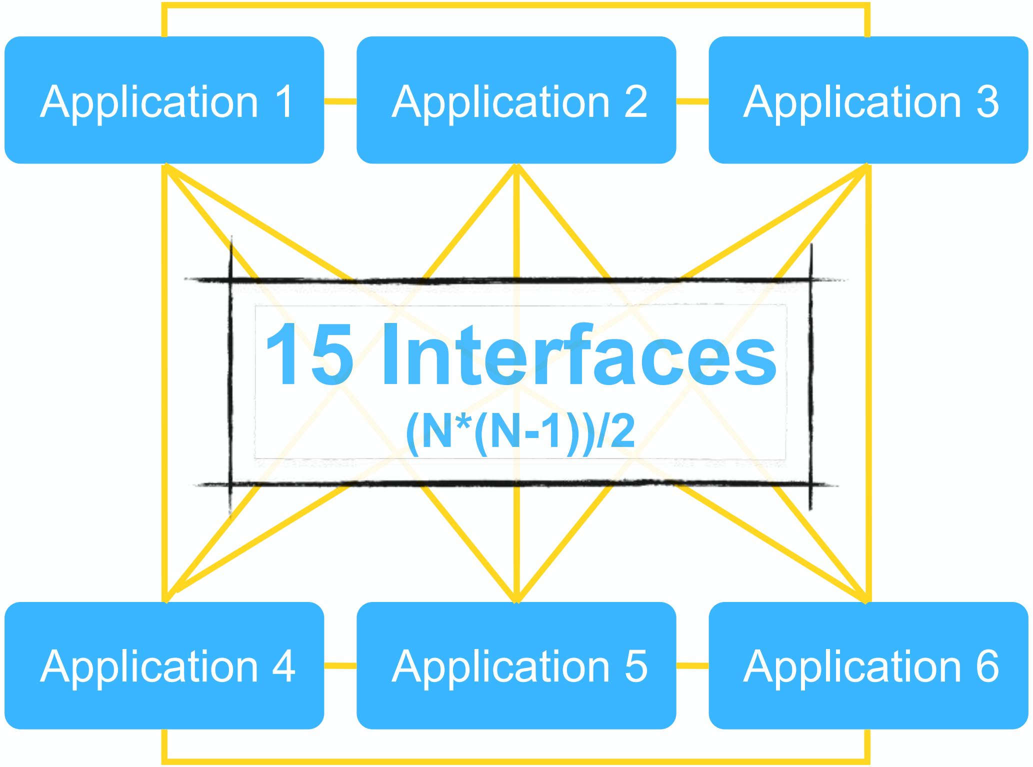
Some data structure characteristics

- Grammar for data objects
  - Grammar is the principles or rules of an art, science, or technique "a grammar of the theater"
- Data Object Constraints
- Ordering
  - Sequential, hierarchical, relational, network, lake, other
- Uniqueness
- Balance
- Optimality
- Future enhanceability
  - Multi-currency
  - Device handoff features


Q: how many of these do we want?
A: as few as possible!
Customer
Each Customer may raise zero, one or more Sales Orders
Each Sales Order must be received from one and only one Customer
Product
Sales Order
Each Product may be sold via zero, one or more Sales Order Lines
Each Sales Order must comprise one or more Sales Order Lines
Each Sales Order Line must feature one and only one Product
Each Sales Order Line must appear on one and only one Sales Order
Sales Order Line

http://www.nist.gov/dads/HTML/datastructur.html

How Many Interfaces Are Required To Solve This Integration Problem?



Application 1

Application 2

Application 3

15 Interfaces

(N*(N-1))/2

Application 4

Application 5

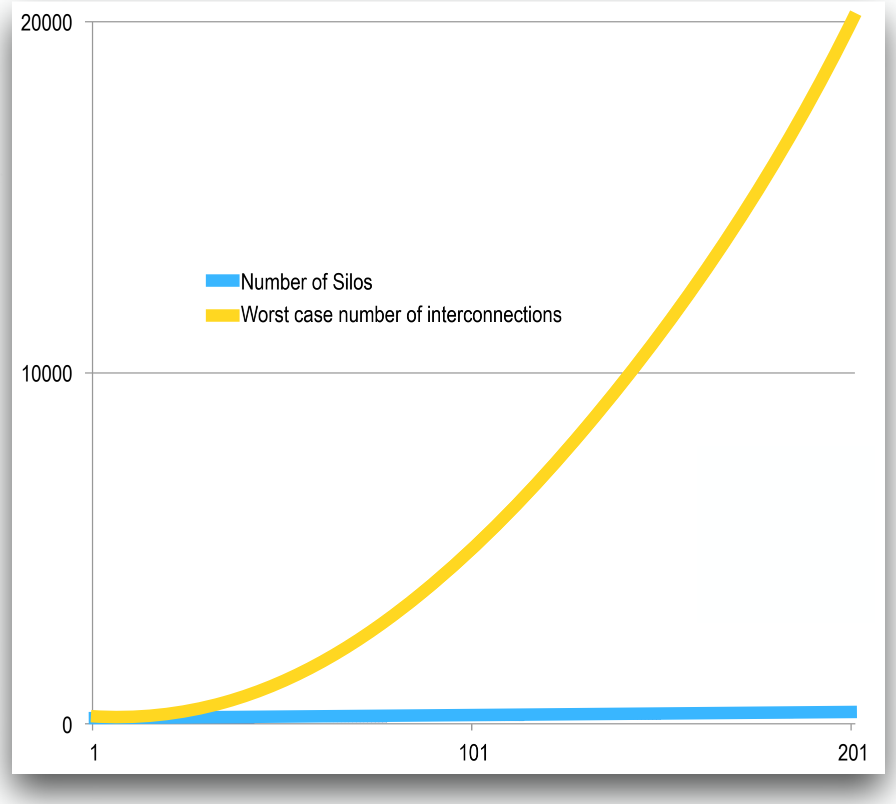
Application 6

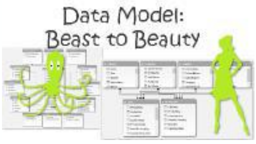
RBC: 200 applications - 4900 batch interfaces

The Rapidly Increasing Cost of Complexity

N

- 6 / 15
- 60 / 1,770
- 600 / 179,700
- 200 / 19,900
- 200 / 5,000 (actual)



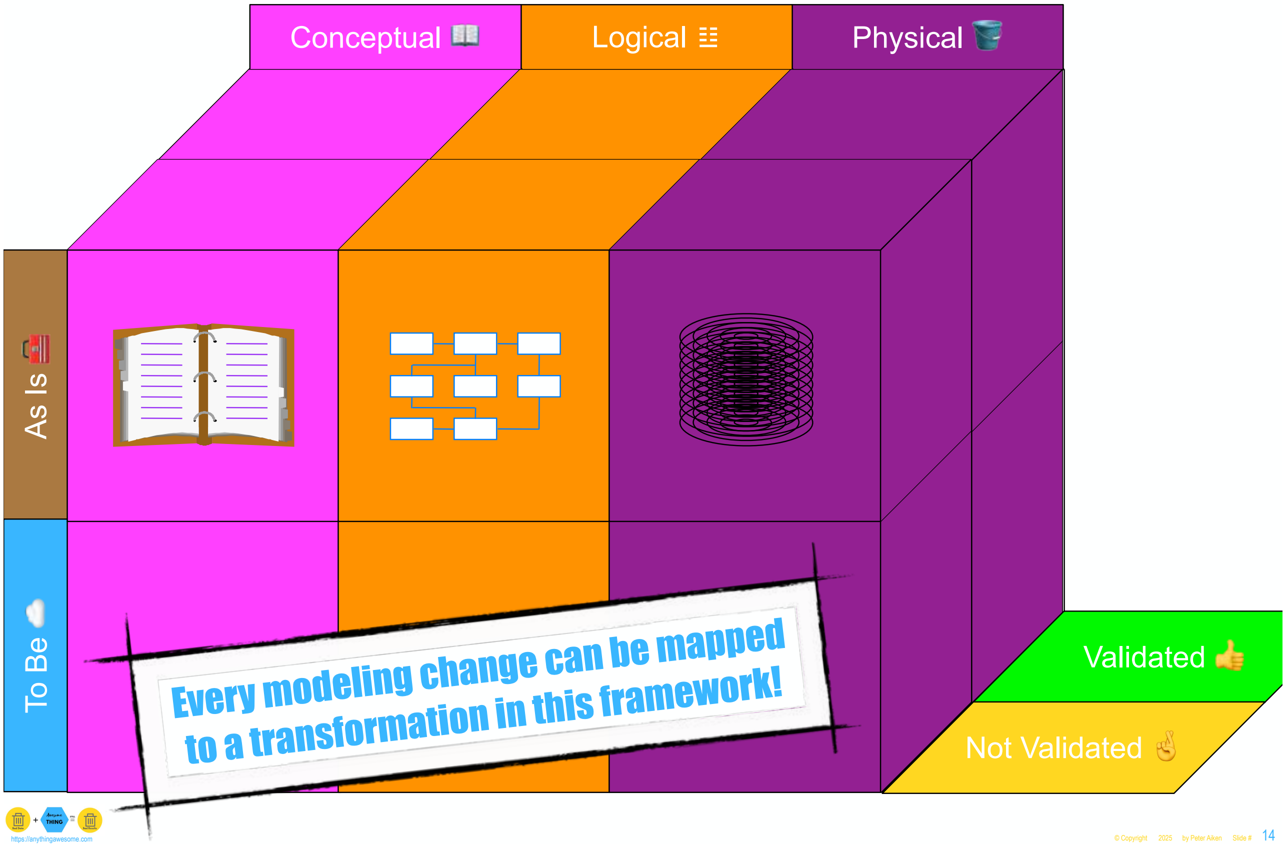


powerpivotpro.com

Why Model?

• Would you build a house without an architecture sketch?	• Model is the sketch of the system to be built in a project.
• Would you like to have an estimate how much your new house is going to cost?	• Your model gives you a very good idea of how demanding the implementation work is going to be!
• If you hired a set of constructors from all over the world to build your house, would you like them to have a common language?	• Model is the common language for the project team.
• Would you like to verify the proposals of the construction team before the work gets started?	• Models can be reviewed before thousands of hours of implementation work will be done.
• If it was a great house, would you like to build something rather similar again, in another place?	• It is possible to implement the system to various platforms using the same model.
• Would you drill into a wall of your house without a map of the plumbing and electric lines?	• Models document the system built in a project. This makes life easier for the support and maintenance!

3-Dimensional Model Evolution Framework



Conceptual 📖

Logical ☷

Physical 🗑

As Is 🧰

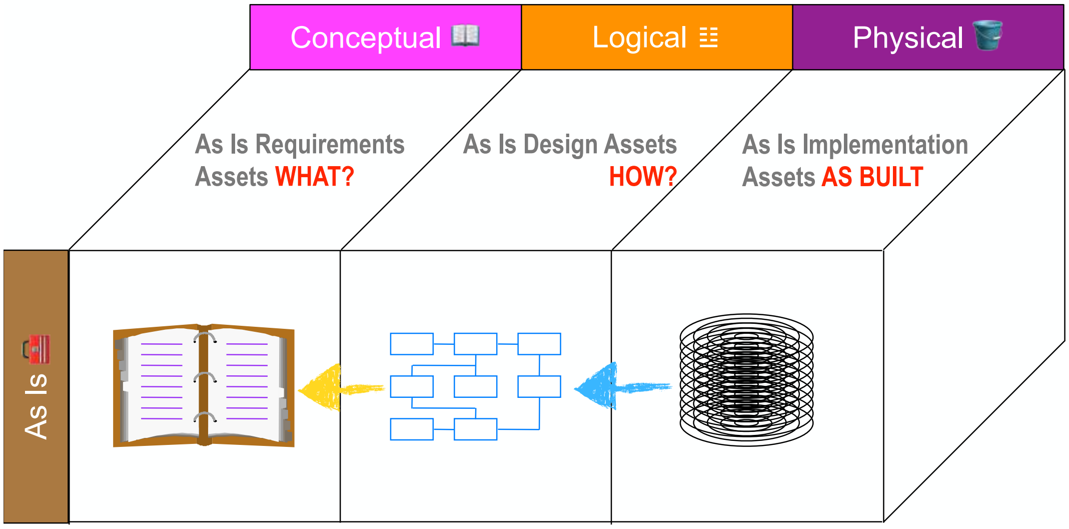
To Be ☁

Validated 👍

Not Validated 🤞

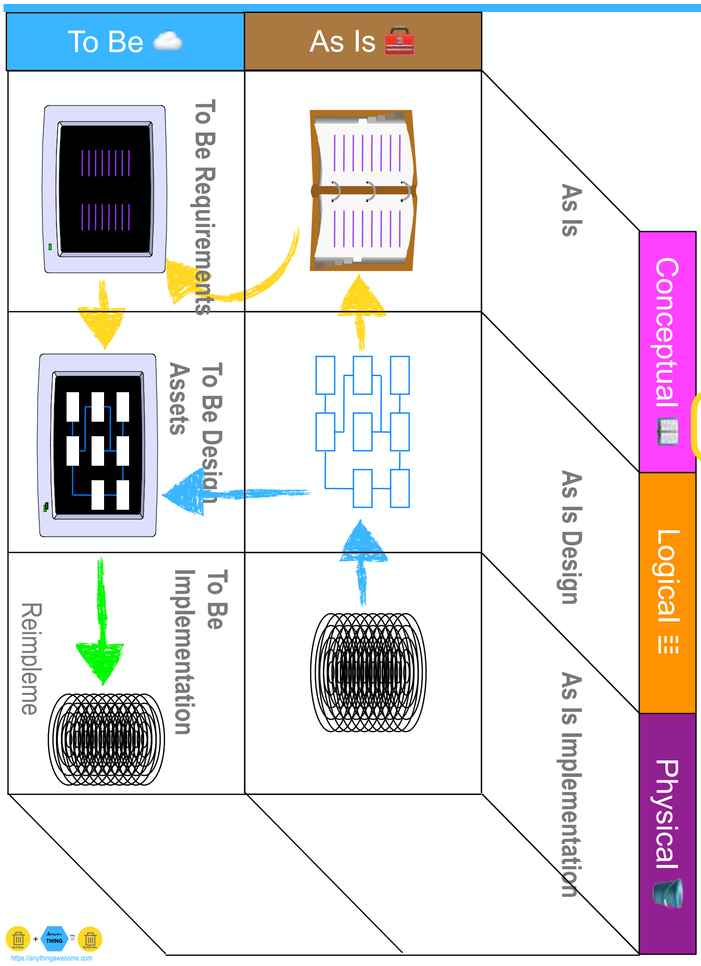
Every modeling change can be mapped to a transformation in this framework!

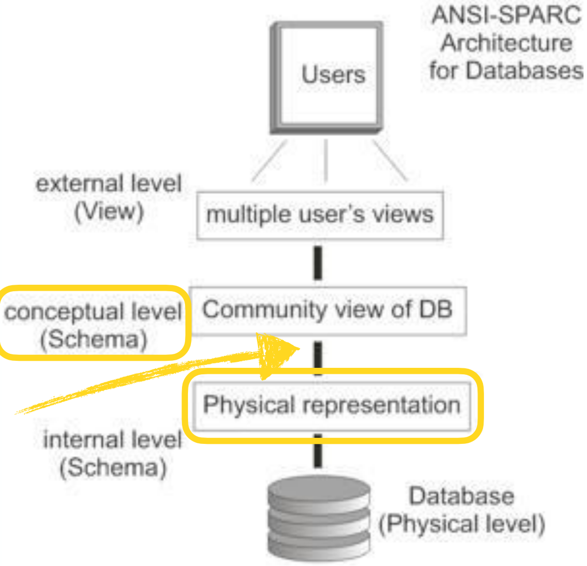
80% of IT Work Is Some Form of Reverse Engineering


Conceptual
Logical
Physical
As Is Requirements Assets WHAT?
As Is Design Assets HOW?
As Is Implementation Assets AS BUILT
As Is

Evolve existing systems using a structured technique aimed at recovering rigorous knowledge of the existing system to leverage enhancement efforts [Chikofsky & Cross 1990]

Turned on its Side



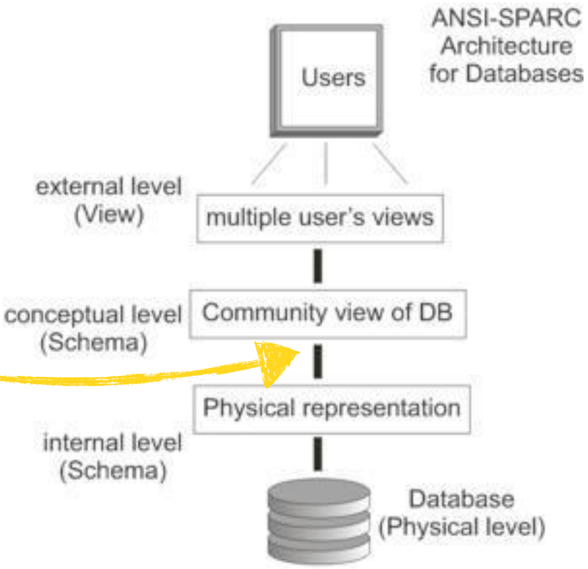


https://en.wikipedia.org/wiki/ANSI-SPARC_Architecture

Trusted Catalog

ANSI-SPARC 3-Layer Schema

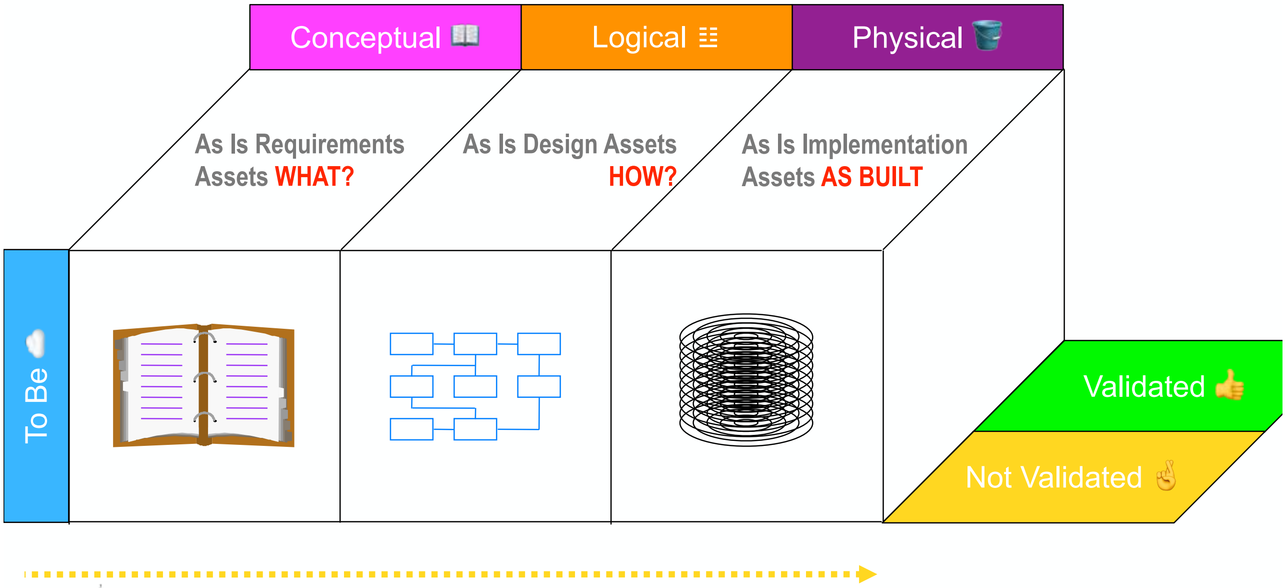
- **Conceptual** - Highest level of abstraction, focused on data requirements (what), linked directly to strategy
- **Logical** - Usually a refinement of conceptual model, focused on how data requirements are met using business terminology
- **Physical** - Implementation of the logical model with security, configuration management, and implementation specific details, specified via DDL



https://en.wikipedia.org/wiki/ANSI-SPARC_Architecture

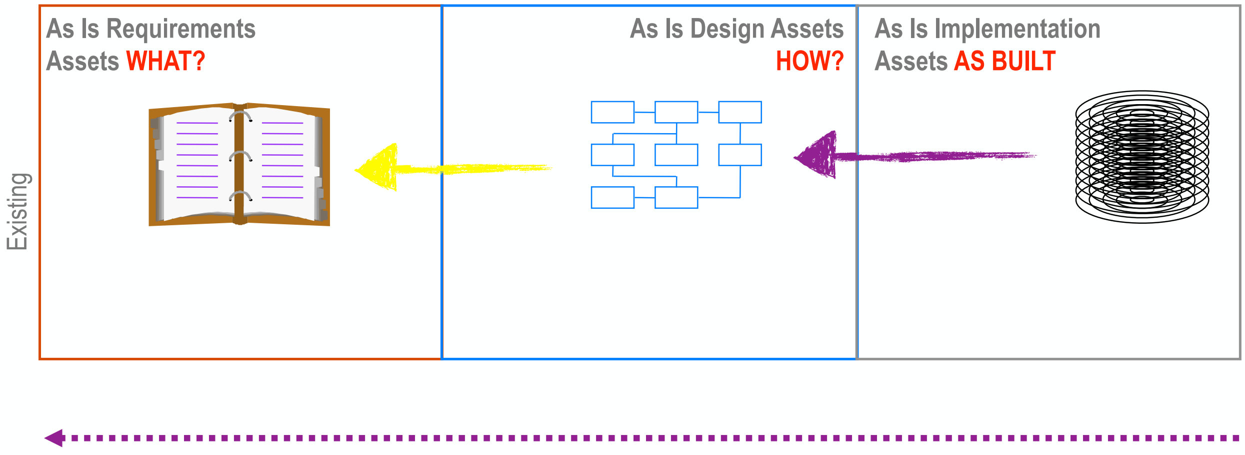
When changing to a new DBMS technology, the database administrator should be able to change the conceptual or global structure of the database without affecting the users

Forward Engineering



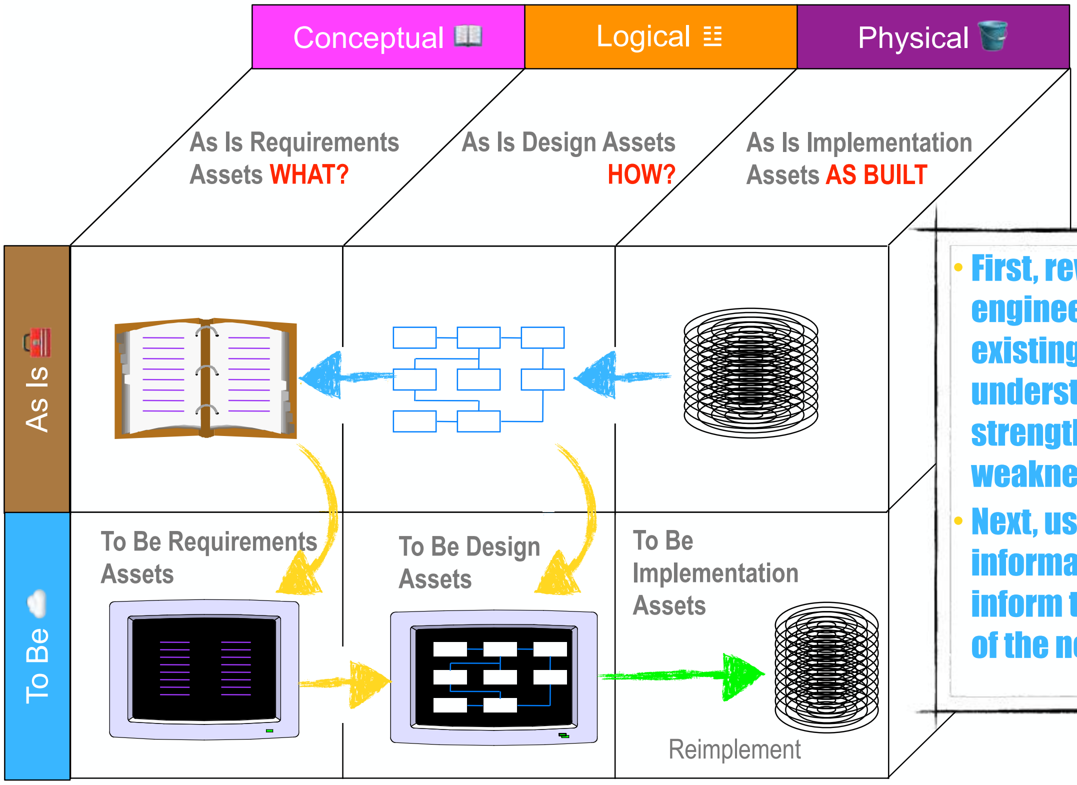
Building new stuff (20% of effort and funding)
Enhancing existing stuff (80%)

Reverse Engineering


As Is Requirements Assets WHAT?
As Is Design Assets HOW?
As Is Implementation Assets AS BUILT
Existing

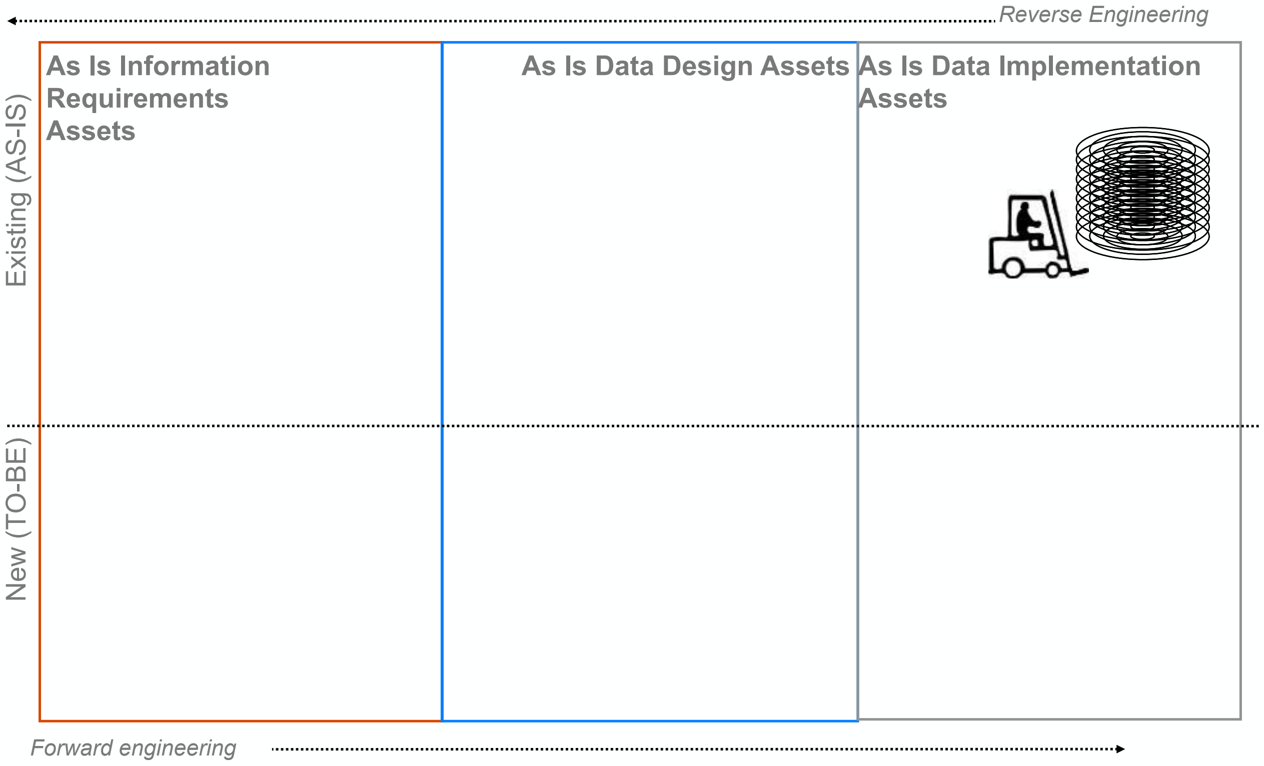
A structured technique aimed at recovering rigorous knowledge of the existing system to leverage enhancement efforts [Chikofsky & Cross 1990]

Reengineering

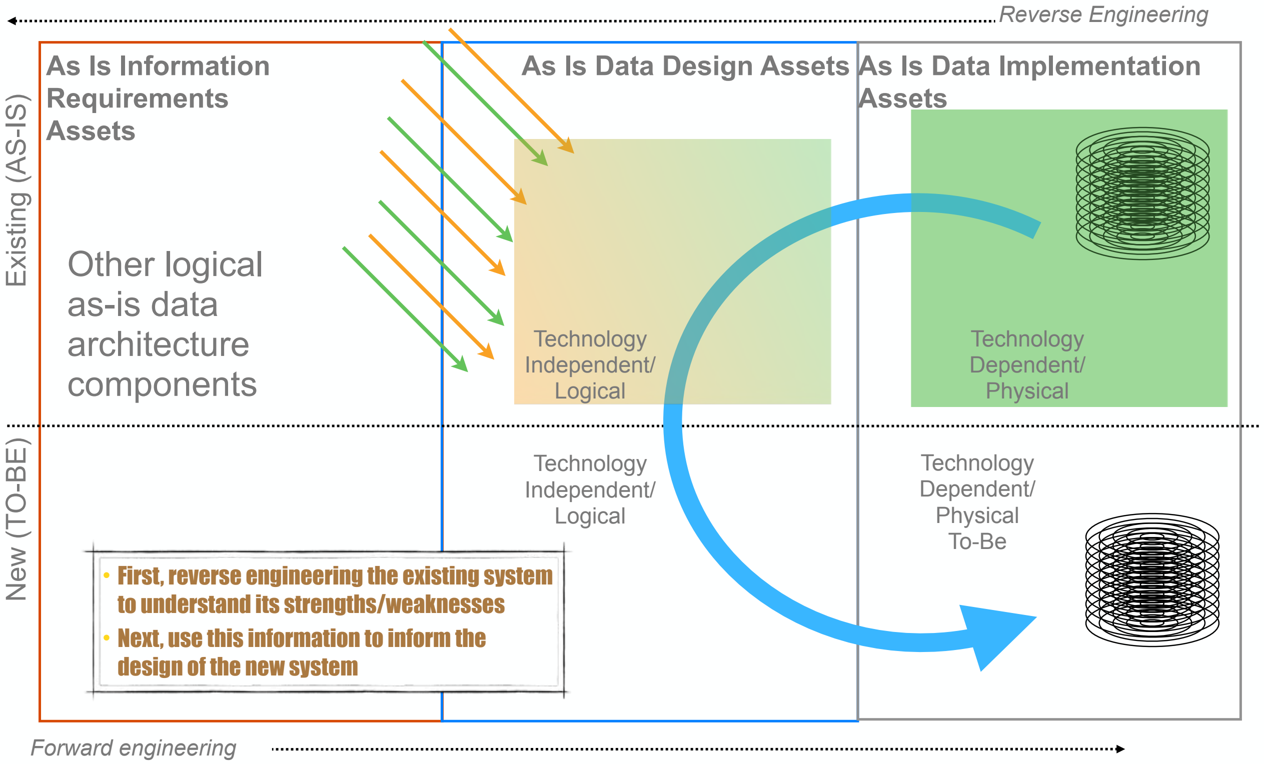

Conceptual 📖
Logical
Physical
As Is Requirements Assets WHAT?
As Is Design Assets HOW?
As Is Implementation Assets AS BUILT
As Is
To Be
To Be Requirements Assets
To Be Design Assets
To Be Implementation Assets
Reimplement

- First, reverse engineering the existing system to understand its strengths/ weaknesses
- Next, use this information to inform the design of the new system

Normalization (how not to do it but how it is always done)

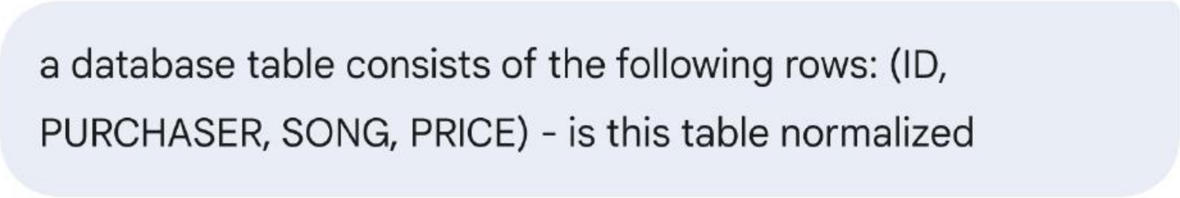


Normalization (better explanation)


Reverse Engineering
Existing (AS-IS)
New (TO-BE)
As Is Information Requirements Assets
As Is Data Design Assets
As Is Data Implementation Assets
Other logical as-is data architecture components
Technology Independent/ Logical
Technology Dependent/ Physical
Technology Independent/ Logical
Technology Dependent/ Physical To-Be
Forward engineering

- First, reverse engineering the existing system to understand its strengths/weaknesses
- Next, use this information to inform the design of the new system

gemini.google.com



a database table consists of the following rows: (ID, PURCHASER, SONG, PRICE) - is this table normalized

gemini.google.com

You're thinking about database normalization! It's a key concept in designing efficient and reliable databases.

Here's an analysis of your table (ID, PURCHASER, SONG, PRICE) and whether it's normalized:

Potential Issues:

- Data Redundancy: If a purchaser buys multiple songs, their information (name, potentially address, etc.) would be repeated for each song, leading to wasted space and potential inconsistencies.
- Update Anomalies: If a purchaser's information changes, you'd have to update it in multiple rows, increasing the risk of errors.
- Deletion Anomalies: If you delete a song, you might unintentionally delete information about the purchaser if it's their only purchase.

Normalization Steps:

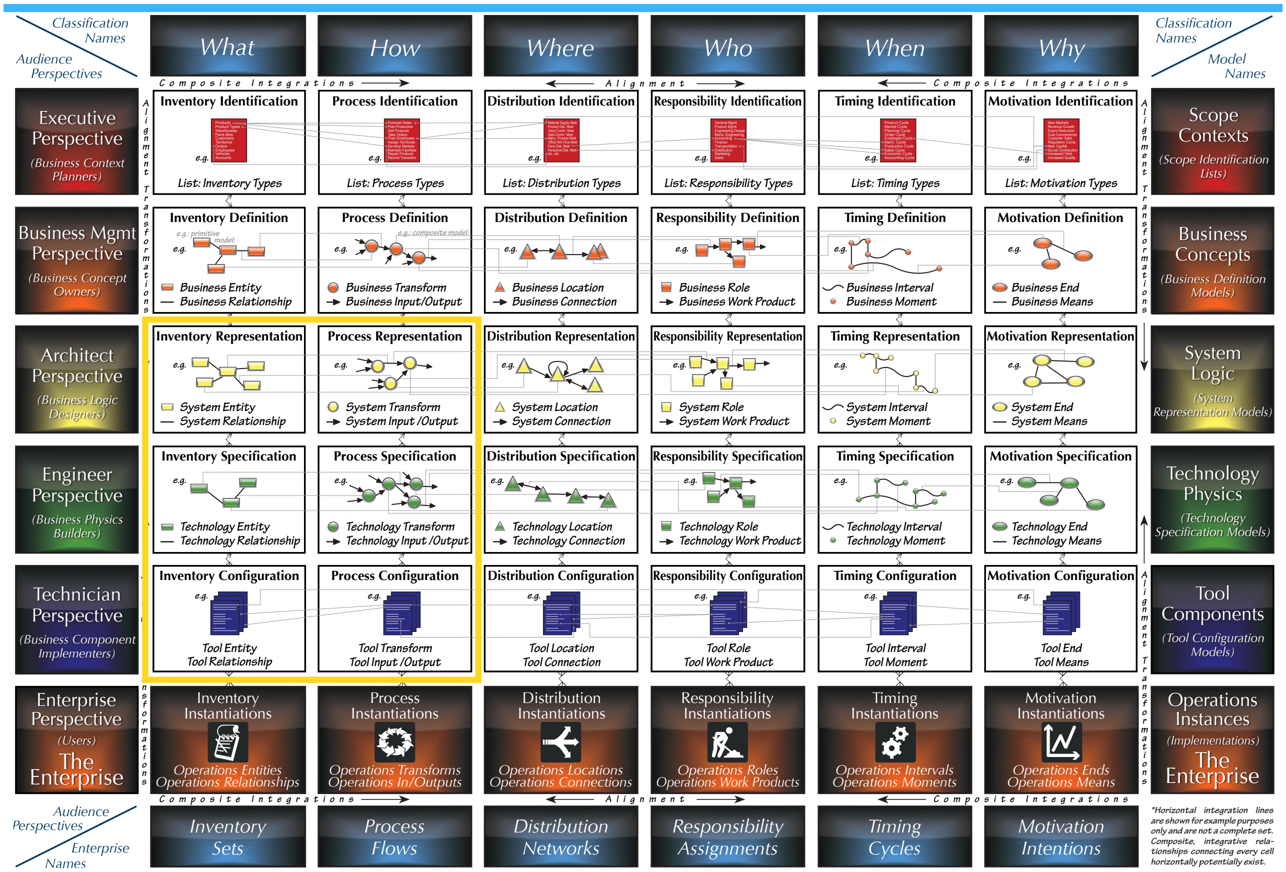
- To address these issues, you would typically normalize the table into multiple related tables:
- Purchaser Table:
- PurchaserID (primary key)
- PurchaserName
- PurchaserAddress
- ...other purchaser attributes
- Song Table:
- SongID (primary key)
- SongTitle
- Artist
- ...other song attributes
- Purchase Table:
- PurchaseID (primary key)
- PurchaserID (foreign key referencing Purchaser table)
- SongID (foreign key referencing Song table)
- Price
- PurchaseDate
- ...other purchase attributes

Benefits of Normalization:

- Reduced Data Redundancy: Purchaser and song information is stored only once, minimizing storage space and ensuring consistency.
- Improved Data Integrity: Updates and deletions are easier and less prone to errors.
- Increased Flexibility: You can easily add new attributes or entities without restructuring the entire database.

Normal Forms:

- This normalization process typically follows a set of "normal forms," with each form addressing specific types of redundancy and anomalies. The above example likely achieves at least Third Normal Form (3NF), which is often sufficient for most applications.
- In summary:
- The original table (ID, PURCHASER, SONG, PRICE) is likely not fully normalized. By breaking it down into separate tables for purchasers, songs, and purchases, you can create a more efficient, reliable, and scalable database structure.



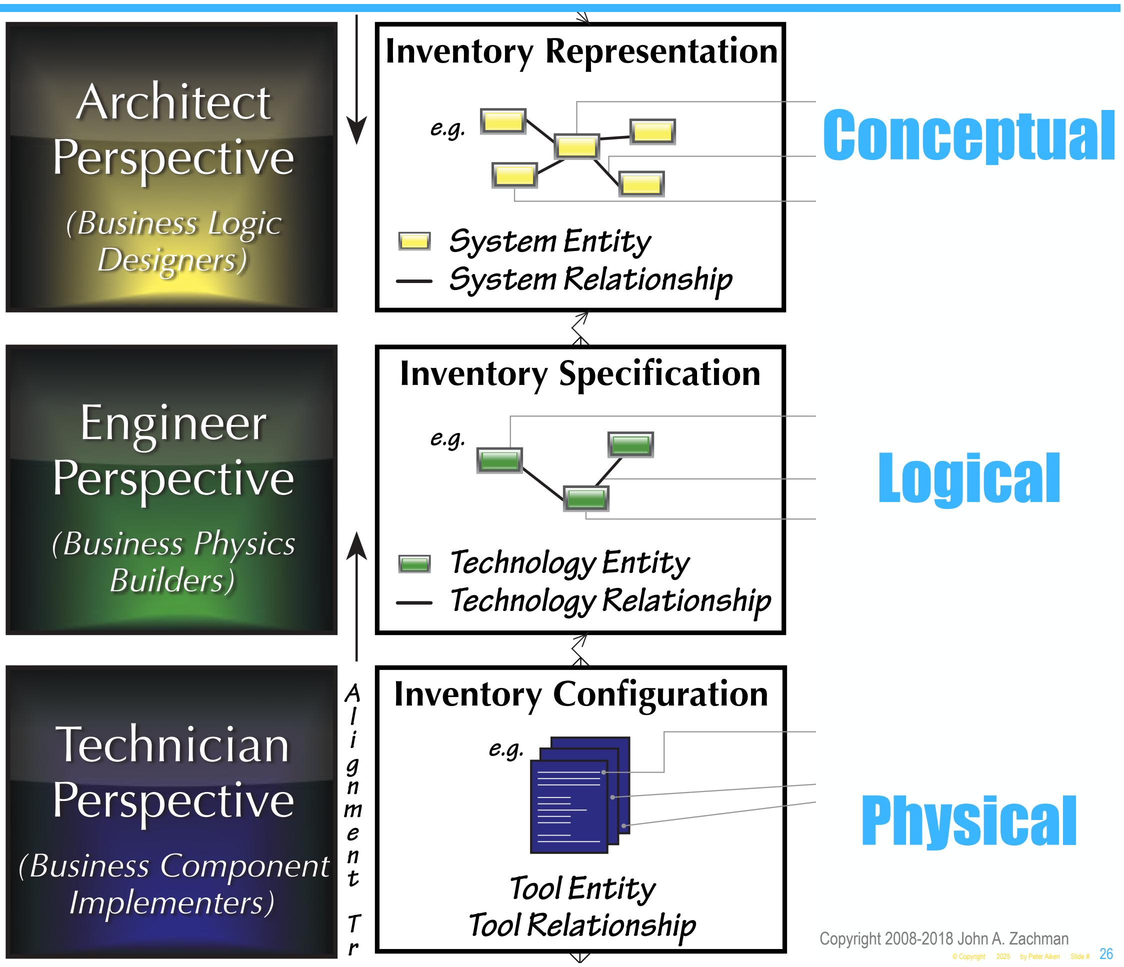
Classification Names / Audience Perspectives	What	How	Where	Who	When	Why	Classification Names / Model Names
Executive Perspective (Business Context Planners)	Inventory Identification e.g. List: Inventory Types	Process Identification e.g. List: Process Types	Distribution Identification e.g. List: Distribution Types	Responsibility Identification e.g. List: Responsibility Types	Timing Identification e.g. List: Timing Types	Motivation Identification e.g. List: Motivation Types	Scope Contexts (Scope Identification Lists)
Business Mgmt Perspective (Business Concept Owners)	Inventory Definition e.g.: primitive model: e.g. Business Entity Business Relationship	Process Definition e.g.: composite model: e.g. Business Transform Business Input/Output	Distribution Definition e.g. Business Location Business Connection	Responsibility Definition e.g. Business Role Business Work Product	Timing Definition e.g. Business Interval Business Moment	Motivation Definition e.g. Business End Business Means	Business Concepts (Business Definition Models)
Architect Perspective (Business Logic Designers)	Inventory Representation e.g. System Entity System Relationship	Process Representation e.g. System Transform System Input /Output	Distribution Representation e.g. System Location System Connection	Responsibility Representation e.g. System Role System Work Product	Timing Representation e.g. System Interval System Moment	Motivation Representation e.g. System End System Means	System Logic (System Representation Models)
Engineer Perspective (Business Physics Builders)	Inventory Specification e.g. Technology Entity Technology Relationship	Process Specification e.g. Technology Transform Technology Input /Output	Distribution Specification e.g. Technology Location Technology Connection	Responsibility Specification e.g. Technology Role Technology Work Product	Timing Specification e.g. Technology Interval Technology Moment	Motivation Specification e.g. Technology End Technology Means	Technology Physics (Technology Specification Models)
Technician Perspective (Business Component Implementers)	Inventory Configuration e.g. Tool Entity Tool Relationship	Process Configuration e.g. Tool Transform Tool Input /Output	Distribution Configuration e.g. Tool Location Tool Connection	Responsibility Configuration e.g. Tool Role Tool Work Product	Timing Configuration e.g. Tool Interval Tool Moment	Motivation Configuration e.g. Tool End Tool Means	Tool Components (Tool Configuration Models)
Enterprise Perspective (Users) The Enterprise	Inventory Instantiations Operations Entities Operations Relationships	Process Instantiations Operations Transforms Operations In/Outputs	Distribution Instantiations Operations Locations Operations Connections	Responsibility Instantiations Operations Roles Operations Work Products	Timing Instantiations Operations Intervals Operations Moments	Motivation Instantiations Operations Ends Operations Means	Operations Instances (Implementations) The Enterprise
Audience Perspectives / Enterprise Names	Inventory Sets	Process Flows	Distribution Networks	Responsibility Assignments	Timing Cycles	Motivation Intentions	

*Horizontal integration lines are shown for example purposes only and are not a complete set. Composite, integrative relationships connecting every cell horizontally potentially exist.

Zachman Framework

https://anythingawesome.com

Rows 3, 4, & 5 of the "What" Column



Architect Perspective
(Business Logic Designers)

Inventory Representation
e.g.
System Entity
System Relationship

Conceptual

Engineer Perspective
(Business Physics Builders)

Inventory Specification
e.g.
Technology Entity
Technology Relationship

Logical

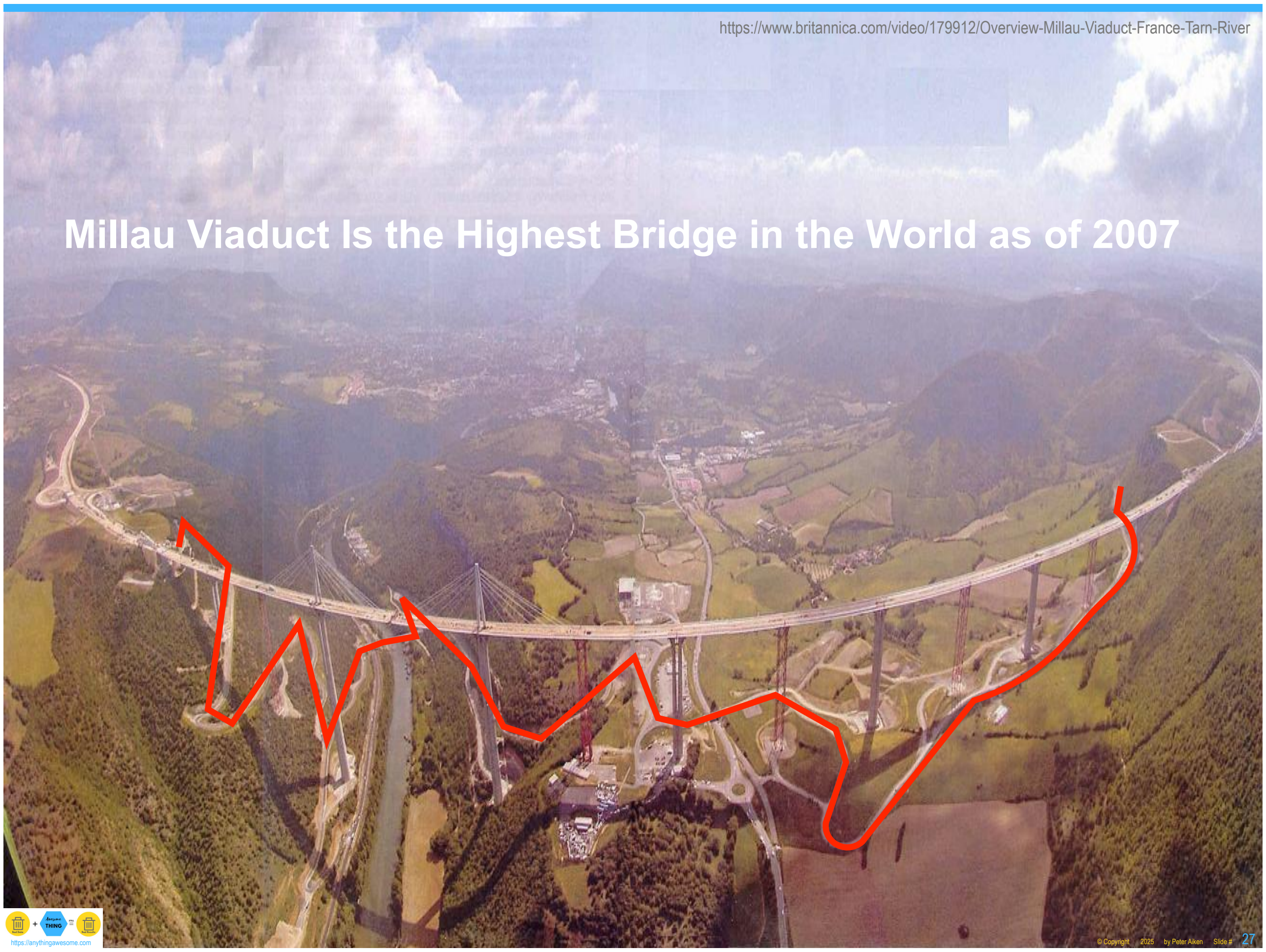
Technician Perspective
(Business Component Implementers)

Inventory Configuration
e.g.
Tool Entity
Tool Relationship

Physical

https://anythingawesome.com

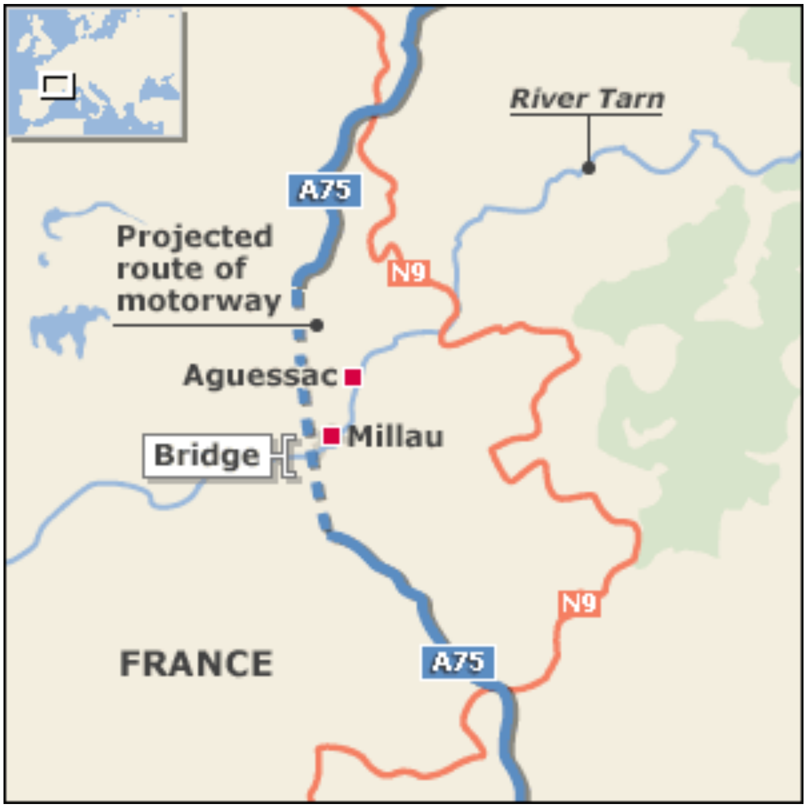
Millau Viaduct Is the Highest Bridge in the World as of 2007



https://www.britannica.com/video/179912/Overview-Millau-Viaduct-France-Tarn-River

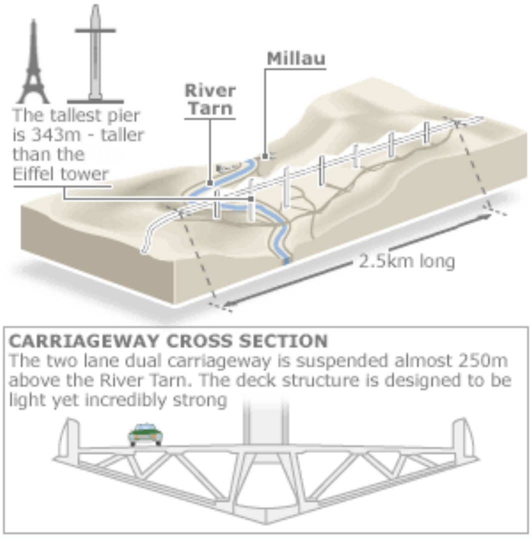
Conceptual Models

- Business focused
- Entity level
- Provides focus, scope, and guidance to modeling effort
- Sometimes thrown away - rarely maintained

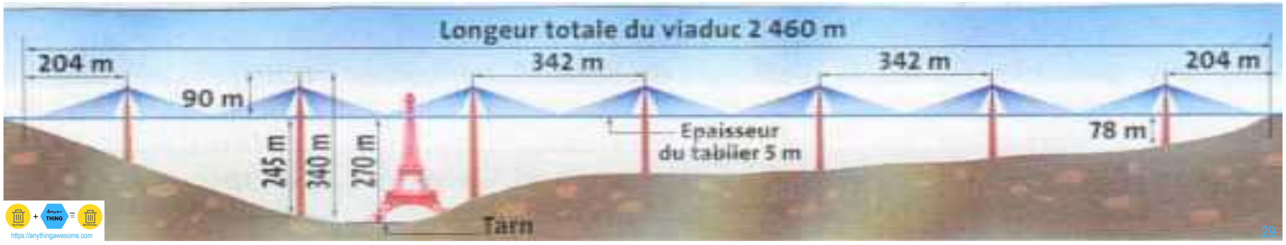


Logical Models

- Required to achieve the transition from conceptual to physical
- Developed to the attribute level and understood at 3rd normal form
- Logical models are developed to be refined to until it becomes a solution - sometimes purchased (as in EDW) always requires tailoring
- Used to guarantee the rigor of the data structures by formally describing the relationship between data items in a strong fashion
- More often maintained



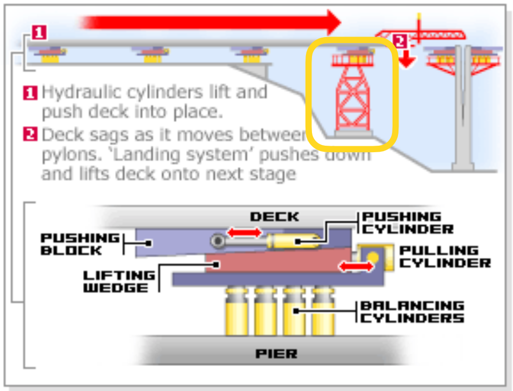
CARRIAGEWAY CROSS SECTION
The two lane dual carriageway is suspended almost 250m above the River Tarn. The deck structure is designed to be light yet incredibly strong



Physical Models

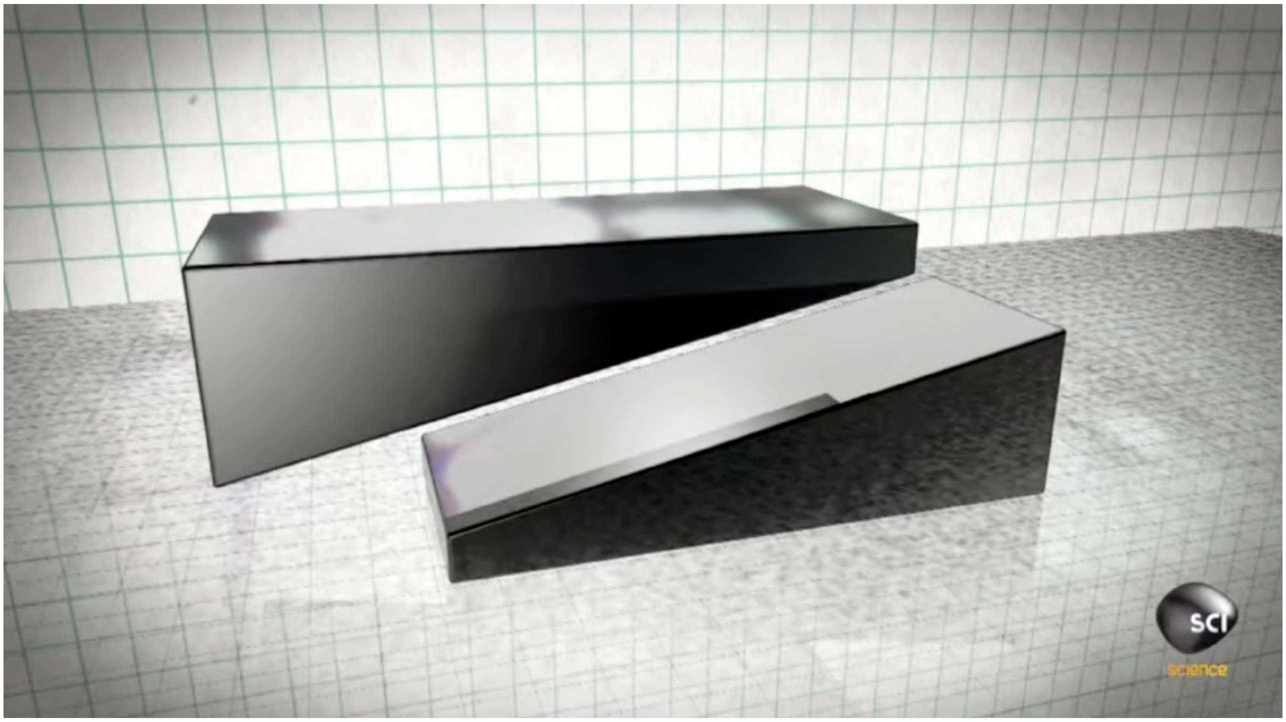
- Become the blueprints for physical construction of the solution
- Blueprints are used for future maintenance of the solution

Temporary piers
Bridge pylon

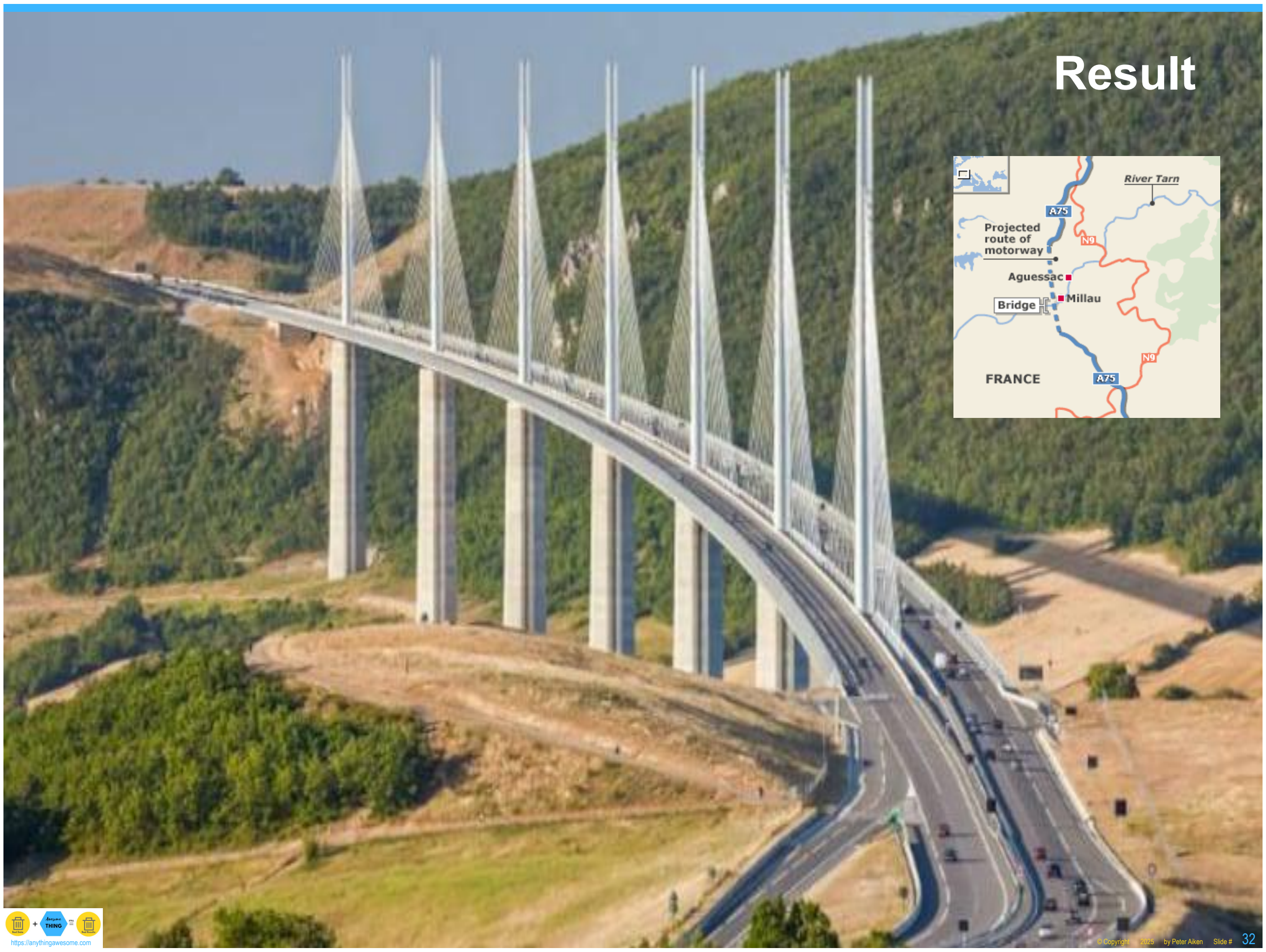


1 Hydraulic cylinders lift and push deck into place.
2 Deck sags as it moves between pylons. 'Landing system' pushes down and lifts deck onto next stage

Avoiding any Side-Pressure on the Supporting Piers



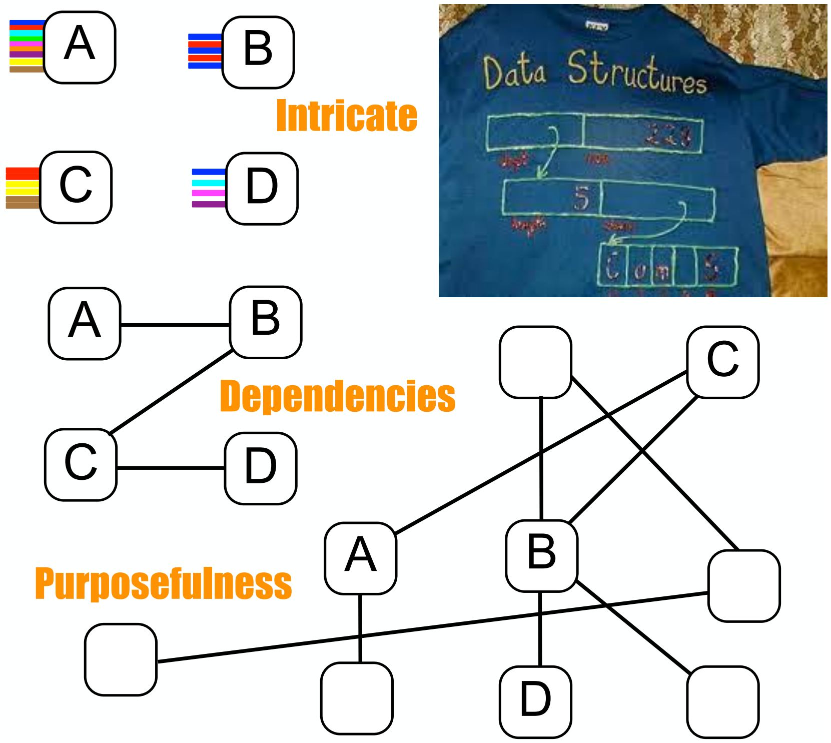
https://www.youtube.com/watch?v=_iK0solvjv8 & https://www.youtube.com/watch?v=DlbTNJ0AU1Y



Result

How Are Components Expressed as Architectures?

- Details are organized into larger components
- Larger components are organized into models
- Models are organized into architectures (composed of architectural components)



How Are Data Structures Expressed as Architectures?

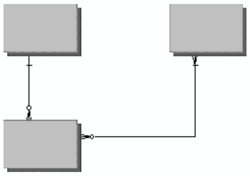
- Attributes are organized into entities/objects
  - Attributes are characteristics of "things"
  - Entitles/objects are "things" whose information is managed in support of strategy
  - Example(s)

Intricate

THING
Club.Id #
Club.Description
Club.Status
Club.Sex.To.Be.Assigned
Club.Reserve.Reason

- Entities/objects are organized into models
  - Combinations of attributes and entities are structured to represent information requirements
  - Poorly structured data, constrains organizational information delivery capabilities
  - Example(s)

Dependencies

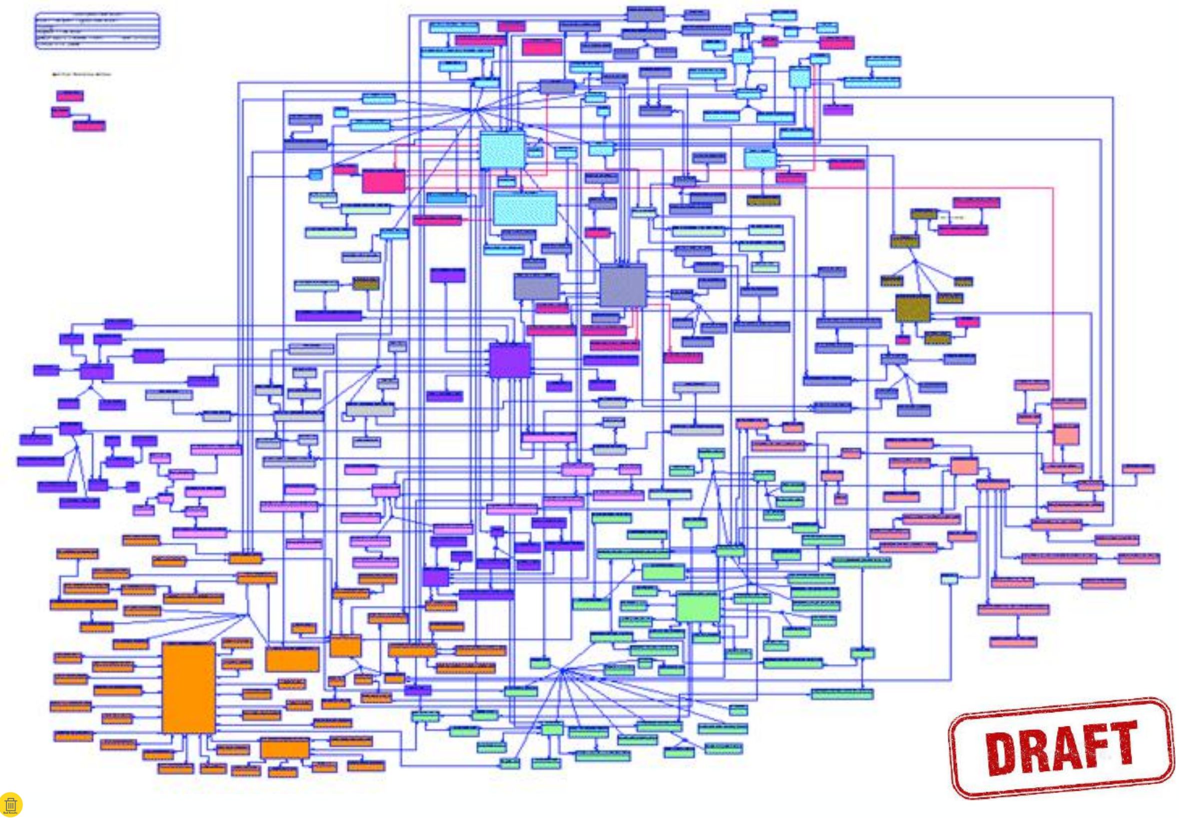


- Models are organized into architectures

Purposefulness

  - When building new systems, architectures are used to plan development
  - More often, data managers do not know what existing architectures are and - therefore - cannot make use of them in support of strategy implementation
  - *Why no examples?*

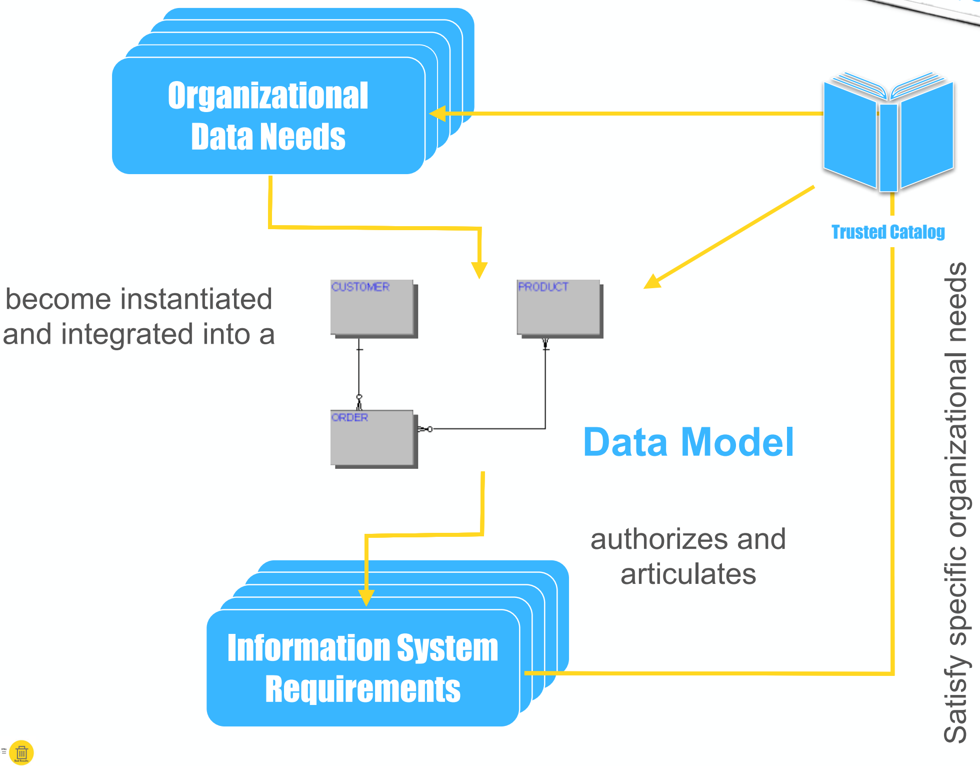
Data Architectures Are Composed of Data Models



DRAFT

Data Modeling Is Iterative

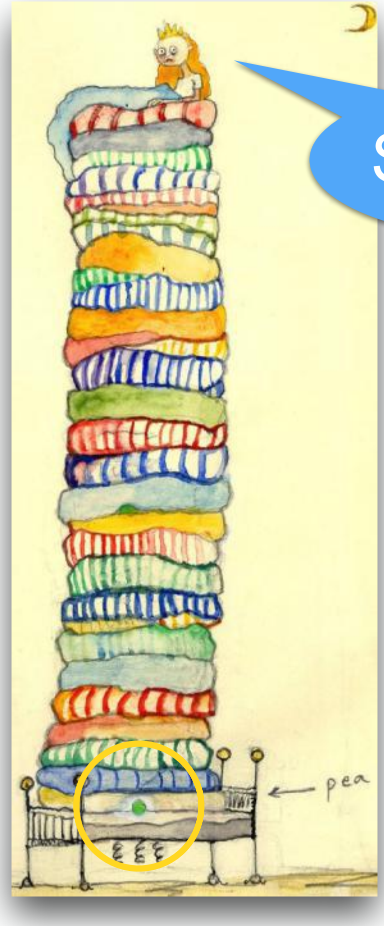
Data models and data architectures are developed in response to needs



Organizational Data Needs
Trusted Catalog
become instantiated and integrated into a
CUSTOMER
PRODUCT
ORDER
Data Model
authorizes and articulates
Information System Requirements
Satisfy specific organizational needs

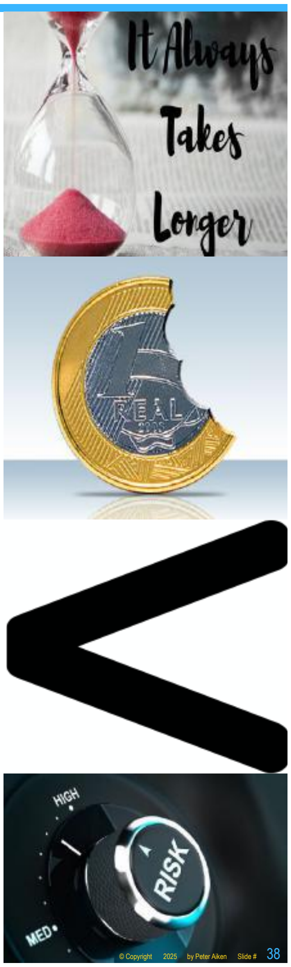
The Princess on the Pea

by
Hans Christian
Andersen



Doing a Poor Job With Data Modeling

- Failure to understand the role of data governance re: proposed and existing software/services
  - Locks in imperfections for the life of the application
  - Restricts data investment benefits
  - Decreases organizational data leverage
- Accounts for 20-40% of IT budgets devoted to evolving
  - Data migration (Changing the data location)
  - Data conversion (Changing data form, state, or product)
  - Data improving (Inspecting and manipulating, or re-keying data to prepare it for subsequent use)
- Lack of data governance causes everything else to
  - Take longer
  - Cost more
  - Deliver less
  - Present greater risk (with thanks to Tom DeMarco)



(A Hypothetical Portion of the) iTunes → Music™ Database

- Question:
  - What information is lost if we delete record #1?

Row	Purchaser ID	Song	Price
1	Peter	We Met Today	$0.99
2	Peter	My Mother's Voice	$1.29
3	Peter	Fortune Smiles	$0.99
4	Lolly	Thousand Pieces of Gold	$0.99

(A Hypothetical Portion of the) Music™ Database: Deletion Anomaly

- Question:
  - What information is lost if we delete record #1?
- Answer:
  - We loose the fact that Peter purchased "We Met Today"
  - We also loose the fact that "We Met Today" costs $0.99
  - These are usually undesirable and unintended

(Deleted)

Row	Purchaser ID	Song	Price
1	Peter	We Met Today	$0.99
2	Peter	My Mother's Voice	$1.29
3	Peter	Fortune Smiles	$0.99
4	Lolly	Thousand Pieces of Gold	$0.99

Music™ Database: Insertion Anomalies

- Question:
  - Suppose we want to add new song SCUBA and that it costs $1.29?
- Answer:
  - Cannot enter it until a purchaser buys SCUBA
  - We cannot insert a full row until we have an additional fact about that row
  - This is usually undesirable and unintended

Row	Purchaser ID	Song	Price
1	Peter	We Met Today	$0.99
2	Peter	My Mother's Voice	$1.29
3	Peter	Fortune Smiles	$0.99
4	Lolly	Thousand Pieces of Gold	$0.99
5	???	SCUBA	$1.29

Music™ Database: Update Anomalies

- Question:
  - Suppose we want to increase the price of 'We Met Today' from $0.99 to $1.29?
- Answer:
  - Change to data items such as Song requires examination of every single record
  - Will not catch spelling errors - such as "We met Toddy"
  - This is usually undesirable and unintended

Row	Purchaser ID	Song	Price
1	Peter	We Met Todday	$0.99
2	Peter	My Mother's Voice	$1.29
3	Peter	Fortune Smiles	$0.99
4	Lolly	Thousand Pieces of Gold	$0.99
5	???	SCUBA	$1.29

There Are Correct Ways To Organize Data

- Optimization can be done for:
  - Flexibility
  - Adaptability
  - Retrievability
  - Risk reduction
  - ...
- Techniques include:
  - Data integrity
  - Smart codes bad/dumb codes good
  - Architecture (table joins)
  - ...

ORIGINAL

Record	Purchaser ID	Song	Pric
1	Purchaser #1	Cool Walk (Live)	$1.99
2	Purchaser #1	Sushi (Live)	$0.99
3	Purchaser #1	Love Ballade (Live)	$0.99
4	Purchaser #2	A Salute to Bach	$0.99
5	Purchaser #3	Coolwalk (Live)	$1.99

How Should It Be Done? (In General)

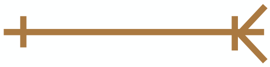
- As much as possible, store 1 fact per row
  - Row 2 is a good example as it shows both that Purchaser #1 has purchased Sushi (Live) and that it costs $0.99
  - These are two distinct facts and are correctly stored in two tables sharing a formal relationship
  - More remains coded

ORIGINAL

Record	Purchaser ID	Song	Pric
1	Purchaser #1	Cool Walk (Live)	$1.99
2	Purchaser #1	Sushi (Live)	$0.99
3	Purchaser #1	Love Ballade (Live)	$0.99
4	Purchaser #2	A Salute to Bach	$0.99
5	Purchaser #3	Coolwalk (Live)	$1.99

PRICING

Record	Song	Price
1	Cool Walk (Live)	$1.99
2	Sushi (Live)	$0.99
3	Love Ballade (Live)	$0.99
4	A Salute to Bach	$0.99
5	Coolwalk (Live)	$1.99



PURCHASES

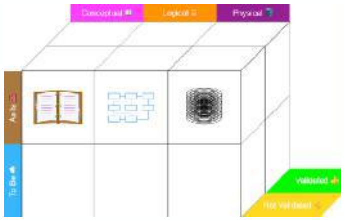
Row	Purchaser ID	Song
1	Purchaser #1	Cool Walk (Live)
2	Purchaser #1	Sushi (Live)
3	Purchaser #1	Love Ballade (Live)
4	Purchaser #2	A Salute to Bach (Medley)
5	Purchaser #3	Coolwalk (Live)
6	Purchaser #3	A Salute to Bach (Medley)

Sand in the Machinery

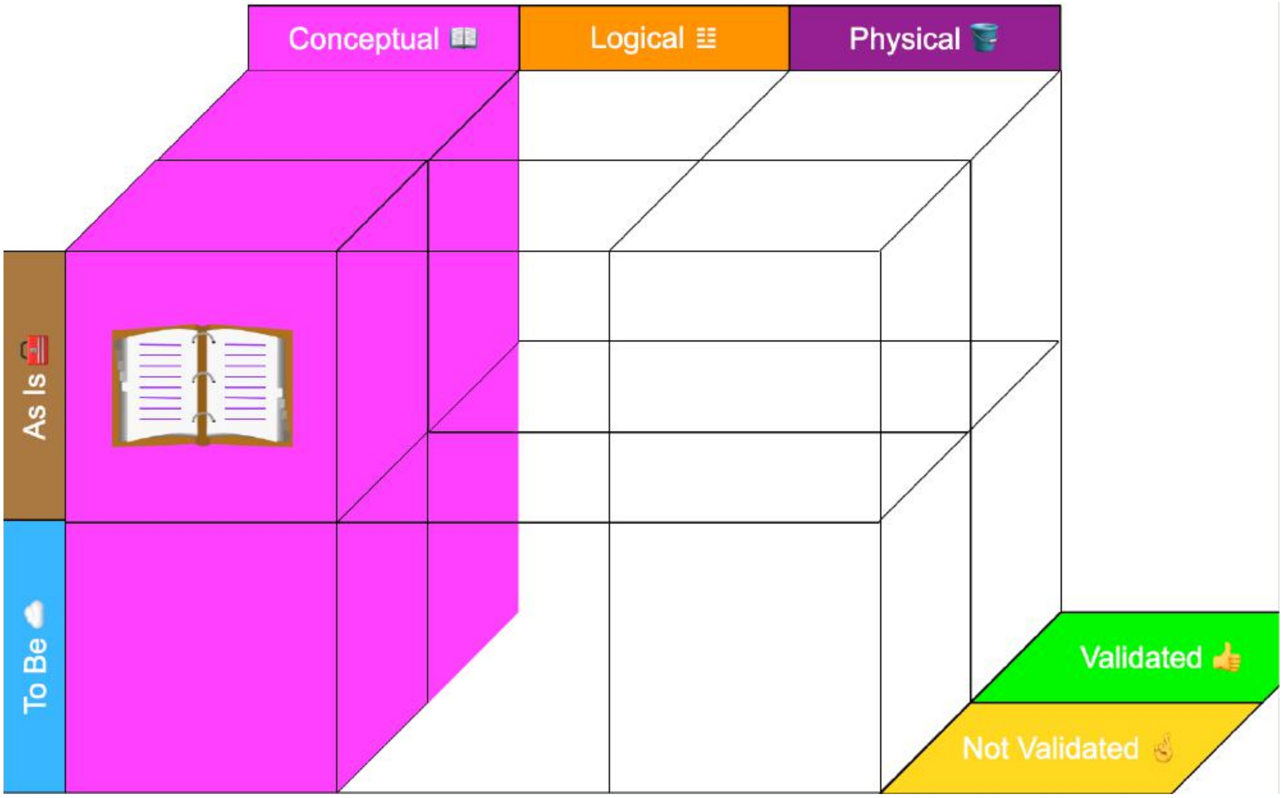
Program Overview

Conceptual
Versus
Logical
Versus
Physical
Data Modeling

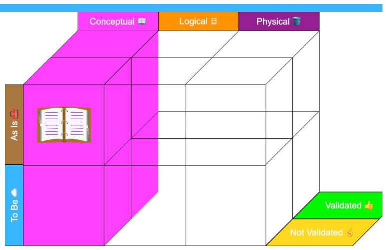
- Introduction to Modeling Data
  - Motivation
  - 3 primary data model types (+ plus two characteristics)
  - Reasons for each
  - Purposeful Modeling Basics (conversions, forward/reverse engineering)
- Conceptual
  - Motivation: Architectural tradeoffs
  - Strategy and conceptual data modeling
  - Glossary/Dictionary capabilities
- Logical
  - Motivation: Simplicity (Operational and Design)
  - Motivation towards standards
  - Business meets strategy
- Physical
  - Motivation: Required documentation and/or facts
  - Become the blueprints for physical construction of the solution
  - Blueprints are used for future maintenance of the solution
- Take Aways/References/Q&A



Conceptual Data Modeling



Conceptual Data Modeling



Motivation

- Harmonize/standardize vocabulary
  - Between business and technologists
  - Between humans and systems
- Focus consideration/analyses on strategic issues and tradeoffs
- Provide specifications comprising organizational data strategic objectives
- Document data requirements satisfying business objectives

Reasons for Unvalidated Conceptual Data Models

- Unvalidated models require the word ‘draft’ on them, indicating a lack of certainty
- Useful for organizing data concepts
- Hypothesizing the relationship of various data things to various other data things

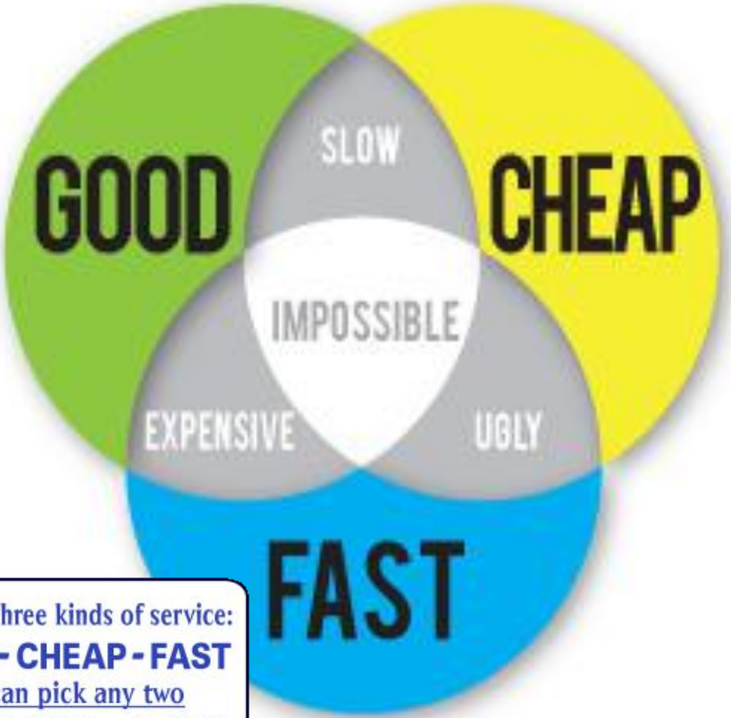
DRAFT

Reasons for Validated Conceptual Data Models

- Documenting the relationship of various data things to various other data things
- Standardizing on 'system-wide' definitions
- Understanding high level process interactions

Architecture Involves at Least ...

- Analysis/model evaluation
- Risk evaluation
- Volume considerations
- Workload forecasting
- Tradeoff analysis
- ...



We offer three kinds of service:

GOOD - CHEAP - FAST

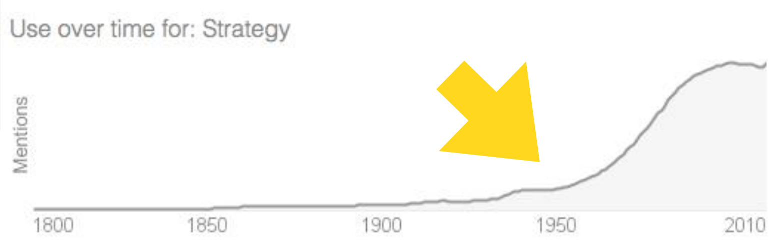
You can pick any two

GOOD service CHEAP won't be FAST

GOOD service FAST won't be CHEAP

FAST service CHEAP won't be GOOD

What Is Strategy?



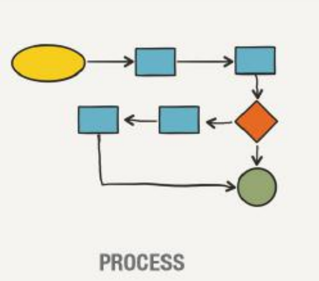
strat·e·gy

/ˈstratəjē/

noun

1. a plan of action or policy designed to achieve a major or overall aim.
   "time to develop a coherent economic strategy"
   synonyms: master plan, grand design, game plan, plan (of action), action plan, policy, program; More

- Current use derived from military
  - **a pattern in a stream of decisions**
    [Henry Mintzberg]



Former Walmart Business Strategy

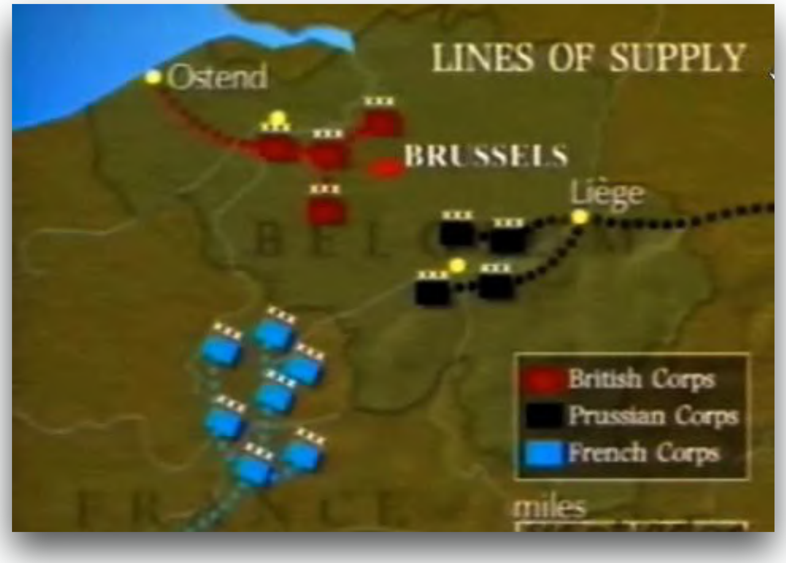
Every Day Low Price

Wayne Gretzky's Strategy

He skates to where he thinks the puck will be ...

Strategy in Action: Napoleon Faces a Larger Enemy

- Question?
  - How do I defeat the competition when their forces are bigger than mine?
- Answer:
  - Divide and conquer!
  - **“a pattern in a stream of decisions”**



Complex Strategy

- First
  - Hit both armies hard at just the right spot

- Then
  - Turn ri
def

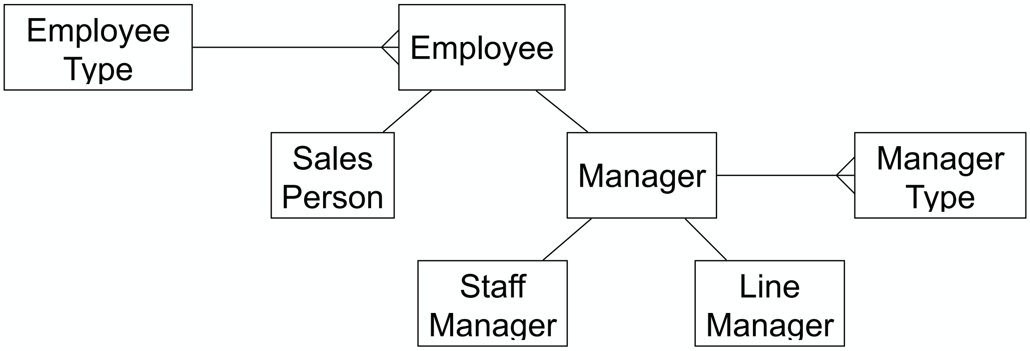
While someone is shooting at you!

-
  - Turn left and defeat the British

Data Models Used To Support Strategy

- Flexible, adaptable data structures
- Cleaner, less complex code
- Ensure strategy effectiveness measurement
- Build in future capabilities
- Form/assess merger and acquisitions strategies

Efficiencies



Adapted from Clive Finkelstein *Information Engineering Strategic Systems Development* 1992

Strategic Use of Data Models (Other Examples)

- SABRE creates flight booking business

Sabre

An innovation technology company

AT&T

- AT&T invents the "new" credit card business overnight
- Amazon invents at home retailing

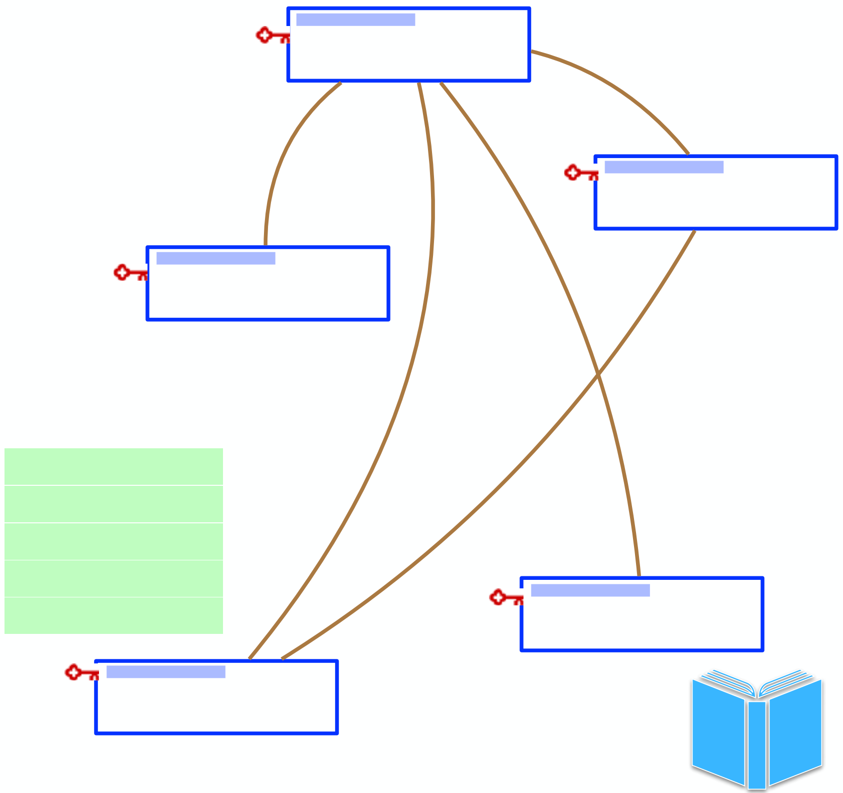
amazon.com

- CapitalOne reinvents solicitation

CapitalOne | what's in your wallet?

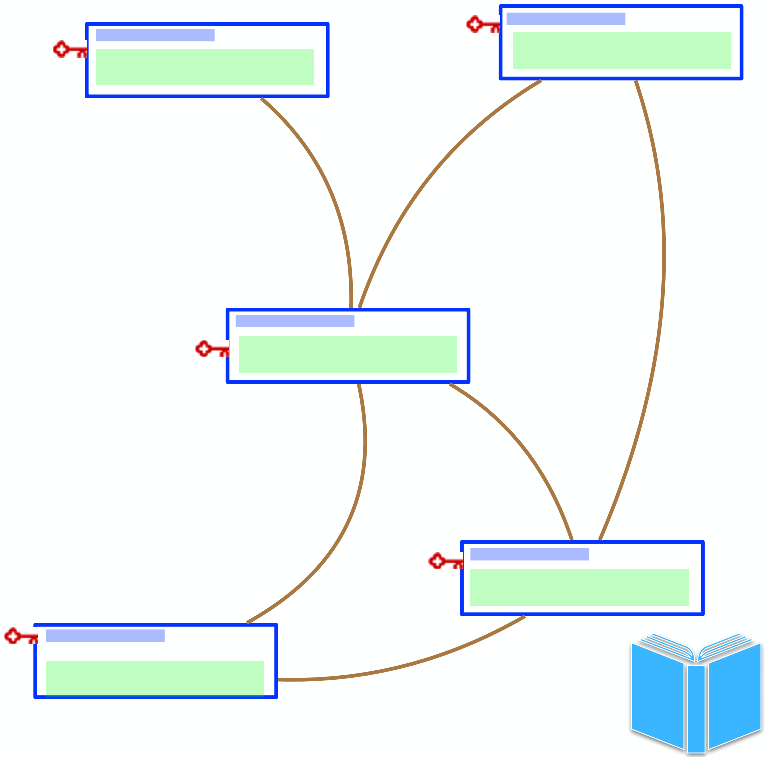
Data Modeling Process

1. Identify entities
2. Identify key for each entity
3. Draw rough draft of entity relationship data model
4. Identify data attributes
5. Map data attributes to entities



Model Evolution Is Good, at First ...

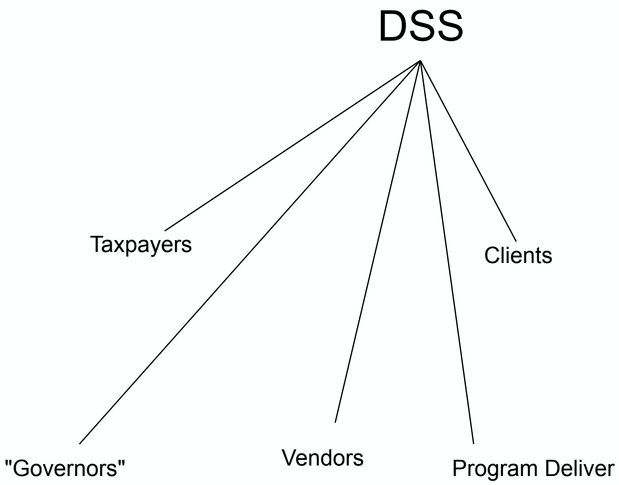
1. Identify entities
2. Identify key for each entity
3. Draw rough draft of entity relationship data model
4. Identify data attributes
5. Map data attributes to entities



This Logical Data Model Is Comprised of 5-Model Views

DSS Strategic Data Model

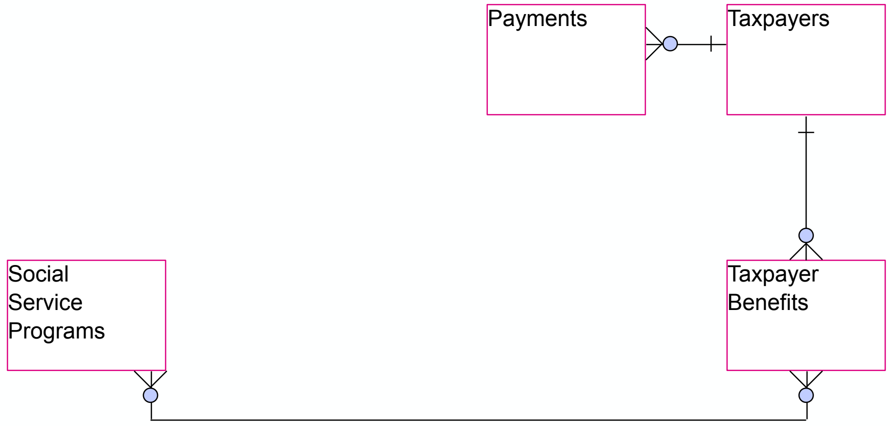
- Taxpayer view
- Client view
- Governance view
- Program Delivery view
- Vendor view



DRAFT

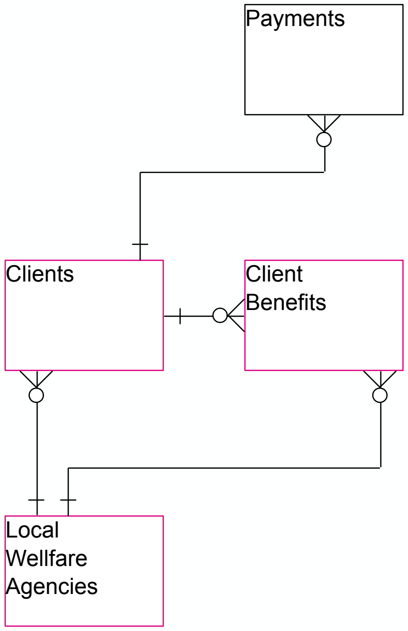
(Please note that all models are currently unvalidated and should be consider as "draft" version until they are validated!)

Taxpayer View



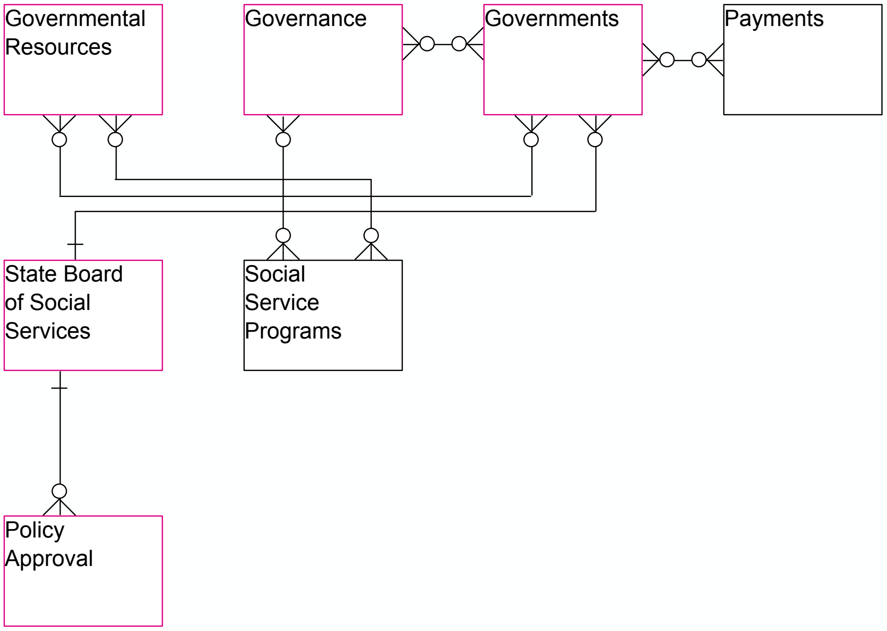
DRAFT

Client View



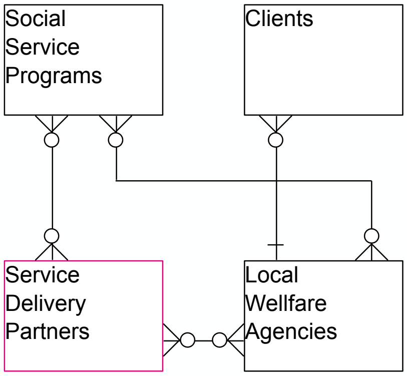
DRAFT

Governance View



DRAFT

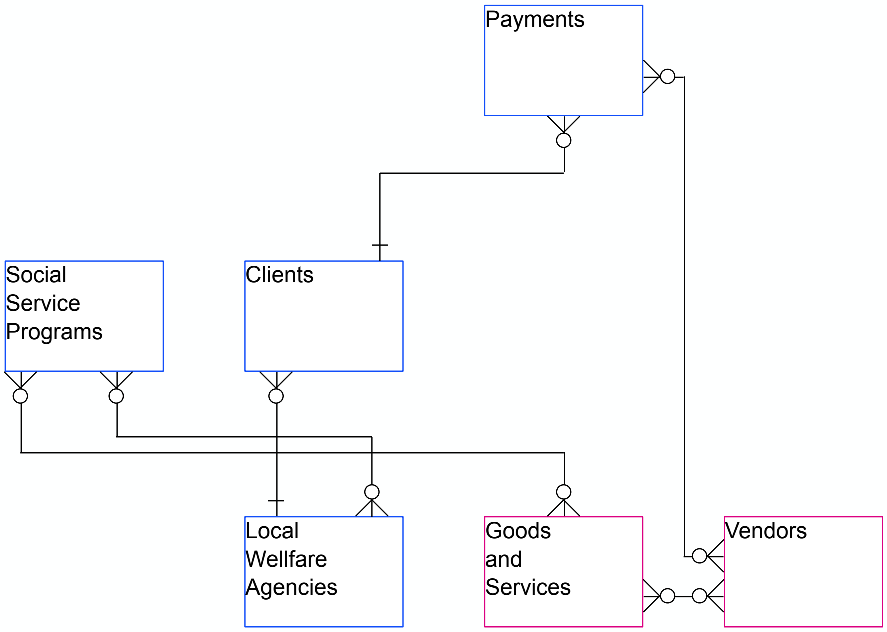
Program Delivery View



DRAFT

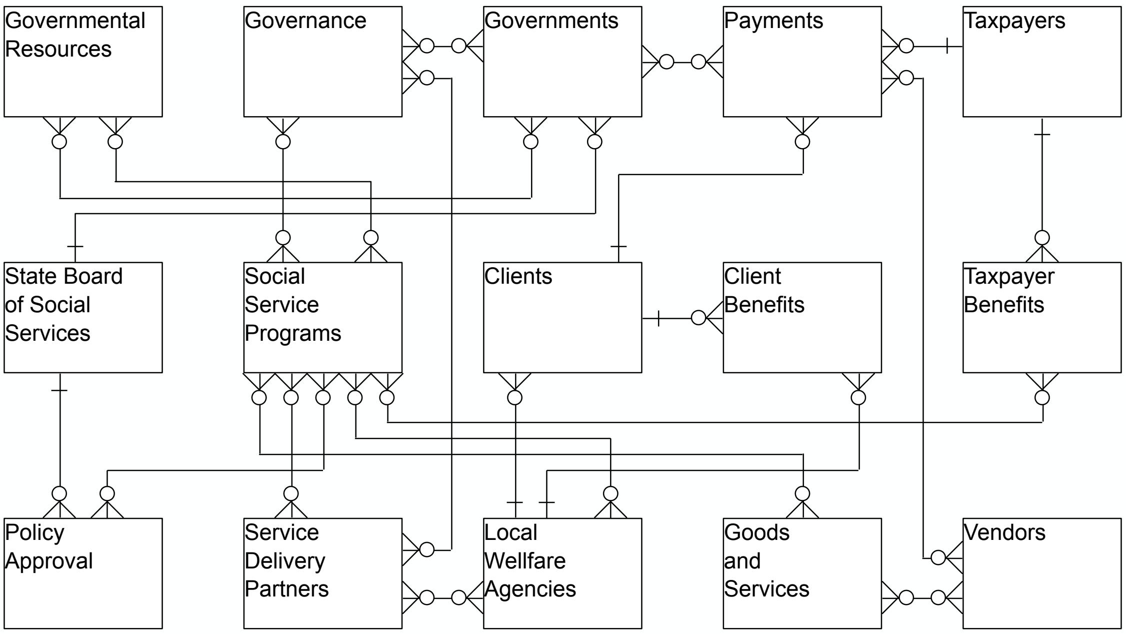
Vendor View

DRAFT



DSS Conceptual Data Model

DRAFT



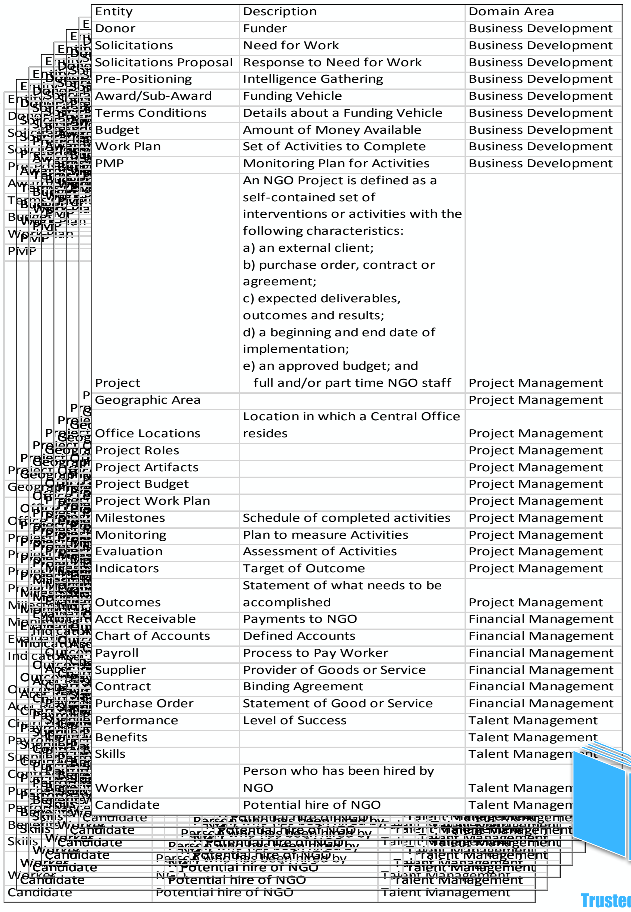
Business Glossary

- Start of enterprise taxonomy

(Previous Versions)

- Defines initial entities for conceptual data model

- Engages the business community to validate entities and provide meaningful business definitions



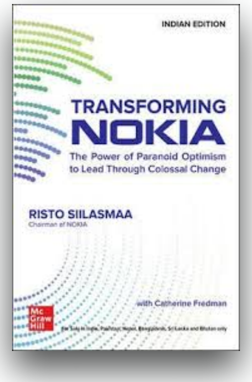
Entity	Description	Domain Area
Donor	Funder	Business Development
Solicitations	Need for Work	Business Development
Solicitations Proposal	Response to Need for Work	Business Development
Pre-Positioning	Intelligence Gathering	Business Development
Award/Sub-Award	Funding Vehicle	Business Development
Terms Conditions	Details about a Funding Vehicle	Business Development
Budget	Amount of Money Available	Business Development
Work Plan	Set of Activities to Complete	Business Development
PMP	Monitoring Plan for Activities	Business Development
Project	An NGO Project is defined as a self-contained set of interventions or activities with the following characteristics: a) an external client; b) purchase order, contract or agreement; c) expected deliverables, outcomes and results; d) a beginning and end date of implementation; e) an approved budget; and full and/or part time NGO staff	Project Management
Geographic Area		Project Management
Office Locations	Location in which a Central Office resides	Project Management
Project Roles		Project Management
Project Artifacts		Project Management
Project Budget		Project Management
Project Work Plan		Project Management
Milestones	Schedule of completed activities	Project Management
Monitoring	Plan to measure Activities	Project Management
Evaluation	Assessment of Activities	Project Management
Indicators	Target of Outcome	Project Management
Outcomes	Statement of what needs to be accomplished	Project Management
Acct Receivable	Payments to NGO	Financial Management
Chart of Accounts	Defined Accounts	Financial Management
Payroll	Process to Pay Worker	Financial Management
Supplier	Provider of Goods or Service	Financial Management
Contract	Binding Agreement	Financial Management
Purchase Order	Statement of Good or Service	Financial Management
Performance	Level of Success	Talent Management
Benefits		Talent Management
Skills		Talent Management
Worker	Person who has been hired by NGO	Talent Managem
Candidate	Potential hire of NGO	Talent Managem

Trusted Catalog

NOKIA

(Pre Microsoft Acquisition)

- Tires, rubber products
- Consumer electronics
- Mobile phones
  - Finns are bilingual (2% of population speaks Swedish)
  - Nokia wanted to play internationally
  - English mandated in all business settings
  - Lots of words were unknown
  - Culturally: Bad to not ask questions
  - Culturally: Good to build common vocabulary
- When an unfamiliar term was used
  - Group: Access NTB to see if there existed a golden definition
  - Group: If not, vote whether to submit it for inclusion in the NTB
  - Weekly: the NTB group reviewed submissions
  - Weekly: the NTB group published new versions of the NTB
  - NTB = Nokia Term Bank



INDIAN EDITION

TRANSFORMING NOKIA

The Power of Paranoid Optimism to Lead Through Colossal Change

RISTO SIILASMAA
Chairman of NOKIA

with Catherine Fredman

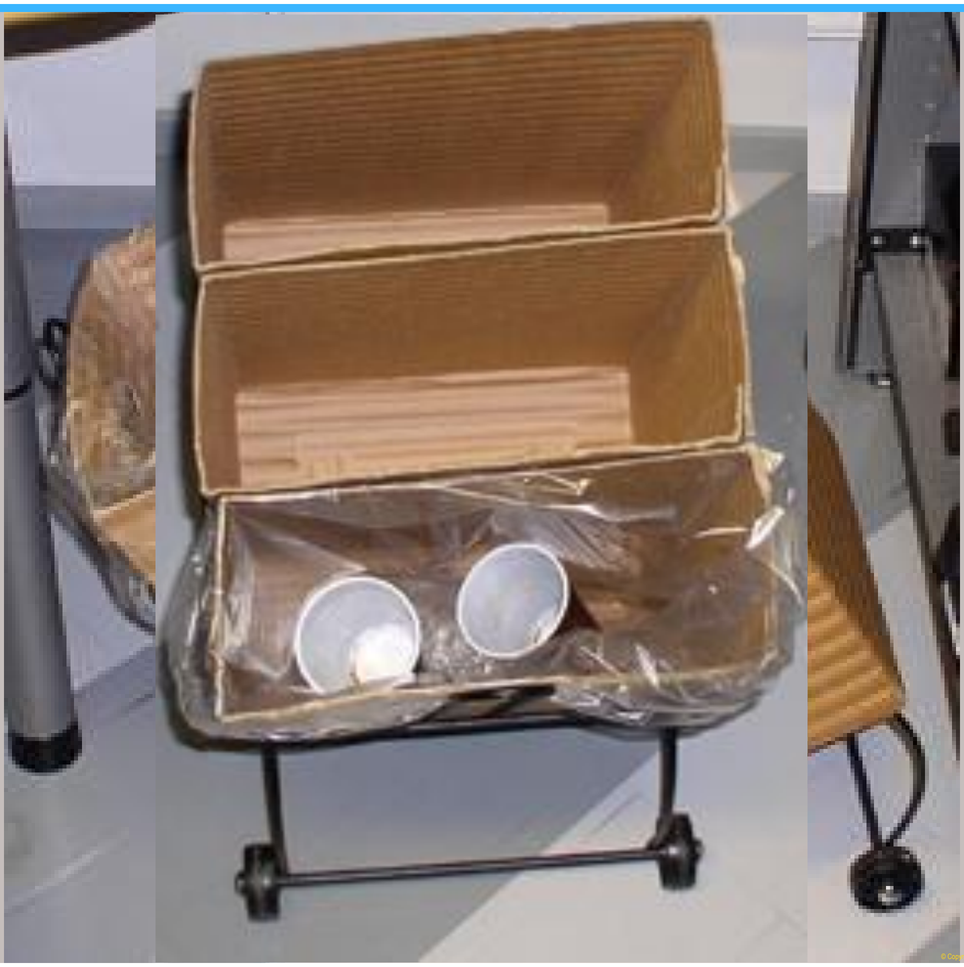
Mc Graw Hill

For Sale in India, Pakistan, Nepal, Bangladesh, Sri Lanka and Bhutan only

NTB = Trusted Catalog

https://anythingawesome.com

Cruiser Collector



NOKIA

https://anythingawesome.com

NOKIA CRUISER-COLLECTOR IN CAPITAL AREA

The Cruiser located by your desk for sorting waste has three sections:

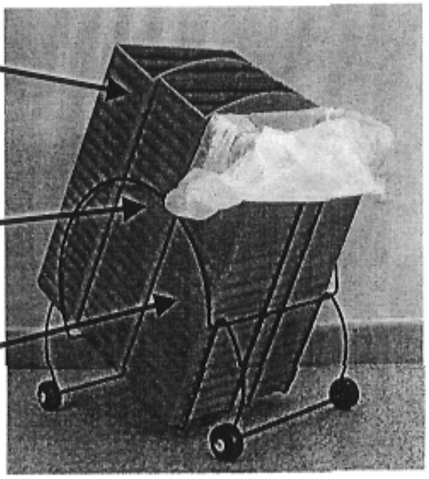
1. Office paper for shredding
- all white office paper, also the printer cover pages
- all white-based copy-paper
- all white-based printing paper
- all white memo slips

Take your confidential papers to the locked container in the office service point.

2. Recycling paper (newspaper, journals)
- newspaper and journals
- advertisement
- non-confidential coloured paper
- envelopes

Take your recycling papers to the non-locked container in the office service point.

3. Mixed waste
- rubbish
- plastic folders
- stickers
- Post-it slips
- wrap around the office paper reams

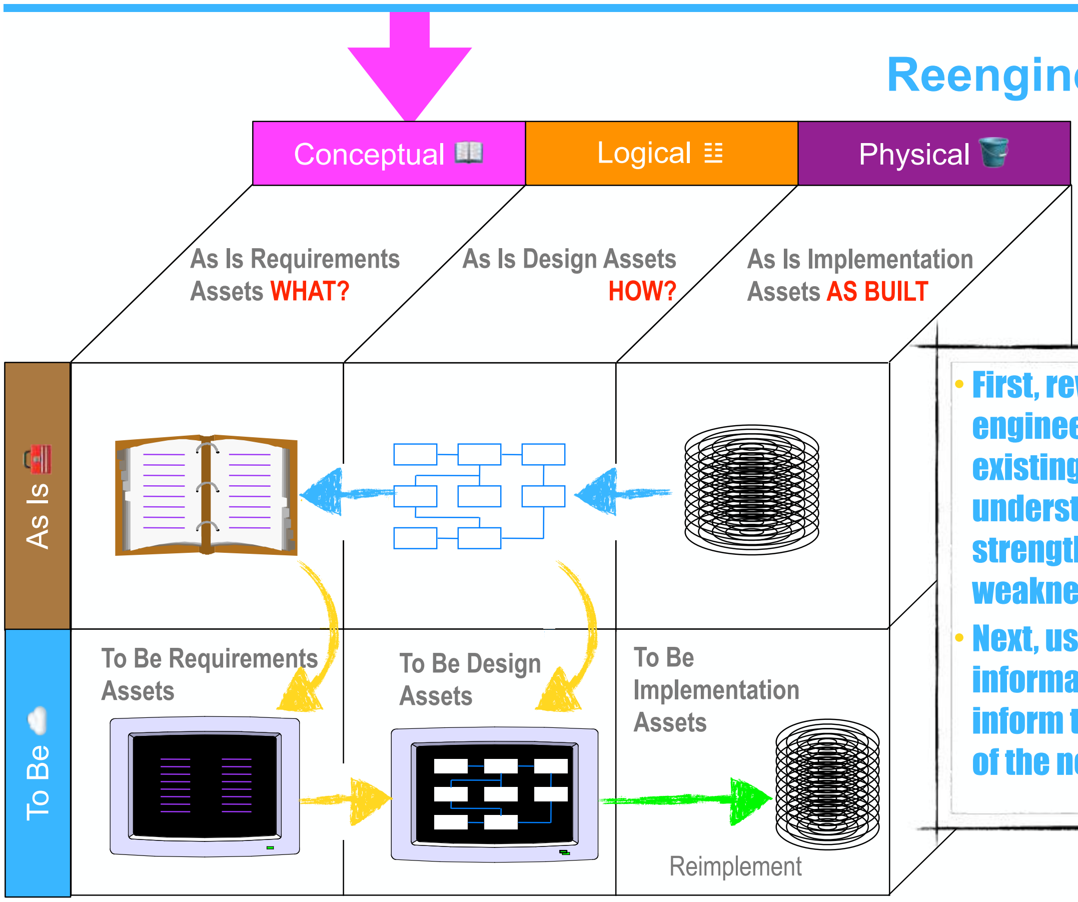


Sort your papers right for information security anc environment!

Take your biowaste to the container in the floor

https://anythingawesome.com

Reengineering



Conceptual

Logical

Physical

As Is Requirements Assets WHAT?

As Is Design Assets HOW?

As Is Implementation Assets AS BUILT

As Is

To Be

To Be Requirements Assets

To Be Design Assets

To Be Implementation Assets

Reimplement

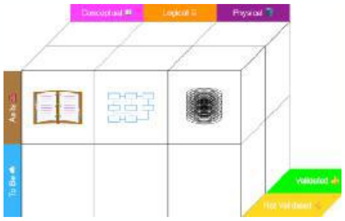
- First, reverse engineering the existing system to understand its strengths/ weaknesses
- Next, use this information to inform the design of the new system

https://anythingawesome.com

Program Overview

- Introduction to Modeling Data
  - Motivation
  - 3 primary data model types (+ plus two characteristics)
  - Reasons for each
  - Purposeful Modeling Basics (conversions, forward/reverse engineering)
- Conceptual
  - Motivation: Architectural tradeoffs
  - Strategy and conceptual data modeling
  - Glossary/Dictionary capabilities
- Logical
  - Motivation: Simplicity (Operational and Design)
  - Motivation towards standards
  - Business meets strategy
- Physical
  - Motivation: Required documentation and/or facts
  - Become the blueprints for physical construction of the solution
  - Blueprints are used for future maintenance of the solution
- Take Aways/References/Q&A

Conceptual
Versus
Logical
Versus
Physical
Data Modeling

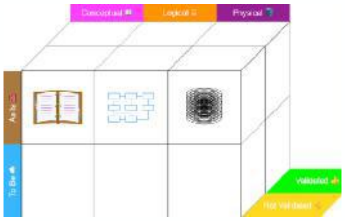


https://anythingawesome.com

Program Overview

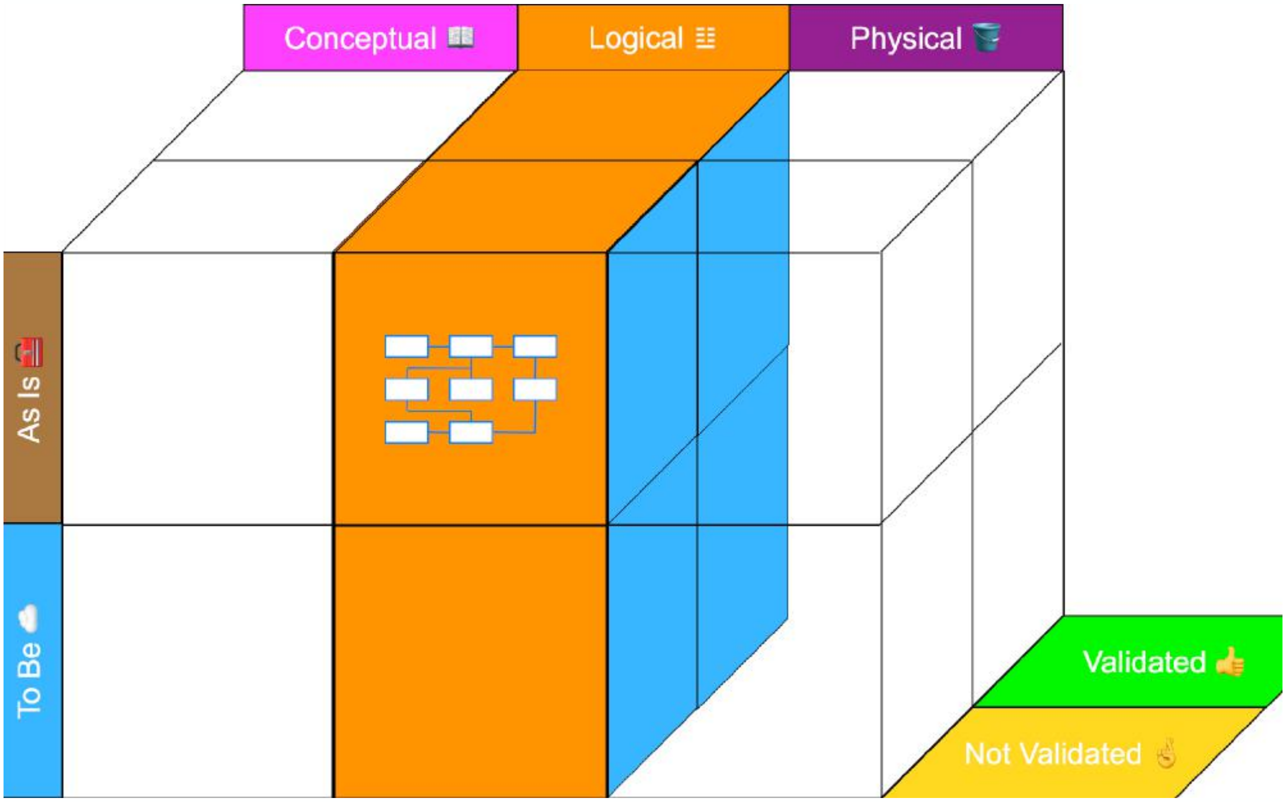
- Introduction to Modeling Data
  - Motivation
  - 3 primary data model types (+ plus two characteristics)
  - Reasons for each
  - Purposeful Modeling Basics (conversions, forward/reverse engineering)
- Conceptual
  - Motivation: Architectural tradeoffs
  - Strategy and conceptual data modeling
  - Glossary/Dictionary capabilities
- Logical
  - Motivation: Simplicity (Operational and Design)
  - Motivation towards standards
  - Business meets strategy
- Physical
  - Motivation: Required documentation and/or facts
  - Become the blueprints for physical construction of the solution
  - Blueprints are used for future maintenance of the solution
- Take Aways/References/Q&A

Conceptual
Versus
Logical
Versus
Physical
Data Modeling

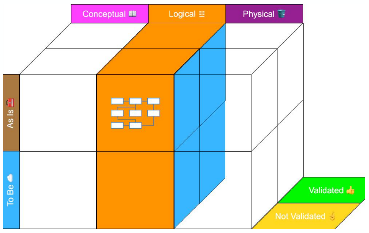


https://anythingawesome.com

Logical Data Modeling



Logical Data Modeling



Motivation

- Provide data specification information about effort
  - Size
  - Shape
  - Provenance
  - Functions
  - Down stream uses
- Free discussions from technological considerations that are separate from business objectives
- Document preliminary data designs satisfying business objectives
- Generate as much as possible

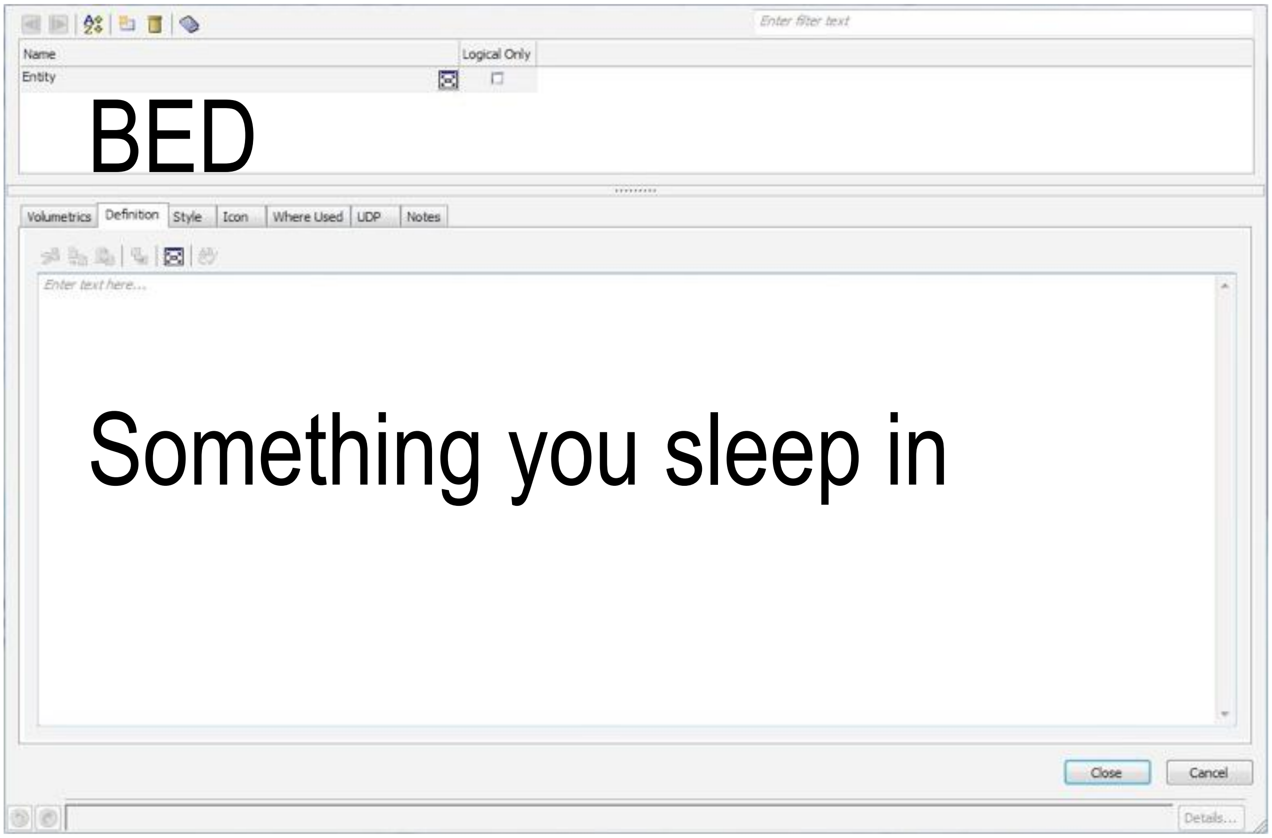
As Is Logical Data Models

- Challenge the conceptual model (if it exists)
- Explicitly incorporate relevant information from existing components

To Be Logical Data Models

- Serve as the organizing principle around which system data capabilities are built
- Facilitates common vocabulary among business and technical analysts

Standard Definition Reporting Does Not Provide Conceptual Context



BED

Something you sleep in

Purpose Statement Incorporates Motivations

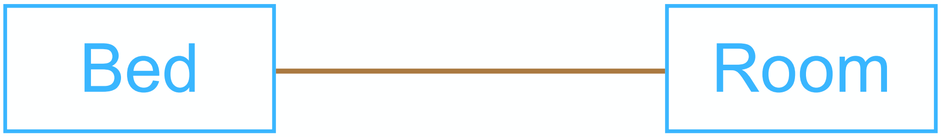
Entity:	BED
Data Asset Type:	Principal Data Entity
Purpose:	This is a substructure within the Room substructure of the Facility Location. It contains information about beds within rooms
Source:	Maintenance Manual for File and Table Data (Software Version 3.0, Release 3.1)
Attributes:	Bed.Description Bed.Status Bed.Sex.To.Be.Assigned Bed.Reserve.Reason
Associations:	>0-+ Room
Status:	DRAFT

A purpose statement describing

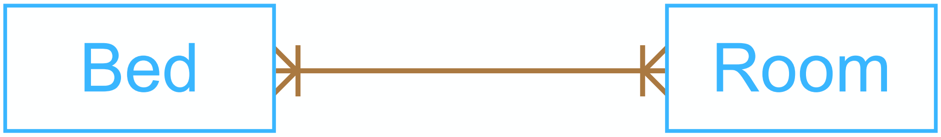
- Why the organization is maintaining information about this business concept;
- Sources of information about it;
- A partial list of the attributes or characteristics of the entity; and
- Associations with other data items(read as "One room contains zero or many beds.")

Q: What Is the Proper Relationship for These Entities?

Data Maps at the Entity Level ➜ Stored Facts

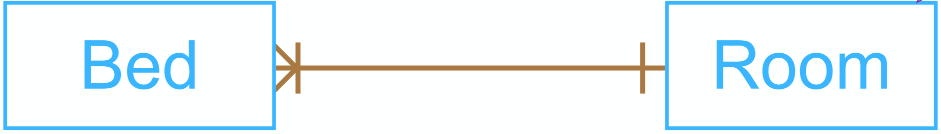


a BED is related to a ROOM



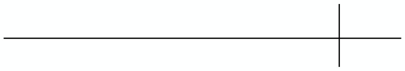
More precision:
many BEDS are related to many ROOMS

What if beds can be moved?

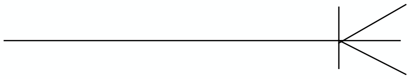


Better information:
many BEDS may be contained in each ROOM and each room may contain many beds

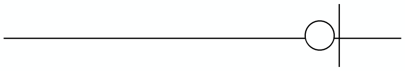
Possible Entity Relationship Cardinality Options



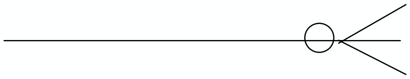
Exactly One (mandatory)



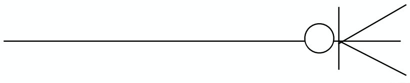
One or Many (mandatory)



Eventually One (optional)



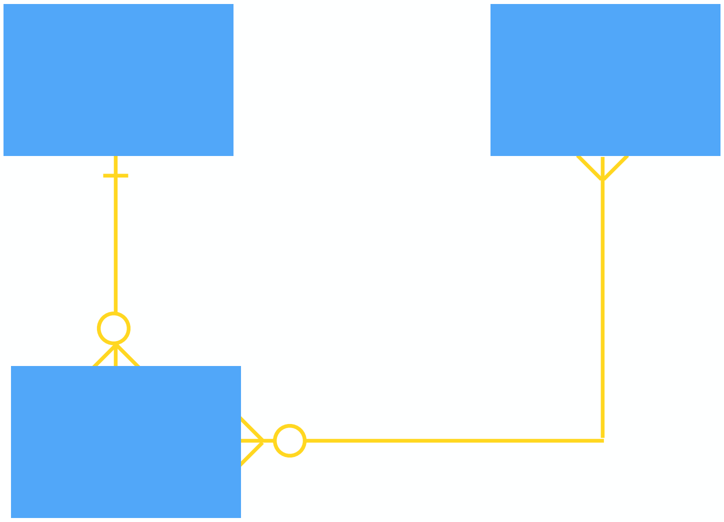
Zero, or Many (optional)



Eventually One or Many (optional)

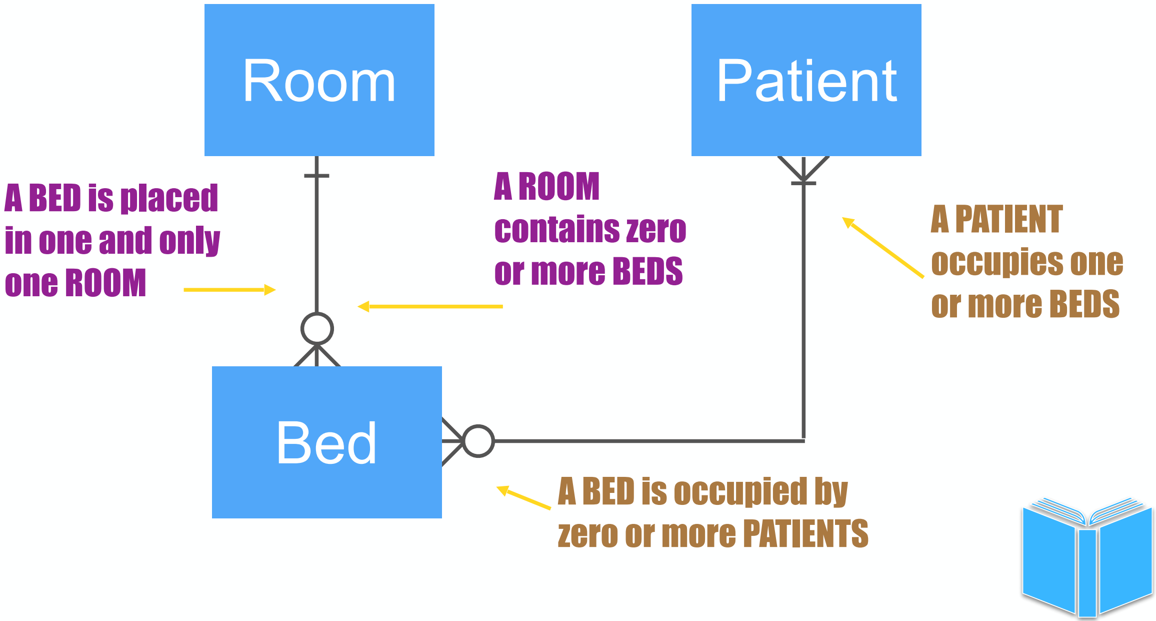
What Is a Relationship?

- Natural associations between two or more entities



Ordinality & Cardinality (Refinements on Relationships)

- Defines mandatory/optional relationships using minimum/ maximum occurrences from one entity to another



A BED is placed in one and only one ROOM

A ROOM contains zero or more BEDS

A PATIENT occupies one or more BEDS

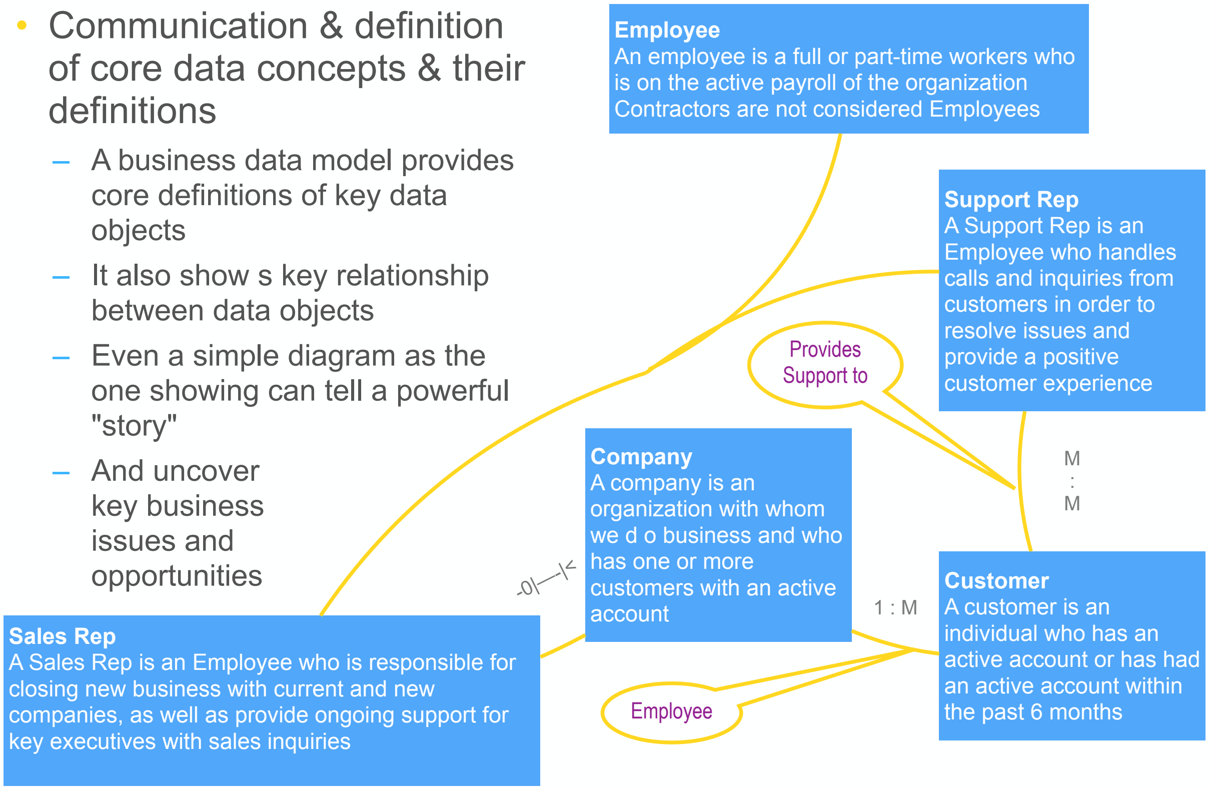
A BED is occupied by zero or more PATIENTS

Trusted Catalog

Example from Global Data Strategies, Ltd. http://globaldatastrategy.com

Business Data Model (Conceptual)

- Communication & definition of core data concepts & their definitions
  - A business data model provides core definitions of key data objects
  - It also show s key relationship between data objects
  - Even a simple diagram as the one showing can tell a powerful "story"
  - And uncover key business issues and opportunities



Employee
An employee is a full or part-time workers who is on the active payroll of the organization
Contractors are not considered Employees

Support Rep
A Support Rep is an Employee who handles calls and inquiries from customers in order to resolve issues and provide a positive customer experience

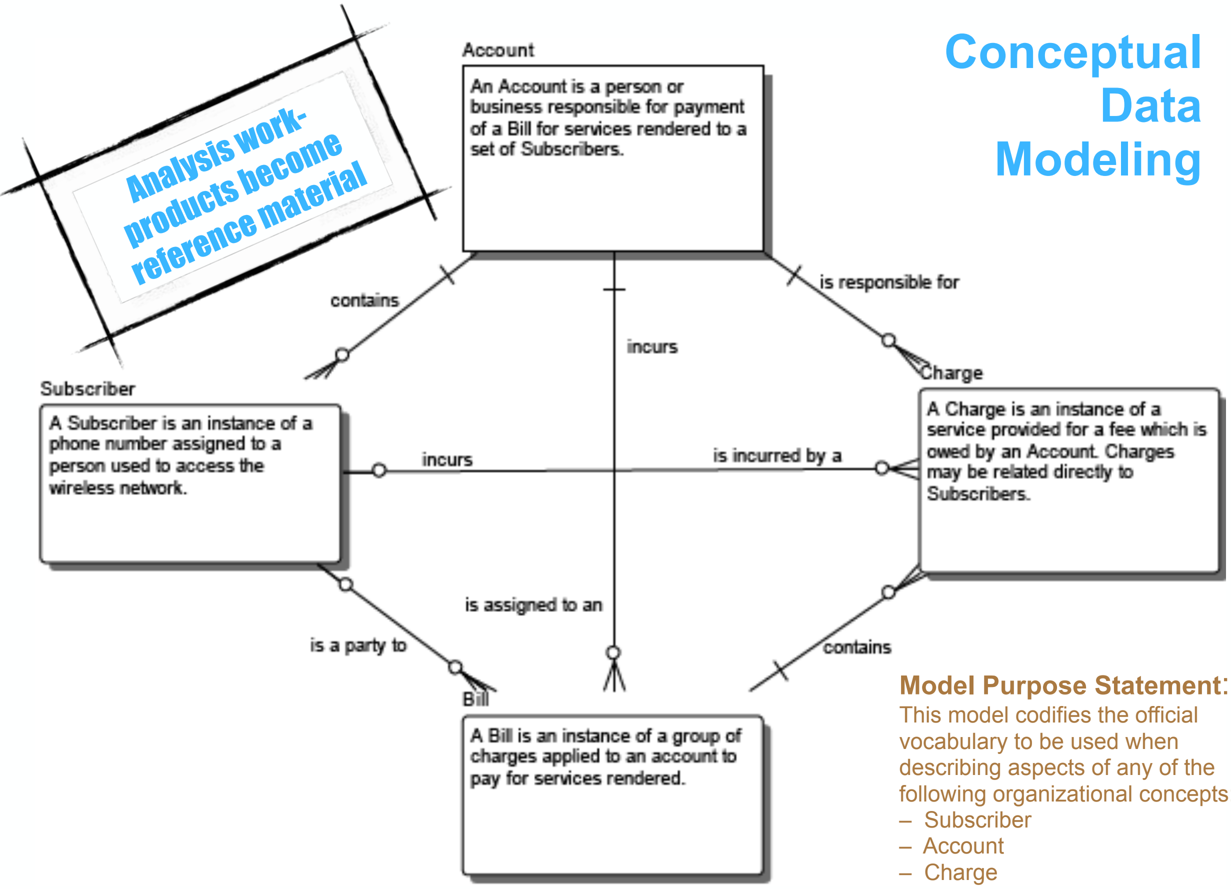
Company
A company is an organization with whom we d o business and who has one or more customers with an active account

Customer
A customer is an individual who has an active account or has had an active account within the past 6 months

Sales Rep
A Sales Rep is an Employee who is responsible for closing new business with current and new companies, as well as provide ongoing support for key executives with sales inquiries

Conceptual Data Modeling

Analysis work-products become reference material



Account

An Account is a person or business responsible for payment of a Bill for services rendered to a set of Subscribers.

Subscriber

A Subscriber is an instance of a phone number assigned to a person used to access the wireless network.

Charge

A Charge is an instance of a service provided for a fee which is owed by an Account. Charges may be related directly to Subscribers.

Bill

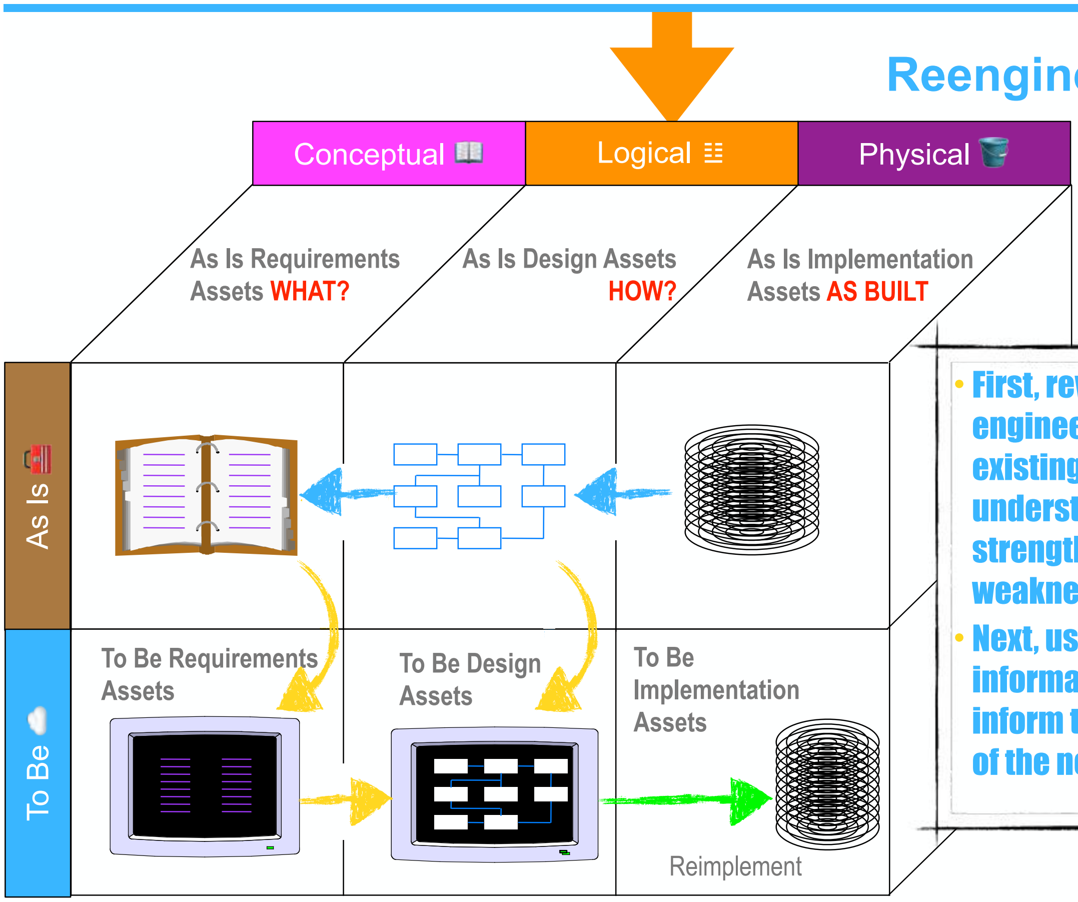
A Bill is an instance of a group of charges applied to an account to pay for services rendered.

Model Purpose Statement:
This model codifies the official vocabulary to be used when describing aspects of any of the following organizational concepts:

- Subscriber
- Account
- Charge
- Bill

from *The DAMA Guide to the Data Management Body of Knowledge* © 2009 by DAMA International

Reengineering

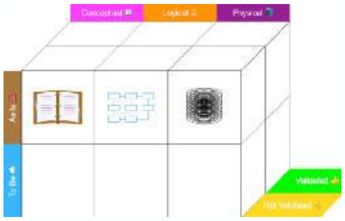

Conceptual
Logical
Physical
As Is Requirements Assets WHAT?
As Is Design Assets HOW?
As Is Implementation Assets AS BUILT
As Is
To Be
To Be Requirements Assets
To Be Design Assets
To Be Implementation Assets
Reimplement

- First, reverse engineering the existing system to understand its strengths/ weaknesses
- Next, use this information to inform the design of the new system

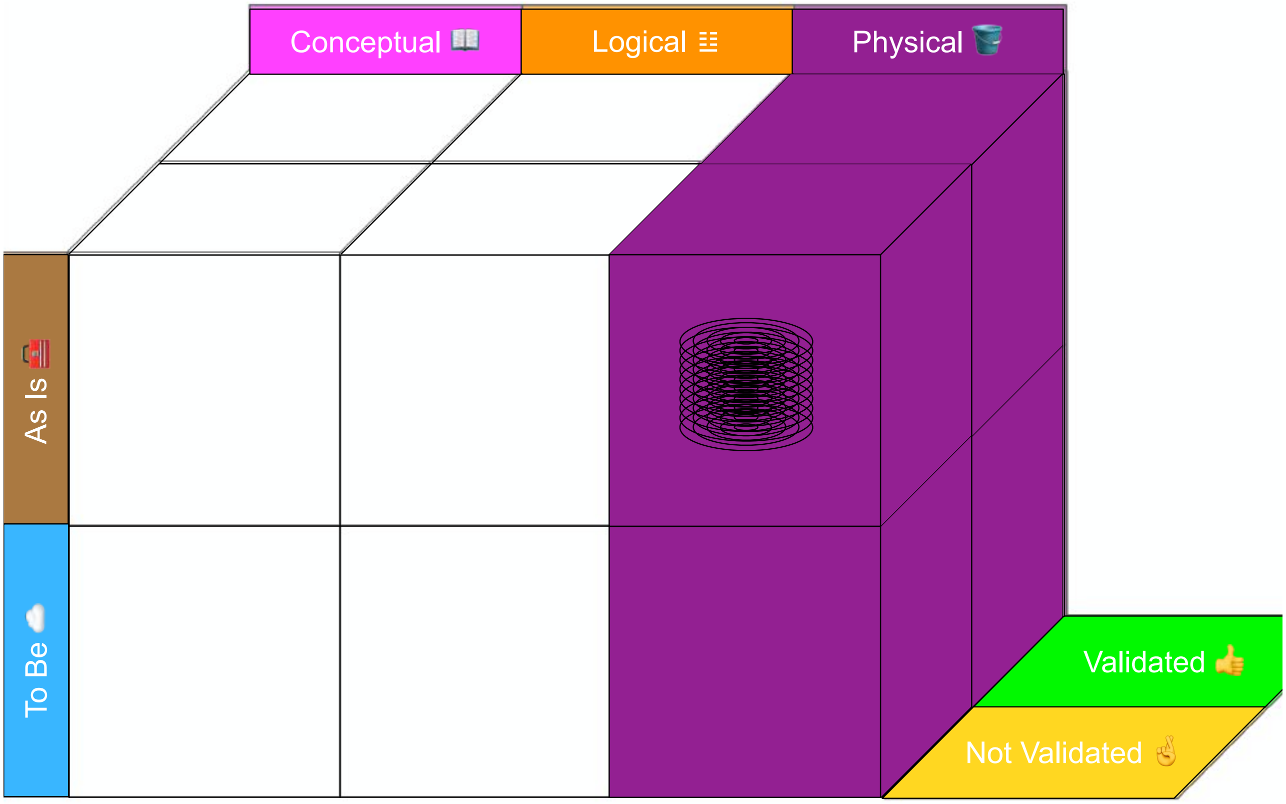
Program Overview

Conceptual
Versus
Logical
Versus
Physical
Data Modeling

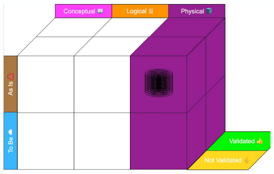
- Introduction to Modeling Data
  - Motivation
  - 3 primary data model types (+ plus two characteristics)
  - Reasons for each
  - Purposeful Modeling Basics (conversions, forward/reverse engineering)
- Conceptual
  - Motivation: Architectural tradeoffs
  - Strategy and conceptual data modeling
  - Glossary/Dictionary capabilities
- Logical
  - Motivation: Simplicity (Operational and Design)
  - Motivation towards standards
  - Business meets strategy
- Physical
  - Motivation: Required documentation and/or facts
  - Become the blueprints for physical construction of the solution
  - Blueprints are used for future maintenance of the solution
- Take Aways/References/Q&A



Physical Data Modeling



Physical Data Modeling



Motivation

- Documentation of specifications of production systems
  - Data flow diagrams
  - Entity-relationship diagrams
  - Dictionary/Glossary/Catalog
- Should exist if system is in production
  - Why would anyone hand craft DDL with today's tool capabilities?
- Must exist to create the system that is put into production
  - Become the blueprints for physical construction of the solution
  - Blueprints are used for future maintenance of the solution

DRAFT

As Is Physical Data Models (Exist too)

- This should be foundational system documentation
- Description required to access data 'in the system'
- Often can be reverse engineered, semi-automatically

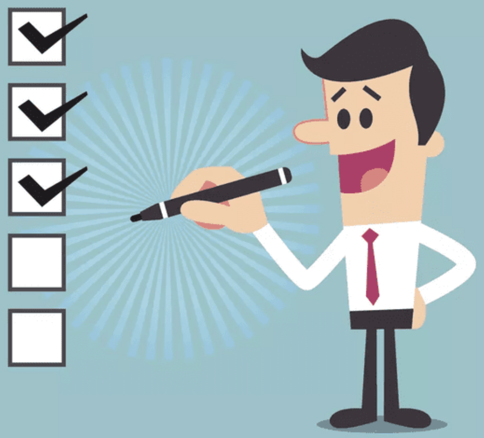
To Be Physical Data Models (Exist too)

- This is a specification of the data that can be accessed by the application
- Specification of current and future data elements to be maintained by application
- Often can be generated, semi-automatically

How Is Data Stored and Represented?

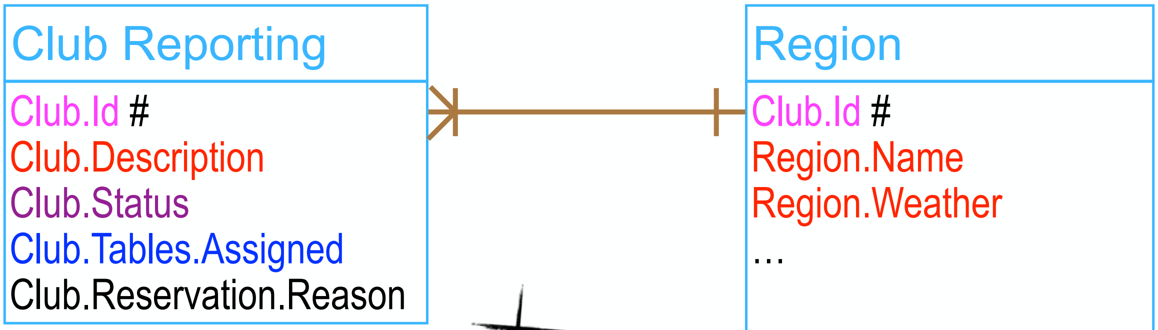
- Lists of organizational persons places things that need to be created read updated deleted archived

- These are called Attributes
  - Attributes are characteristics of "things"



Analyzing Data Attributes and Relationships

- Characteristics of CLUBS and REGIONS



Club Reporting

Club.Id #
Club.Description
Club.Status
Club.Tables.Assigned
Club.Reservation.Reason

Region

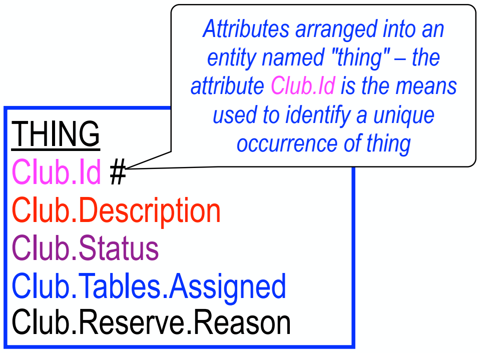
Club.Id #
Region.Name
Region.Weather
…

Each CLUB must be part of a Region

- What does the existence of this attribute tell us?
  - Clubs need to be identified (#) separately from one another
  - Club-specific information is likely maintained
  - Some concept (organization) exists above the 'club level'
  - ...

Data Modeling Uses

- An organization might decide to characterize the parts of a THING as:
  - Attributes: ID, description, status, Tables.Assigned, reserve.reason
- Decisions to manage information about each specific attribute has direct consequences
  - A decision to use the above data attributes permits the organization to determine if it has tables are available to be reserved
- Characteristics can be shared
  - All CLUBS may have a status
  - Many REASONS can be assigned to reservation (free text)
- Characteristics may be required to be unique
  - ID permits identification every CLUB as distinct for every other CLUB
  - Description is likely to be unique for each CLUB



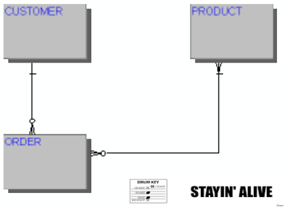
Attributes arranged into an entity named "thing" – the attribute Club.Id is the means used to identify a unique occurrence of thing

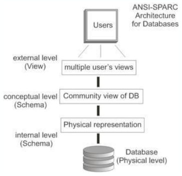
THING
Club.Id #
Club.Description
Club.Status
Club.Tables.Assigned
Club.Reserve.Reason

Model level variances are often among additions of keys and evolving definitions–hence the mandatory glossary!

Data Modeling Requirements

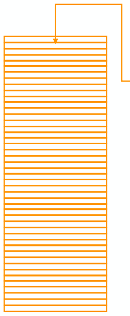
- The process of discovering, analyzing, and scoping data requirements
  - Understand what the data things are?
  - What do they do?
  - How do they interact?
- Representing/communicating requirements in a precise form called a data model
  - Maps of critical business assets
  - Compose and contain metadata essential to data consumers
  - Function as a kind of sheet music language
  - Metadata is essential to other business functions (definitions for governance, lineage for analytics, etc.)
- The process is iterative and may include conceptual, logical, and physical models
- Modeling is done to accomplish a goal!





https://anythingawesome.com

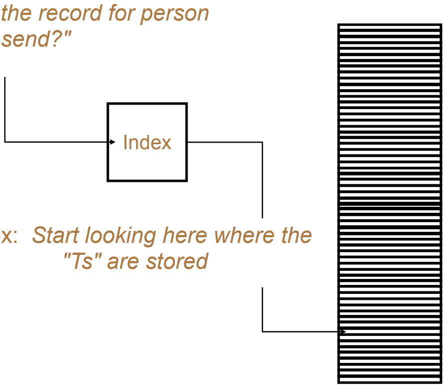
5 Basic Database Structures



Program: *Must start at the beginning and read each record when looking for person "Townsend?"*

Flat File: Records are typically sorted according to some criteria and must be searched from the beginning for each access

Program: *Where is the record for person "Townsend?"*

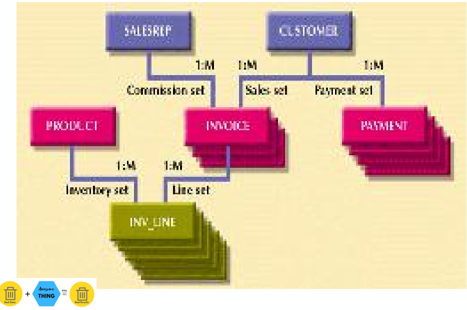


Index: *Start looking here where the "Ts" are stored*

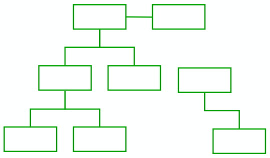
Indexed Sequential File: Built-in index permits location of records of persons with last names starting with "T"

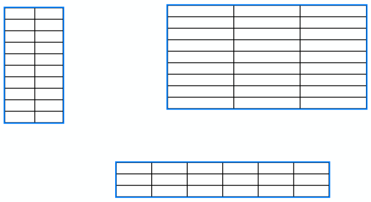
Associative Concept-oriented, Multi-dimensional, XML database, 3NF, Star schema, Data Vault, graph, LakeHouse

Network Database: Records are related to each other using arranged master records associated with multiple detail records using linked lists and pointers



Relational Database: Records are related to each other using relationships describable using relational algebra



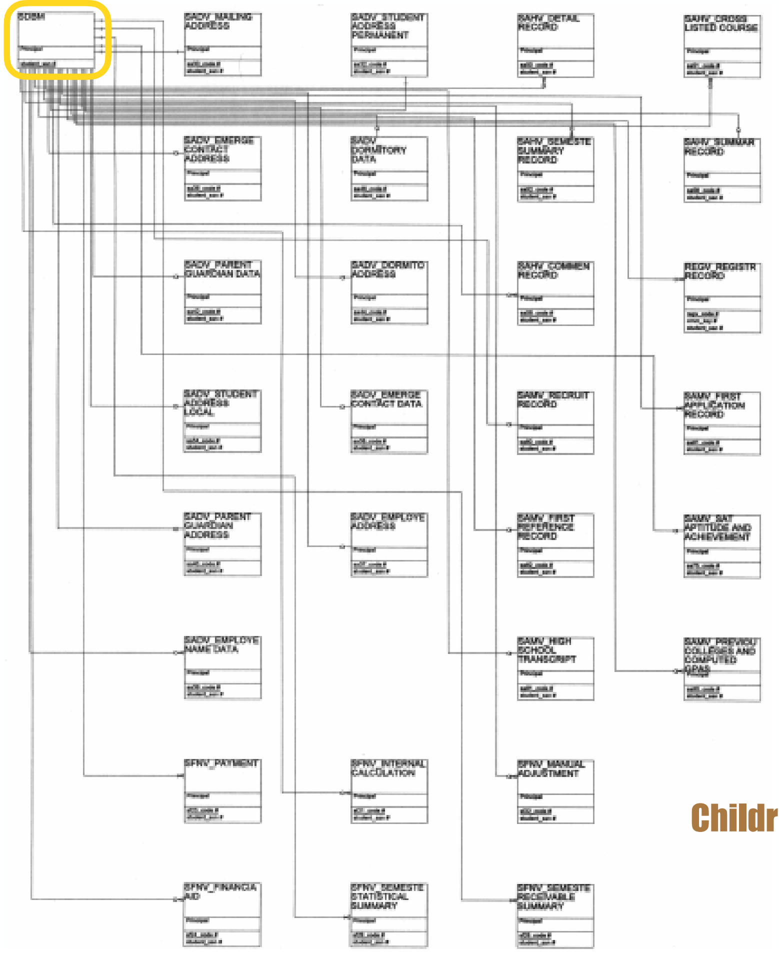


Hierarchical Database: Records are related to each other hierarchically using 'parent child' relationships

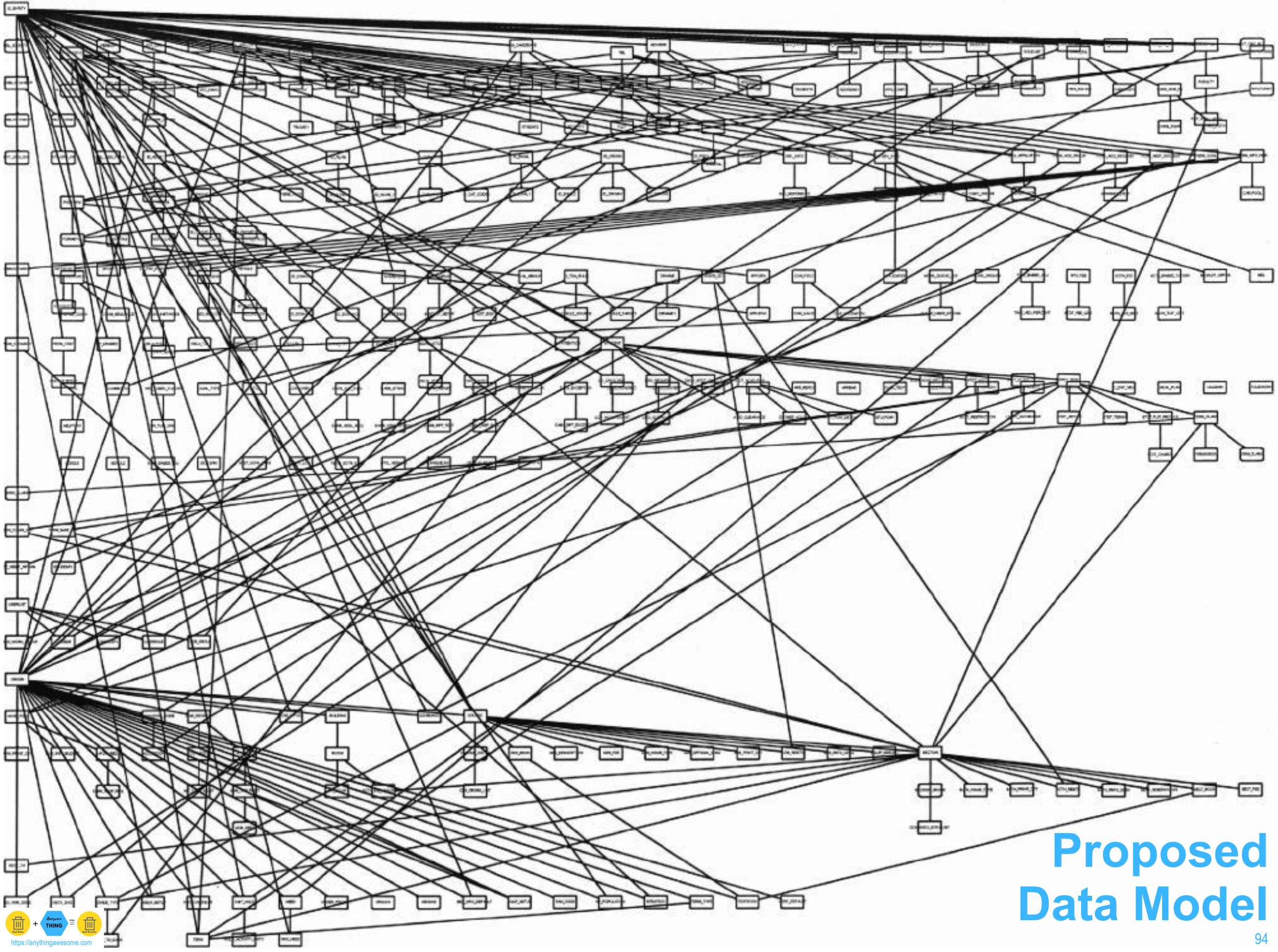
https://anythingawesome.com

Parent

Student Data Base Master

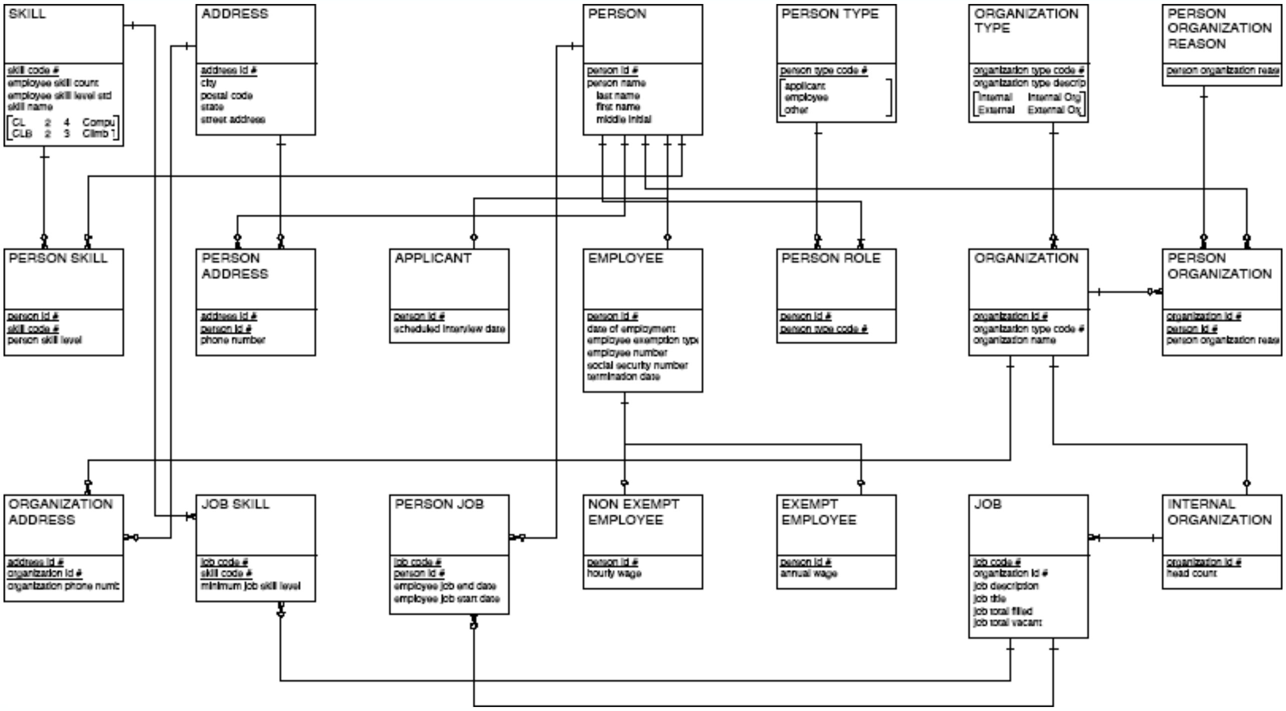

SDBM
Principal
SADV_MAILING ADDRESS
SADV_STUDENT ADDRESS PERMANENT
SAHV_DETAIL RECORD
SAHV_CROSS LISTED COURSE
SADV_EMERGE CONTACT ADDRESS
SADV DORMITORY DATA
SAHV_SEMESTE SUMMARY RECORD
SAHV_SUMMAR RECORD
SADV_PARENT GUARDIAN DATA
SADV_DORMITO ADDRESS
SAHV_COMMEN RECORD
REGV_REGISTR RECORD
SADV_STUDENT ADDRESS LOCAL
SADV_EMERGE CONTACT DATA
SAMV_RECRUIT RECORD
SAMV_FIRST APPLICATION RECORD
SADV_PARENT GUARDIAN ADDRESS
SADV_EMPLOYE ADDRESS
SAMV_FIRST REFERENCE RECORD
SAMV_SAT APTITUDE AND ACHIEVEMENT
SADV_EMPLOYE NAME DATA
SAMV_HIGH SCHOOL TRANSCRIPT
SAMV_PREVIOU COLLEGES AND COMPUTED GPAS
SFNV_PAYMENT
SFNV_INTERNAL CALCULATION
SFNV_MANUAL ADJUSTMENT
SFNV_FINANCIA AID
SFNV_SEMESTE STATISTICAL SUMMARY
SFNV_SEMESTE RECEIVABLE SUMMARY

Children



Proposed Data Model

HR Conceptual Model



SKILL
skill code #
employee skill count
employee skill level std
skill name
CL 2 4 Compu
CLB 2 3 Climb 1

ADDRESS
address id #
city
postal code
state
street address

PERSON
person id #
person name
last name
first name
middle initial

PERSON TYPE
person type code #
applicant
employee
other

ORGANIZATION TYPE
organization type code #
organization type descrip
Internal Internal Org
External External Or

PERSON ORGANIZATION REASON
person organization reas

PERSON SKILL
person id #
skill code #
person skill level

PERSON ADDRESS
address id #
person id #
phone number

APPLICANT
person id #
scheduled interview date

EMPLOYEE
person id #
date of employment
employee exemption typ
employee number
social security number
termination date

PERSON ROLE
person id #
person type code #

ORGANIZATION
organization id #
organization type code #
organization name

PERSON ORGANIZATION
organization id #
person id #
person organization reas

ORGANIZATION ADDRESS
address id #
organization id #
organization phone numb

JOB SKILL
job code #
skill code #
minimum job skill level

PERSON JOB
job code #
person id #
employee job end date
employee job start date

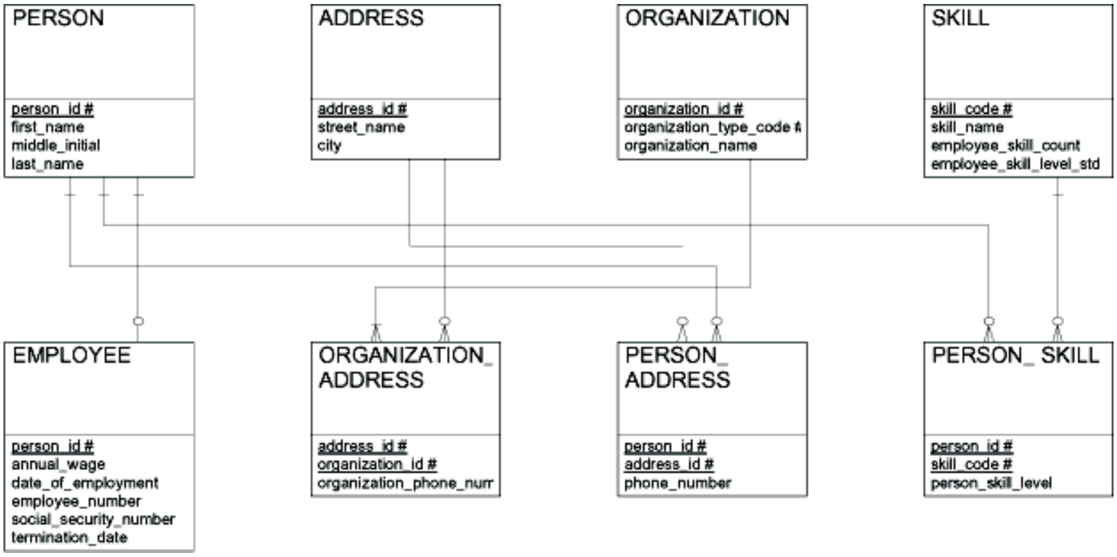
NON EXEMPT EMPLOYEE
person id #
hourly wage

EXEMPT EMPLOYEE
person id #
annual wage

JOB
job code #
organization id #
job description
job title
job total filled
job total vacant

INTERNAL ORGANIZATION
organization id #
head count

HR Logical Model



PERSON
person_id #
first_name
middle_initial
last_name

ADDRESS
address_id #
street_name
city

ORGANIZATION
organization_id #
organization_type_code #
organization_name

SKILL
skill_code #
skill_name
employee_skill_count
employee_skill_level_std

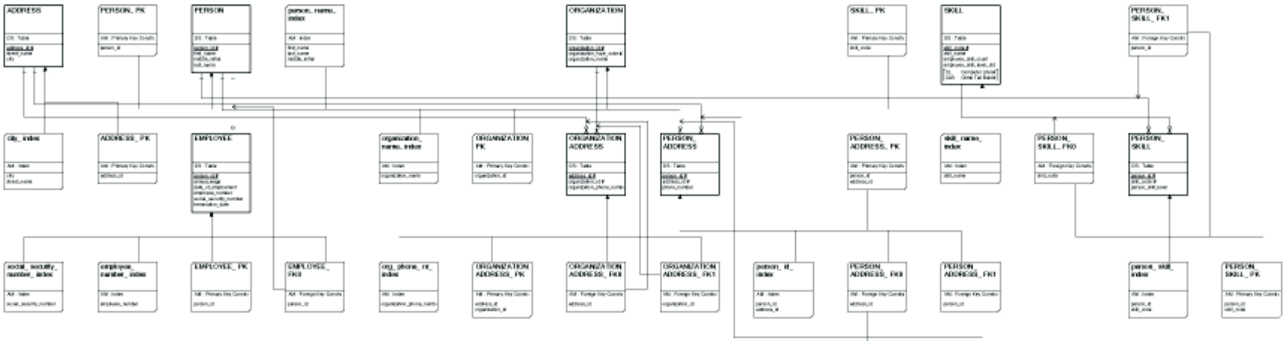
EMPLOYEE
person_id #
annual_wage
date_of_employment
employee_number
social_security_number
termination_date

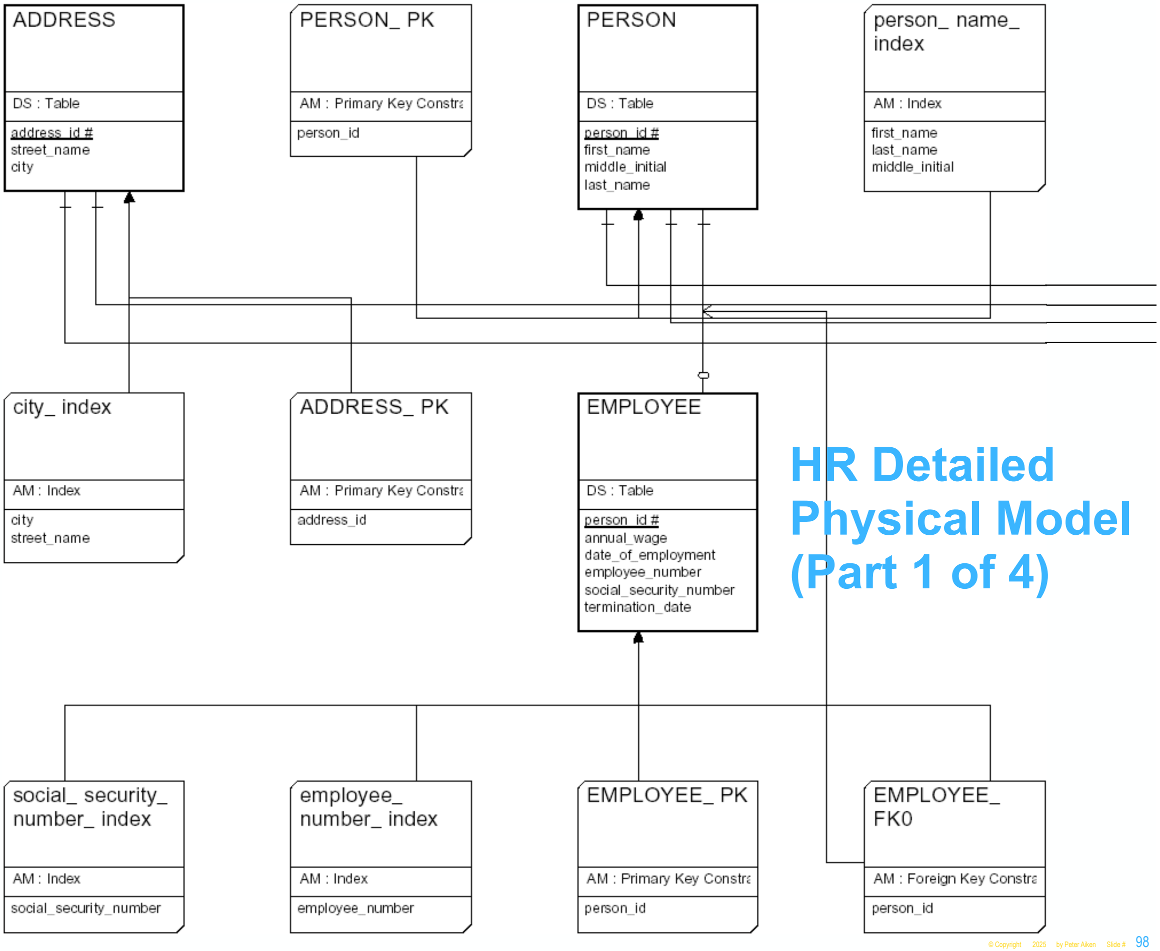
ORGANIZATION_ ADDRESS
address_id #
organization_id #
organization_phone_num

PERSON_ ADDRESS
person_id #
address_id #
phone_number

PERSON_ SKILL
person_id #
skill_code #
person_skill_level

HR Detailed Physical Model Overview





ADDRESS
DS : Table
address_id #
street_name
city

PERSON_ PK
AM : Primary Key Constra
person_id

PERSON
DS : Table
person_id #
first_name
middle_initial
last_name

person_ name_ index
AM : Index
first_name
last_name
middle_initial

city_ index
AM : Index
city
street_name

ADDRESS_ PK
AM : Primary Key Constra
address_id

EMPLOYEE
DS : Table
person_id #
annual_wage
date_of_employment
employee_number
social_security_number
termination_date

social_ security_ number_ index
AM : Index
social_security_number

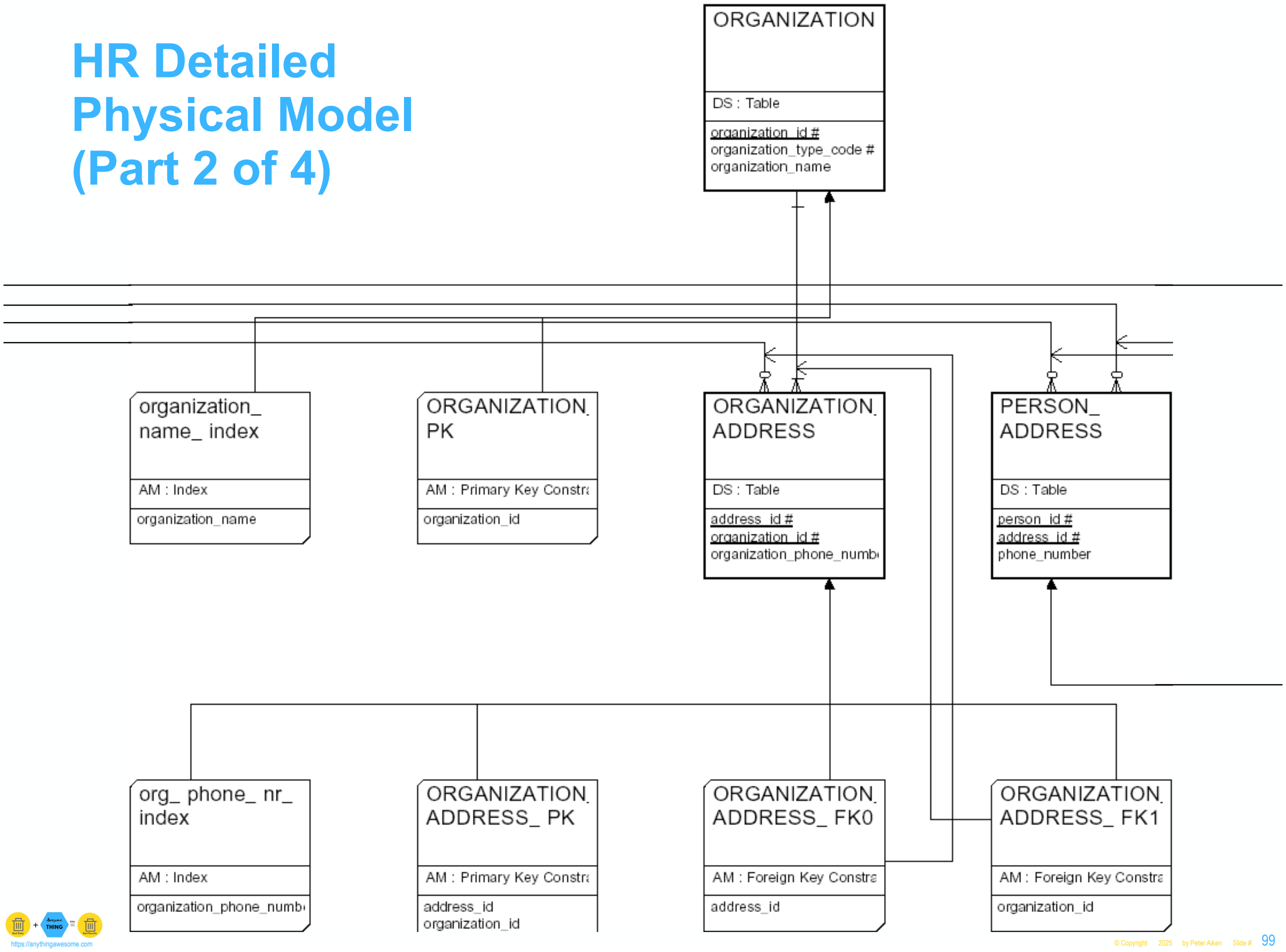
employee_ number_ index
AM : Index
employee_number

EMPLOYEE_ PK
AM : Primary Key Constra
person_id

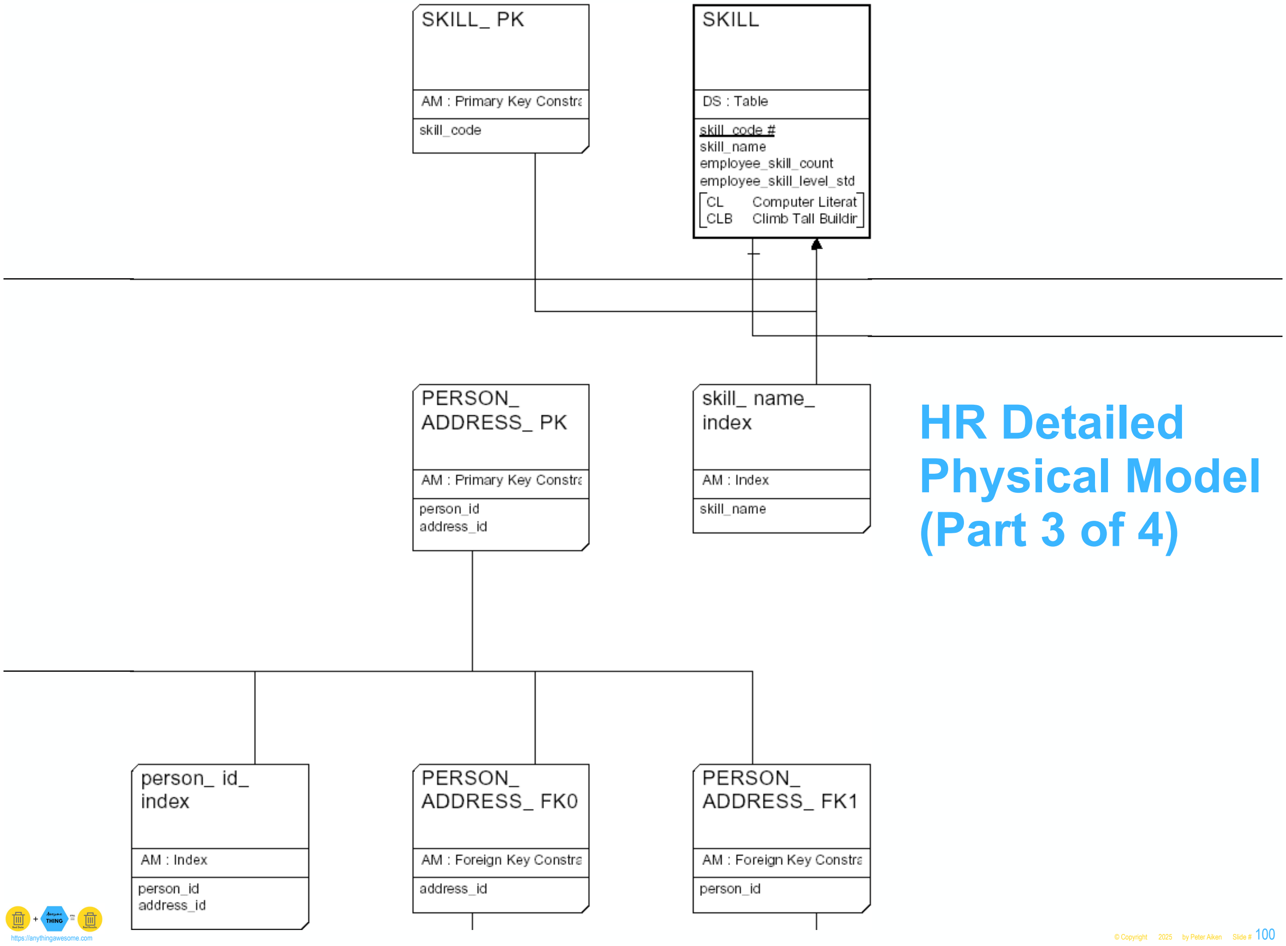
EMPLOYEE_ FK0
AM : Foreign Key Constra
person_id

HR Detailed Physical Model (Part 1 of 4)

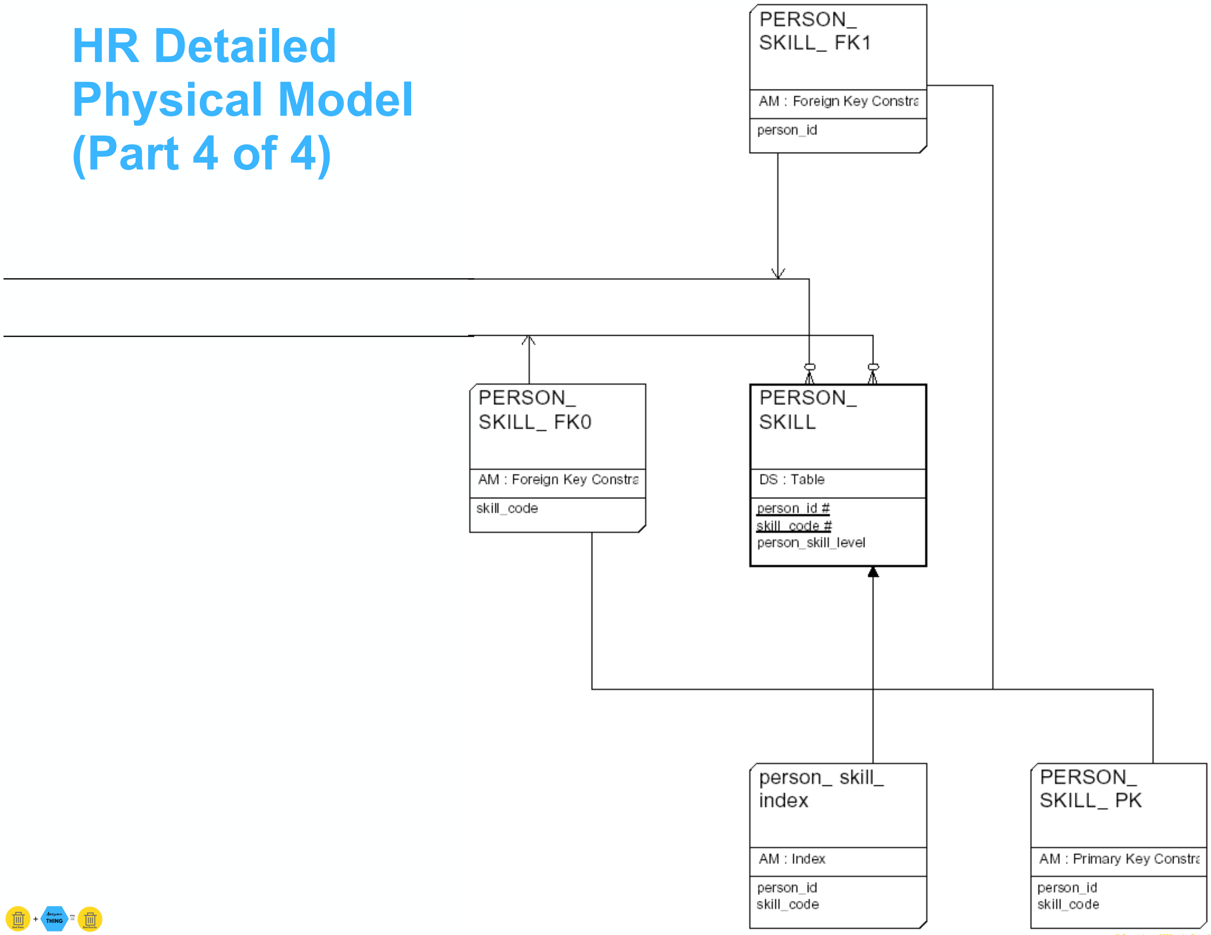
HR Detailed Physical Model (Part 2 of 4)



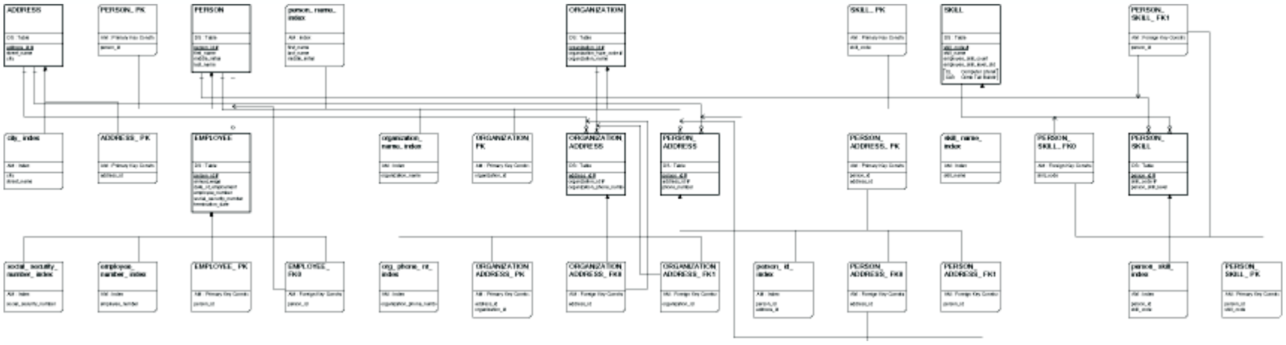
HR Detailed Physical Model (Part 3 of 4)



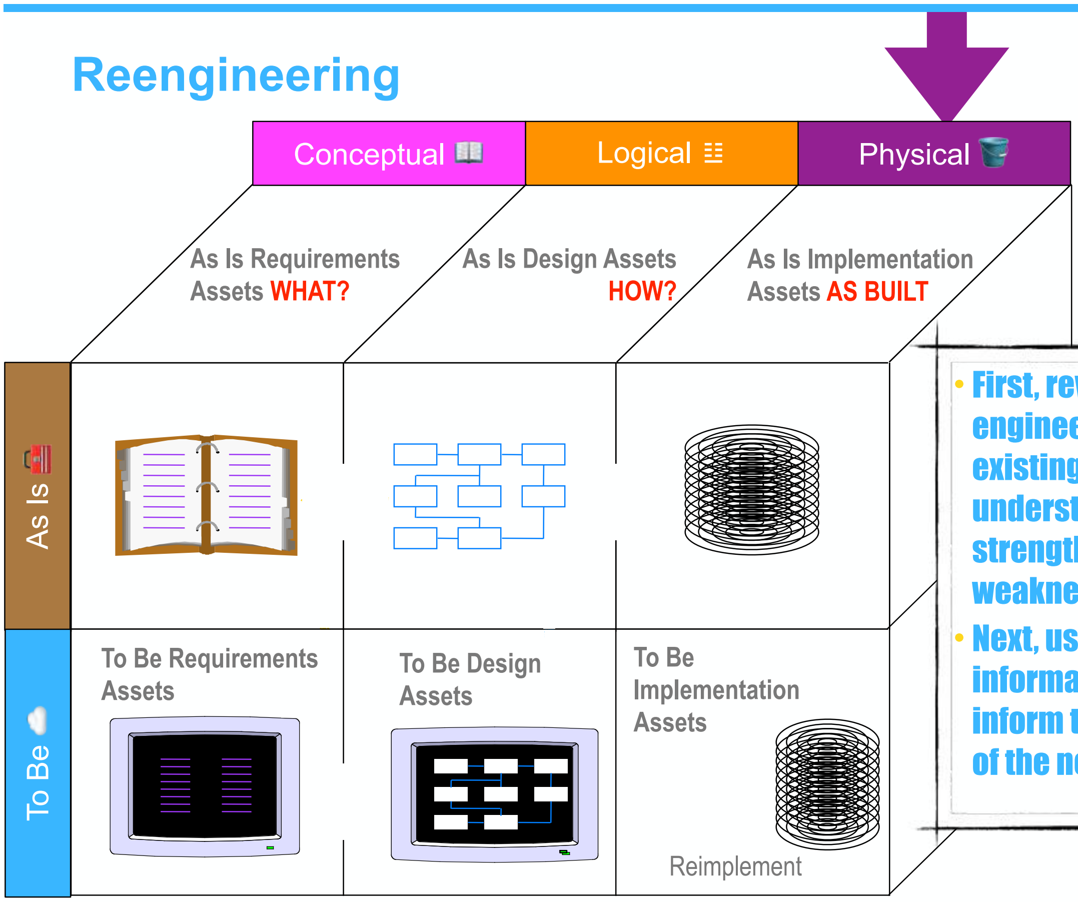
HR Detailed Physical Model (Part 4 of 4)



HR Detailed Physical Model Overview



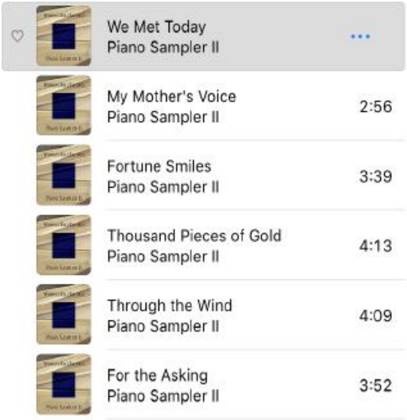
Reengineering


Conceptual
Logical
Physical
As Is Requirements Assets WHAT?
As Is Design Assets HOW?
As Is Implementation Assets AS BUILT
As Is
To Be
To Be Requirements Assets
To Be Design Assets
To Be Implementation Assets
Reimplement

- First, reverse engineering the existing system to understand its strengths/ weaknesses
- Next, use this information to inform the design of the new system

There Are Correct Ways To Organize Data

- All involve data modeling
- Optimization can be done for:
  - Flexibility
  - Adaptability
  - Retrievability
  - Risk reduction
  - ...
- Techniques include:
  - Data integrity
  - Smart codes bad/dumb codes good
  - Architecture (table joins)
  - ...



We Met Today
Piano Sampler II

My Mother's Voice
Piano Sampler II
2:56

Fortune Smiles
Piano Sampler II
3:39

Thousand Pieces of Gold
Piano Sampler II
4:13

Through the Wind
Piano Sampler II
4:09

For the Asking
Piano Sampler II
3:52

Don’t Tell Them That You Are Modeling!

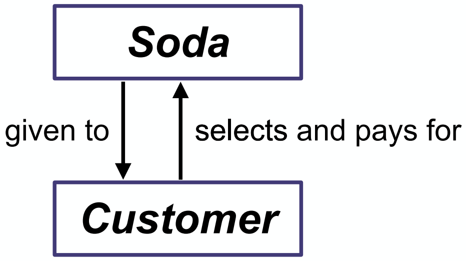
Just write some stuff down

Then arrange it

Then make some appropriate connections between your objects

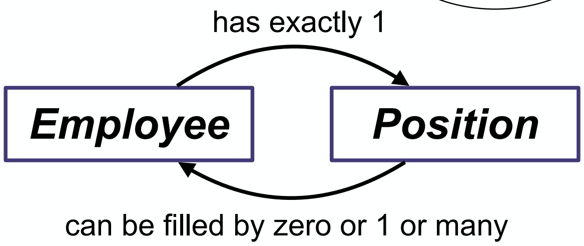
Keep Focused on the Data Model's Purpose

- The reason we are locked in this room is to:
  - Mission: *Understand formal relationship between soda and customer*
    - Outcome: Walk out the door with an as is physical and logical data model this relationship
  - Mission: *Understand the characteristics that differ between our hospital beds*
    - Outcome: We will walk out the door when we identify the top three characteristics that represent the brand with a logical data model
  - Mission: *Could our systems handle the following business rule **tomorrow**?*
  - "Is job-sharing permitted?"
    - Outcomes: Confirm that it is possible to staff a position with multiple employees effective tomorrow - need conceptual model for board presentation



Bed	
Entity:	BED
Purpose:	This is a substructure within the room substructure of the facility location. It contains information about beds within rooms.
Attributes:	Bed.Description Bed.Status Bed.Sex.To.Be.Assigned Bed.Reserve.Reason
Associations:	>0-+ Room
Status:	Validated

How does our perspective change: *the primary means of tracking a patient*



Inspired by: Karen Lopez http://www.information-management.com/newsletters/enterprise_architecture_data_model_ERP_BI-10020246-1.html?pg=2

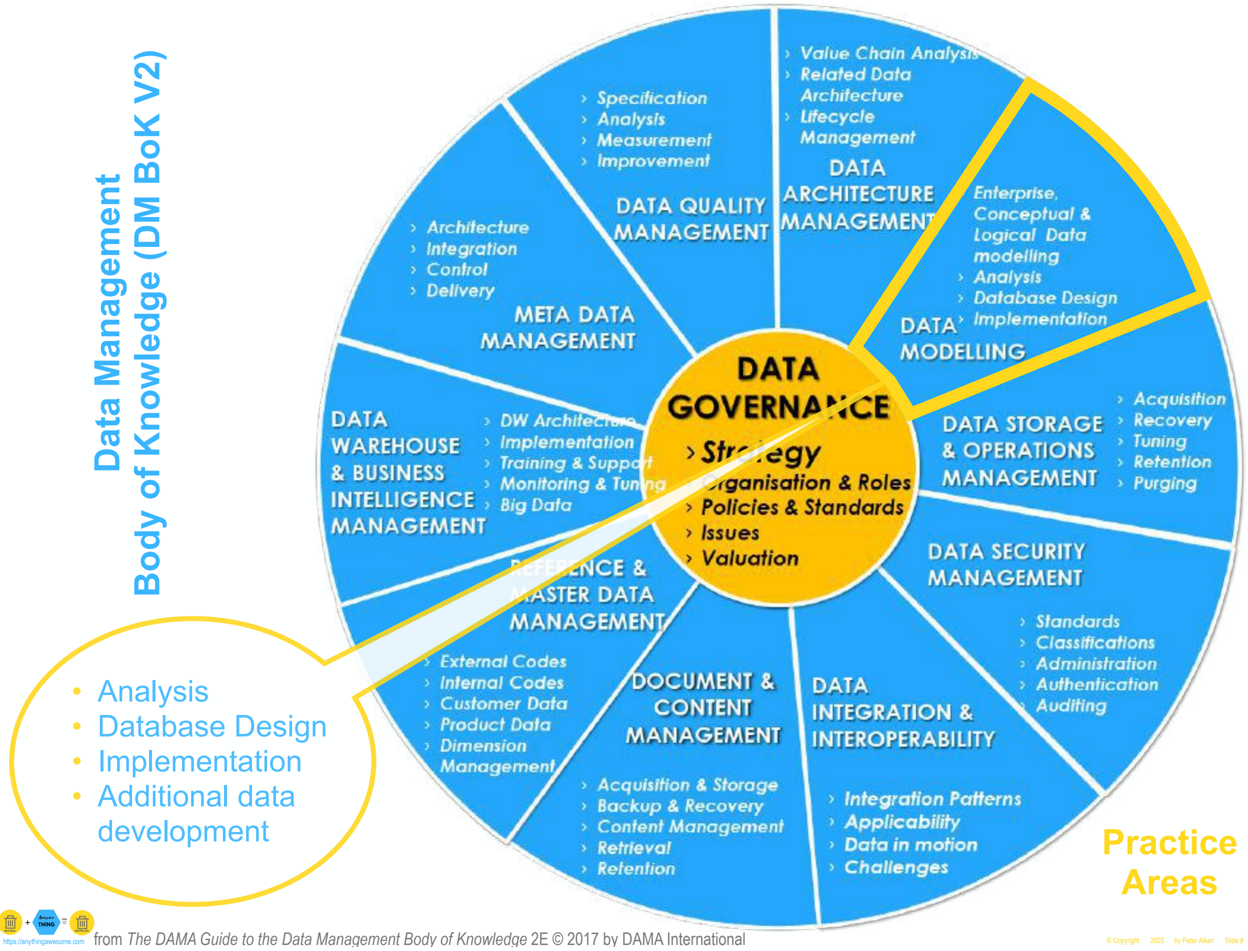
Data Modeling for Business Value

- Goal must be shared IT/business understanding
  - No disagreements/refinements means insufficient communication
- Data sharing/exchange is automated and dependent on successful engineering/architecture
  - Requires a sound foundation of data modeling basics (the essence) on which to build technologies
- Incorporate motivation (purpose statements) in all modeling
  - Modeling is a problem defining as well as a problem solving activity
- Modeling characteristics evolve during the analysis
  - Different modeling challenges for different problems
  - Use of modeling is more important than use of a specific method
  - Models must be maintained as living documents
  - Models need to be available in an easily searchable manner
- Utility is paramount
  - Adding color and diagramming objects customizes models and allows for a more engaging and enjoyable interaction
- Value is derived from
  - Improving organizational data
  - Improving the way people use data
  - Improving peoples use of data to support strategy

This can only be accomplished incrementally using an iterative, approach focusing on one aspect at a time and applying formal transformation methods

To Learn More

Data Management
Body of Knowledge (DM BoK V2)


DATA GOVERNANCE
- Strategy
- Organisation & Roles
- Policies & Standards
- Issues
- Valuation

DATA ARCHITECTURE MANAGEMENT
- Value Chain Analysis
- Related Data Architecture
- Lifecycle Management

DATA MODELLING
- Enterprise, Conceptual & Logical Data modelling
- Analysis
- Database Design
- Implementation

DATA STORAGE & OPERATIONS MANAGEMENT
- Acquisition
- Recovery
- Tuning
- Retention
- Purging

DATA SECURITY MANAGEMENT
- Standards
- Classifications
- Administration
- Authentication
- Auditing

DATA INTEGRATION & INTEROPERABILITY
- Integration Patterns
- Applicability
- Data in motion
- Challenges

DOCUMENT & CONTENT MANAGEMENT
- Acquisition & Storage
- Backup & Recovery
- Content Management
- Retrieval
- Retention

REFERENCE & MASTER DATA MANAGEMENT
- External Codes
- Internal Codes
- Customer Data
- Product Data
- Dimension Management

DATA WAREHOUSE & BUSINESS INTELLIGENCE MANAGEMENT
- DW Architecture
- Implementation
- Training & Support
- Monitoring & Tuning
- Big Data

META DATA MANAGEMENT
- Architecture
- Integration
- Control
- Delivery

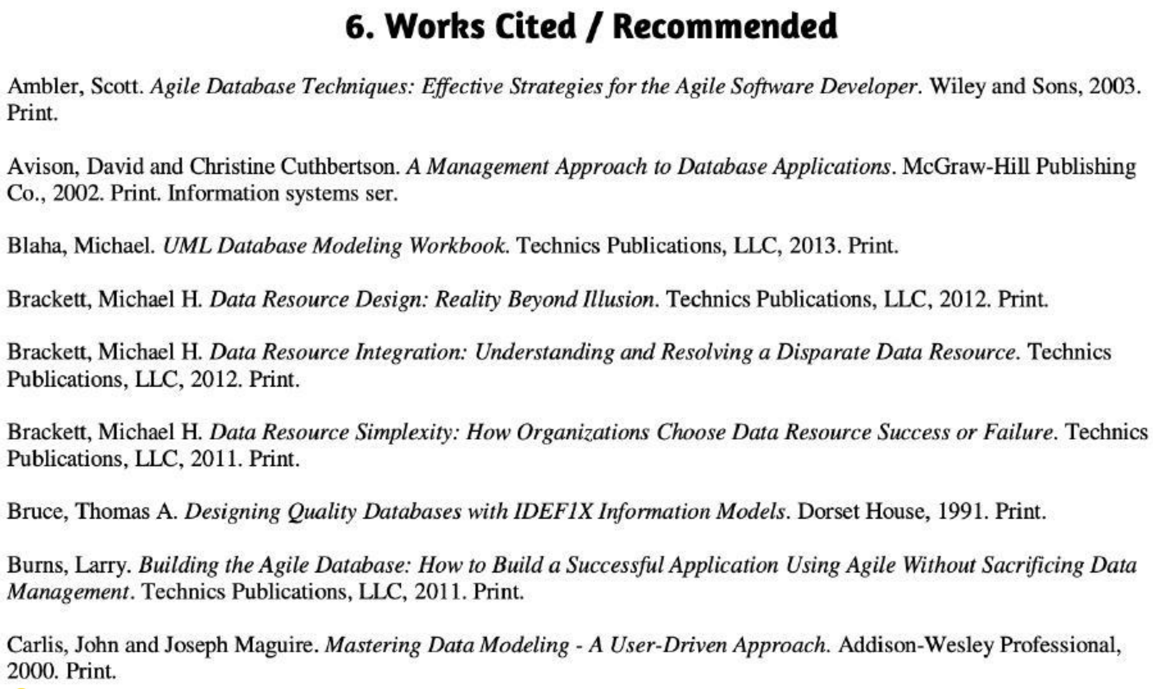
DATA QUALITY MANAGEMENT
- Specification
- Analysis
- Measurement
- Improvement

- Analysis
- Database Design
- Implementation
- Additional data development

Practice Areas

from *The DAMA Guide to the Data Management Body of Knowledge* 2E © 2017 by DAMA International

Works Cited


6. Works Cited / Recommended

Ambler, Scott. *Agile Database Techniques: Effective Strategies for the Agile Software Developer*. Wiley and Sons, 2003. Print.

Avison, David and Christine Cuthbertson. *A Management Approach to Database Applications*. McGraw-Hill Publishing Co., 2002. Print. Information systems ser.

Blaha, Michael. *UML Database Modeling Workbook*. Technics Publications, LLC, 2013. Print.

Brackett, Michael H. *Data Resource Design: Reality Beyond Illusion*. Technics Publications, LLC, 2012. Print.

Brackett, Michael H. *Data Resource Integration: Understanding and Resolving a Disparate Data Resource*. Technics Publications, LLC, 2012. Print.

Brackett, Michael H. *Data Resource Simplexity: How Organizations Choose Data Resource Success or Failure*. Technics Publications, LLC, 2011. Print.

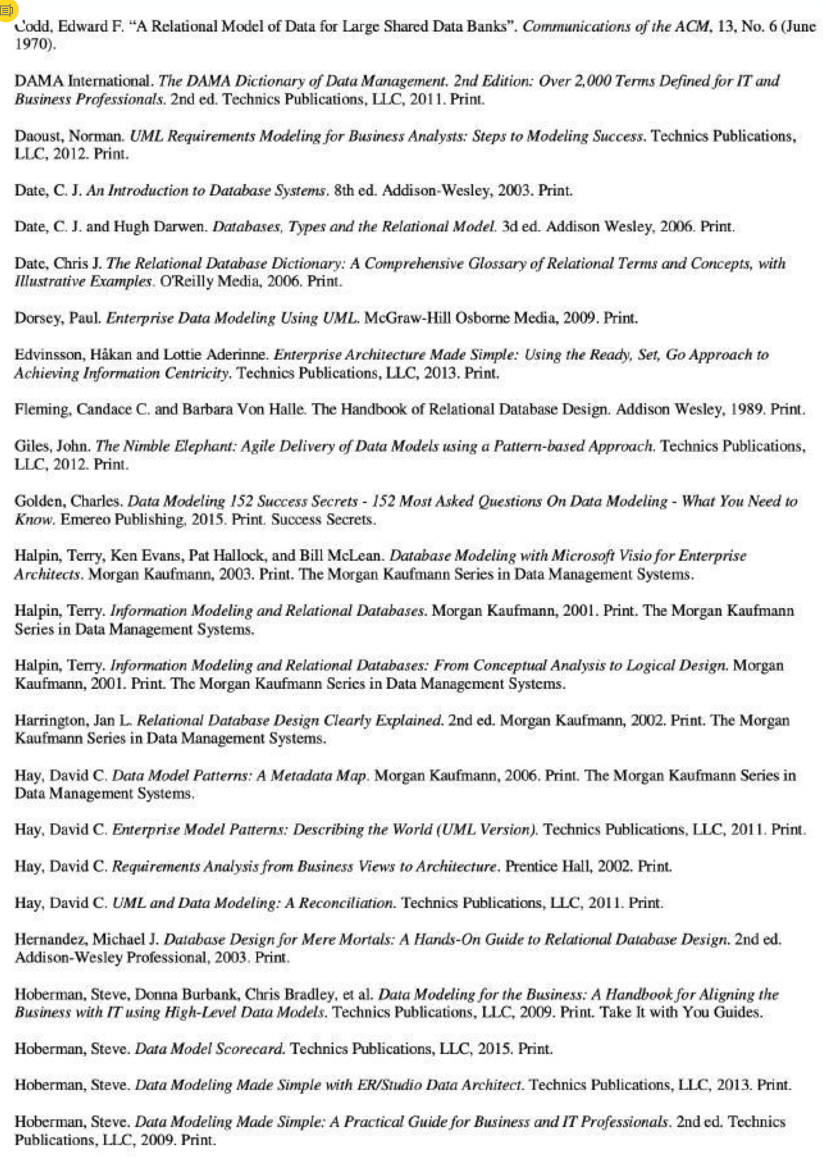
Bruce, Thomas A. *Designing Quality Databases with IDEF1X Information Models*. Dorset House, 1991. Print.

Burns, Larry. *Building the Agile Database: How to Build a Successful Application Using Agile Without Sacrificing Data Management*. Technics Publications, LLC, 2011. Print.

Carlis, John and Joseph Maguire. *Mastering Data Modeling - A User-Driven Approach.* Addison-Wesley Professional, 2000. Print.

from *The DAMA Guide to the Data Management Body of Knowledge* 2E © 2017 by DAMA International

Works Cited



Codd, Edward F. "A Relational Model of Data for Large Shared Data Banks". *Communications of the ACM*, 13, No. 6 (June 1970).

DAMA International. *The DAMA Dictionary of Data Management. 2nd Edition: Over 2,000 Terms Defined for IT and Business Professionals*. 2nd ed. Technics Publications, LLC, 2011. Print.

Daoust, Norman. *UML Requirements Modeling for Business Analysts: Steps to Modeling Success*. Technics Publications, LLC, 2012. Print.

Date, C. J. *An Introduction to Database Systems*. 8th ed. Addison-Wesley, 2003. Print.

Date, C. J. and Hugh Darwen. *Databases, Types and the Relational Model*. 3d ed. Addison Wesley, 2006. Print.

Date, Chris J. *The Relational Database Dictionary: A Comprehensive Glossary of Relational Terms and Concepts, with Illustrative Examples*. O'Reilly Media, 2006. Print.

Dorsey, Paul. *Enterprise Data Modeling Using UML*. McGraw-Hill Osborne Media, 2009. Print.

Edvinsson, Håkan and Lottie Aderinne. *Enterprise Architecture Made Simple: Using the Ready, Set, Go Approach to Achieving Information Centricity*. Technics Publications, LLC, 2013. Print.

Fleming, Candace C. and Barbara Von Halle. The Handbook of Relational Database Design. Addison Wesley, 1989. Print.

Giles, John. *The Nimble Elephant: Agile Delivery of Data Models using a Pattern-based Approach*. Technics Publications, LLC, 2012. Print.

Golden, Charles. *Data Modeling 152 Success Secrets - 152 Most Asked Questions On Data Modeling - What You Need to Know*. Emereo Publishing, 2015. Print. Success Secrets.

Halpin, Terry, Ken Evans, Pat Hallock, and Bill McLean. *Database Modeling with Microsoft Visio for Enterprise Architects*. Morgan Kaufmann, 2003. Print. The Morgan Kaufmann Series in Data Management Systems.

Halpin, Terry. *Information Modeling and Relational Databases*. Morgan Kaufmann, 2001. Print. The Morgan Kaufmann Series in Data Management Systems.

Halpin, Terry. *Information Modeling and Relational Databases: From Conceptual Analysis to Logical Design*. Morgan Kaufmann, 2001. Print. The Morgan Kaufmann Series in Data Management Systems.

Harrington, Jan L. *Relational Database Design Clearly Explained*. 2nd ed. Morgan Kaufmann, 2002. Print. The Morgan Kaufmann Series in Data Management Systems.

Hay, David C. *Data Model Patterns: A Metadata Map*. Morgan Kaufmann, 2006. Print. The Morgan Kaufmann Series in Data Management Systems.

Hay, David C. *Enterprise Model Patterns: Describing the World (UML Version)*. Technics Publications, LLC, 2011. Print.

Hay, David C. *Requirements Analysis from Business Views to Architecture*. Prentice Hall, 2002. Print.

Hay, David C. *UML and Data Modeling: A Reconciliation*. Technics Publications, LLC, 2011. Print.

Hernandez, Michael J. *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design*. 2nd ed. Addison-Wesley Professional, 2003. Print.

Hoberman, Steve, Donna Burbank, Chris Bradley, et al. *Data Modeling for the Business: A Handbook for Aligning the Business with IT using High-Level Data Models*. Technics Publications, LLC, 2009. Print. Take It with You Guides.

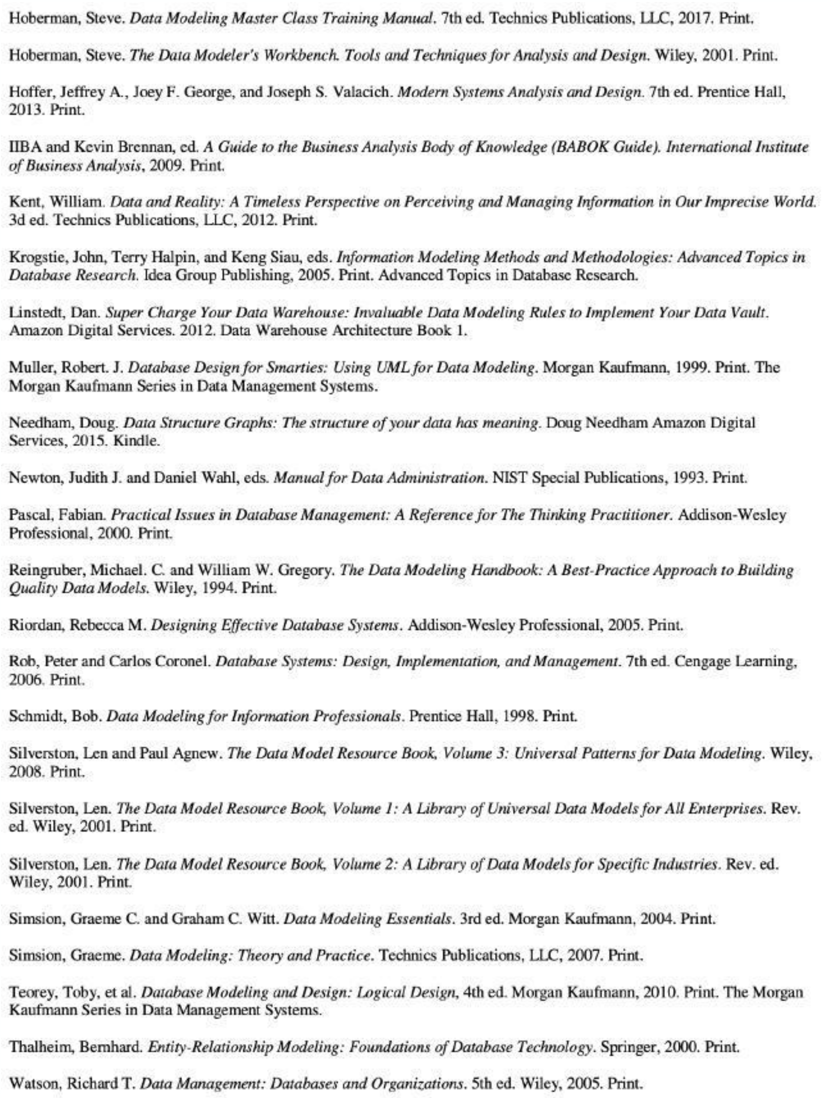
Hoberman, Steve. *Data Model Scorecard*. Technics Publications, LLC, 2015. Print.

Hoberman, Steve. *Data Modeling Made Simple with ER/Studio Data Architect*. Technics Publications, LLC, 2013. Print.

Hoberman, Steve. *Data Modeling Made Simple: A Practical Guide for Business and IT Professionals*. 2nd ed. Technics Publications, LLC, 2009. Print.

from *The DAMA Guide to the Data Management Body of Knowledge* 2E © 2017 by DAMA International

Works Cited



Hoberman, Steve. *Data Modeling Master Class Training Manual*. 7th ed. Technics Publications, LLC, 2017. Print.

Hoberman, Steve. *The Data Modeler's Workbench. Tools and Techniques for Analysis and Design*. Wiley, 2001. Print.

Hoffer, Jeffrey A., Joey F. George, and Joseph S. Valacich. *Modern Systems Analysis and Design*. 7th ed. Prentice Hall, 2013. Print.

IIBA and Kevin Brennan, ed. *A Guide to the Business Analysis Body of Knowledge (BABOK Guide). International Institute of Business Analysis*, 2009. Print.

Kent, William. *Data and Reality: A Timeless Perspective on Perceiving and Managing Information in Our Imprecise World*. 3d ed. Technics Publications, LLC, 2012. Print.

Krogstie, John, Terry Halpin, and Keng Siau, eds. *Information Modeling Methods and Methodologies: Advanced Topics in Database Research*. Idea Group Publishing, 2005. Print. Advanced Topics in Database Research.

Linstedt, Dan. *Super Charge Your Data Warehouse: Invaluable Data Modeling Rules to Implement Your Data Vault*. Amazon Digital Services. 2012. Data Warehouse Architecture Book 1.

Muller, Robert. J. *Database Design for Smarties: Using UML for Data Modeling*. Morgan Kaufmann, 1999. Print. The Morgan Kaufmann Series in Data Management Systems.

Needham, Doug. *Data Structure Graphs: The structure of your data has meaning*. Doug Needham Amazon Digital Services, 2015. Kindle.

Newton, Judith J. and Daniel Wahl, eds. *Manual for Data Administration*. NIST Special Publications, 1993. Print.

Pascal, Fabian. *Practical Issues in Database Management: A Reference for The Thinking Practitioner*. Addison-Wesley Professional, 2000. Print.

Reingruber, Michael. C. and William W. Gregory. *The Data Modeling Handbook: A Best-Practice Approach to Building Quality Data Models*. Wiley, 1994. Print.

Riordan, Rebecca M. *Designing Effective Database Systems*. Addison-Wesley Professional, 2005. Print.

Rob, Peter and Carlos Coronel. *Database Systems: Design, Implementation, and Management*. 7th ed. Cengage Learning, 2006. Print.

Schmidt, Bob. *Data Modeling for Information Professionals*. Prentice Hall, 1998. Print.

Silverston, Len and Paul Agnew. *The Data Model Resource Book, Volume 3: Universal Patterns for Data Modeling*. Wiley, 2008. Print.

Silverston, Len. *The Data Model Resource Book, Volume 1: A Library of Universal Data Models for All Enterprises*. Rev. ed. Wiley, 2001. Print.

Silverston, Len. *The Data Model Resource Book, Volume 2: A Library of Data Models for Specific Industries*. Rev. ed. Wiley, 2001. Print.

Simsion, Graeme C. and Graham C. Witt. *Data Modeling Essentials*. 3rd ed. Morgan Kaufmann, 2004. Print.

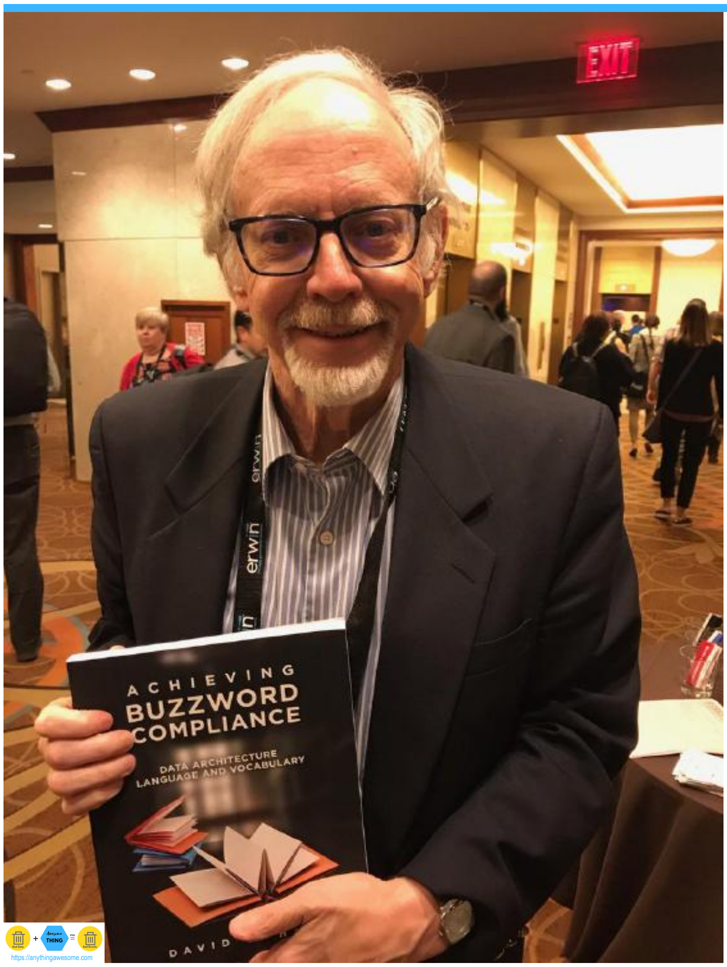
Simsion, Graeme. *Data Modeling: Theory and Practice*. Technics Publications, LLC, 2007. Print.

Teorey, Toby, et al. *Database Modeling and Design: Logical Design*, 4th ed. Morgan Kaufmann, 2010. Print. The Morgan Kaufmann Series in Data Management Systems.

Thalheim, Bernhard. *Entity-Relationship Modeling: Foundations of Database Technology*. Springer, 2000. Print.

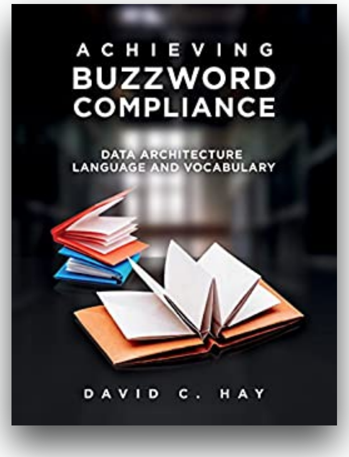
Watson, Richard T. *Data Management: Databases and Organizations*. 5th ed. Wiley, 2005. Print.

from *The DAMA Guide to the Data Management Body of Knowledge* 2E © 2017 by DAMA International



Achieving Buzzword Compliance

Data Architecture Language and Vocabulary



ACHIEVING
BUZZWORD
COMPLIANCE

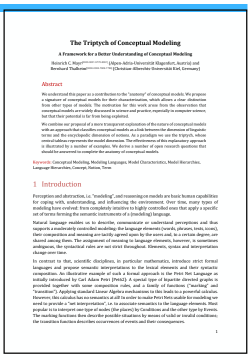
DATA ARCHITECTURE
LANGUAGE AND VOCABULARY

DAVID C. HAY

amazon.com link:

https://www.amazon.com/Achieving-Buzzword-Compliance-Architecture-Vocabulary-ebook/dp/B07FG1WRSD/ref=sr_1_1?crid=2QL3ZWKU2L3VC&keywords=Achieving+Buzzword+Compliance%3A+Data+Architecture+Language+and+Vocabulary&qid=1657032460&sprefix=achieving+buzzword+compliance+data+architecture+language+and+vocabulary%2Caps%2C324&sr=8-1

Research Efforts



The Triptych of Conceptual Modeling

A Framework for a Better Understanding of Conceptual Modeling

Heinrich C. Mayr[0000-0001-5770-8091] (Alpen-Adria-Universität Klagenfurt, Austria) and
Bernhard Thalheim[0000-0002-7909-7786] (Christian-Albrechts-Universität Kiel, Germany)

Abstract

We understand this paper as a contribution to the "anatomy" of conceptual models. We propose a signature of conceptual models for their characterization, which allows a clear distinction from other types of models. The motivation for this work arose from the observation that conceptual models are widely discussed in science and practice, especially in computer science, but that their potential is far from being exploited.

We combine our proposal of a more transparent explanation of the nature of conceptual models with an approach that classifies conceptual models as a link between the dimension of linguistic terms and the encyclopedic dimension of notions. As a paradigm we use the triptych, whose central tableau represents the model dimension. The effectiveness of this explanatory approach is illustrated by a number of examples. We derive a number of open research questions that should be answered to complete the anatomy of conceptual models.

Keywords: Conceptual Modeling, Modeling Languages, Model Characteristics, Model Hierarchies, Language Hierarchies, Concept, Notion, Term

1 Introduction

Perception and abstraction, i.e. "modeling", and reasoning on models are basic human capabilities for coping with, understanding, and influencing the environment. Over time, many types of modeling have evolved: from completely intuitive to highly controlled ones that apply a specific set of terms forming the semantic instruments of a (modeling) language.

Natural language enables us to describe, communicate or understand perceptions and thus supports a moderately controlled modeling: the language elements (words, phrases, texts, icons), their composition and meaning are tacitly agreed upon by the users and, to a certain degree, are shared among them. The assignment of meaning to language elements, however, is sometimes ambiguous, the syntactical rules are not strict throughout. Elements, syntax and interpretation change over time.

In contrast to that, scientific disciplines, in particular mathematics, introduce strict formal languages and propose semantic interpretations to the lexical elements and their syntactic composition. An illustrative example of such a formal approach is the Petri Net Language as initially introduced by Carl Adam Petri [Pet62]: A special type of bipartite directed graphs is provided together with some composition rules, and a family of functions ("marking" and "transition"). Applying standard Linear Algebra mechanisms to this leads to a powerful calculus. However, this calculus has no semantics at all! In order to make Petri Nets usable for modeling we need to provide a "net interpretation", i.e. to associate semantics to the language elements. Most popular is to interpret one type of nodes (the places) by Conditions and the other type by Events. The marking functions then describe possible situations by means of valid or invalid conditions; the transition function describes occurrences of events and their consequences.

1

- Professor Bernhard Thalheim and associated research efforts have contributed much to these topics including:

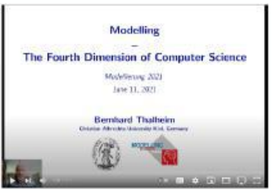
- Conceptual modelling
  - https://www.youtube.com/watch?v=Y9_7KSsSUpg
  - https://www.youtube.com/watch?v=mKcwbR6uJwU

Models and Modelling in Computer Science: The Grand Opus

MMM module lectures: Models, To Model, Modelling & Conceptualisation and Conceptual Models

April 2020

Prof. Dr. Bernhard Thalheim
Information Systems Engineering Group
Computer Science Institute
Kiel University, Germany



Modelling
–
The Fourth Dimension of Computer Science

Modellierung 2021
June 11, 2021

Bernhard Thalheim
Christian Albrechts University Kiel, Germany

- Claim: logical models also conceptual models
  - https://www.youtube.com/watch?v=L8yGjEbwTsQ
  - https://link.springer.com/article/10.1007/s10270-020-00836-z
  - https://dl.acm.org/doi/10.1007/s10270-020-00836-z

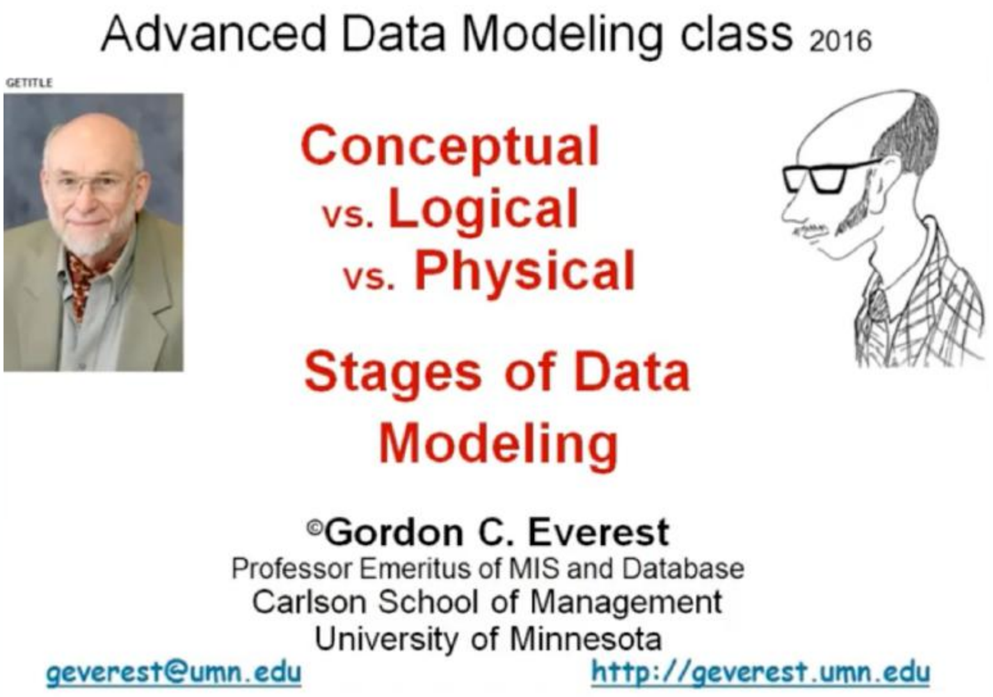
SAMVAAD TALK
WEEKLY SERIES • ONLINE

Title:
Towards Foundations of Conceptual Modeling

Speaker:
Prof. Bernhard Thalheim
[illegible]
Information Systems Engineering
Christian-Albrechts
-Universität zu Kiel
[illegible]

https://anythingawesome.com

Https://Www.Youtube.Com/Watch?V=9qWjpVtr_Hg&T=2s


Advanced Data Modeling class 2016

GETITLE

Conceptual
vs. Logical
vs. Physical

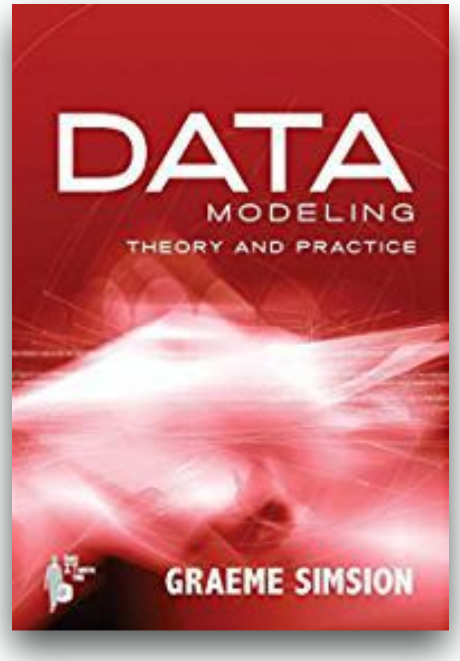
Stages of Data Modeling

©Gordon C. Everest
Professor Emeritus of MIS and Database
Carlson School of Management
University of Minnesota

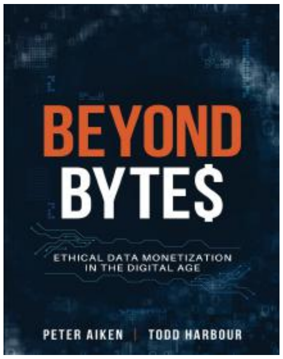
geverest@umn.edu http://geverest.umn.edu

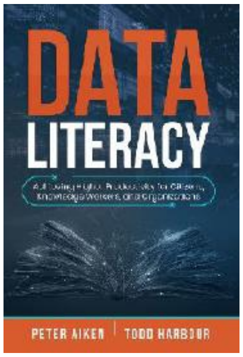
Data Modeling: Theory and Practice

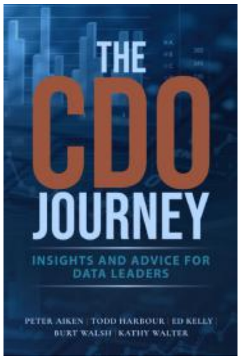
Graeme Simsion

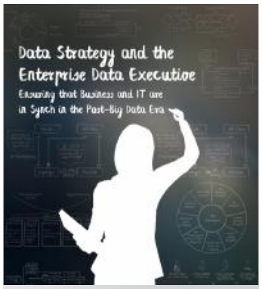


Event Pricing on Peter's Books

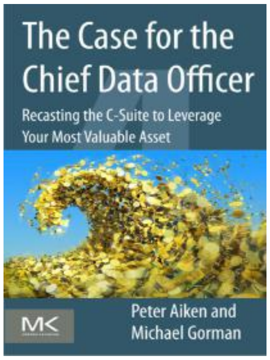

BEYOND BYTE$
ETHICAL DATA MONETIZATION IN THE DIGITAL AGE
PETER AIKEN | TODD HARBOUR

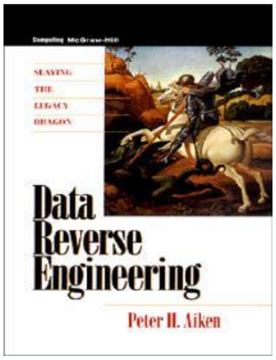


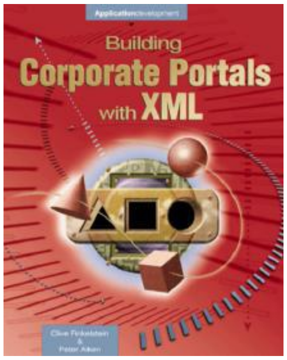

THE CDO JOURNEY
INSIGHTS AND ADVICE FOR DATA LEADERS
PETER AIKEN | TODD HARBOUR | ED KELLY | BURT WALSH | KATHY WALTER

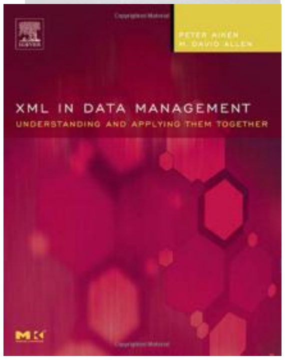

Data Strategy and the Enterprise Data Executive
Ensuring that Business and IT are in Synch in the Post-Big Data Era

- 25% off directly from the publisher on some titles
- My 'Book Store' @
  https://anythingawesome.com/books
- Enter the code
  “TP25" at the Technics bookstore
  checkout where it says to "Apply Coupon"


The Case for the Chief Data Officer
Recasting the C-Suite to Leverage Your Most Valuable Asset
MK
Peter Aiken and Michael Gorman







[Clicking any webinar title will link directly to the registration page]

Upcoming Events

Four Effective Metadata Strategies

12 Aug 2025

Engineering Data Quality: A Framework for Effective Data Management

9 September 2025

Data-Ed Online: Data Strategy: Where Data Architecture and Data Governance Collide

14 October 2025

Brought to you by:

Time: 19:00 UTC (2:00 PM NYC) | Presented by: Peter Aiken, PhD

DATAVERSITY

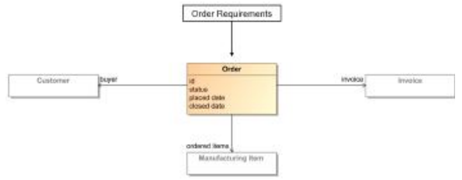
5. Create Read Update Delete (CRUD) Matrix

Model Purpose Statement: To describe the manner and sequence in which processes interact with specific data items

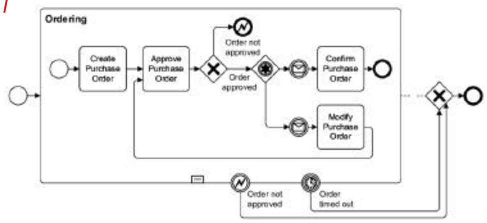
	Data Item 1	Data Item 2	Data Item 3	Data Item 4
Process A	Creates			
Process B	Reads			
Process C	Updates			
Process D	Deletes			

Motivation-EA Concepts/Components

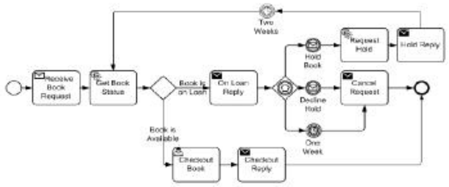
Context Model



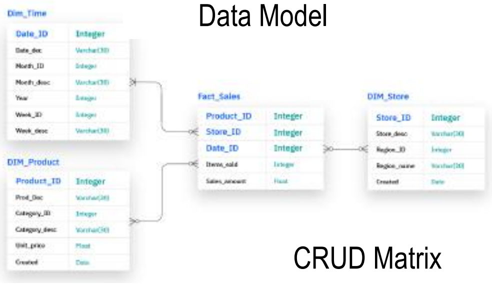
Process A



Process B



Data Model



CRUD Matrix

	Data Item 1	Data Item 2	Data Item 3	Data Item 4
Process A	Creates			
Process B	Reads			
Process C	Updates			
Process D	Deletes			

Controlled Vocabulary

Controlled Vocabulary

Controlled Vocabulary

FM Ontology

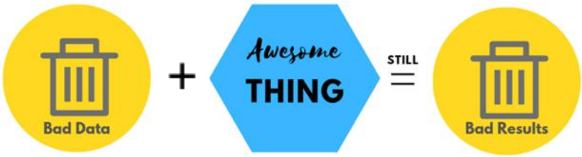
Critical Design Review?

Mentoring?

Collaboration?

Independent Verification & Validation

Executive Data Literacy Training?



Peter.Aiken@AnythingAwesome.com +1.804.382.5957

Reverse Engineering Expertise?

Hiring Assistance?

Thank You!

Use your data more strategically?

Tool/automation evaluation?

Book a call with Peter to discuss anything - https://anythingawesome.com/OfficeHours.html

The Triptych of Conceptual Modeling

A Framework for a Better Understanding of Conceptual Modeling

Heinrich C. Mayr[0000-0001-5770-8091] (Alpen-Adria-Universität Klagenfurt, Austria) and
Bernhard Thalheim[0000-0002-7909-7786] (Christian-Albrechts-Universität Kiel, Germany)

Abstract

We understand this paper as a contribution to the "anatomy" of conceptual models. We propose a signature of conceptual models for their characterization, which allows a clear distinction from other types of models. The motivation for this work arose from the observation that conceptual models are widely discussed in science and practice, especially in computer science, but that their potential is far from being exploited.

We combine our proposal of a more transparent explanation of the nature of conceptual models with an approach that classifies conceptual models as a link between the dimension of linguistic terms and the encyclopedic dimension of notions. As a paradigm we use the triptych, whose central tableau represents the model dimension. The effectiveness of this explanatory approach is illustrated by a number of examples. We derive a number of open research questions that should be answered to complete the anatomy of conceptual models.

Keywords: Conceptual Modeling, Modeling Languages, Model Characteristics, Model Hierarchies, Language Hierarchies, Concept, Notion, Term

1 Introduction

Perception and abstraction, i.e. "modeling", and reasoning on models are basic human capabilities for coping with, understanding, and influencing the environment. Over time, many types of modeling have evolved: from completely intuitive to highly controlled ones that apply a specific set of terms forming the semantic instruments of a (modeling) language.

Natural language enables us to describe, communicate or understand perceptions and thus supports a moderately controlled modeling: the language elements (words, phrases, texts, icons), their composition and meaning are tacitly agreed upon by the users and, to a certain degree, are shared among them. The assignment of meaning to language elements, however, is sometimes ambiguous, the syntactical rules are not strict throughout. Elements, syntax and interpretation change over time.

In contrast to that, scientific disciplines, in particular mathematics, introduce strict formal languages and propose semantic interpretations to the lexical elements and their syntactic composition. An illustrative example of such a formal approach is the Petri Net Language as initially introduced by Carl Adam Petri [Pet62]: A special type of bipartite directed graphs is provided together with some composition rules, and a family of functions ("marking" and "transition"). Applying standard Linear Algebra mechanisms to this leads to a powerful calculus. However, this calculus has no semantics at all! In order to make Petri Nets usable for modeling we need to provide a "net interpretation", i.e. to associate semantics to the language elements. Most popular is to interpret one type of nodes (the places) by Conditions and the other type by Events. The marking functions then describe possible situations by means of valid or invalid conditions; the transition function describes occurrences of events and their consequences.

In natural or technical sciences, this approach is reversed: initially, a conceptualization[1] of the domain of interest is established and subsequently one or more (textual and/or graphical) languages are defined for representing its elements and relationships. Think for instance of a conceptualization of electrical components that are represented using electrical circuit diagrams. The Unified Modeling Language UML comes with a conceptualization of abstract elements like class, attribute, relation, state, activity for describing domains of interest. Similarly, the Business process Model and Notation BPMN comes with a conceptualization abstract elements like actors, activities, or decisions. A branch of Knowledge Engineering deals with so-called action languages that are based on the claim that "*action theories always model - explicitly or implicitly - the general notions of time, change and causality*" [PP19].

Often, such languages are called "*conceptual modeling language*s" and their use as "*conceptual modeling*" – although despite countless attempts, there is no generally used strict definition of what constitutes conceptual modeling and what does not.

One group of such definition attempts are variants of "*Conceptual Modeling is Modeling with Concepts*" [Kan15, Tha18], and

- introduce these concepts via more or less rigid ontological frameworks, or by simple explanation using natural language; [vF72, vF91] called this latter approach "a priori semantics";
- propose more or less formalized constructs for representation, i.e. a 'modeling language";
- and often call the approach "semiformal", an awful wording per se as it just indicates, that the proposed framework does not fulfil the criteria demanded for a consistent calculus that can be used for correctness proofs etc.

This way of defining "conceptual modeling", however, does not provide hard criteria for differentiating it from other modeling methods in individual cases. For example, most conceptual modelers would say that programming or relational database design is not conceptual modeling (see, e.g. [Myl20]). Nevertheless, programming languages or the SQL DDL work with conceptualizations, the latter for instance featuring elements like "Relation" or "Attribute" that have some basic semantics and therefore might be seen to be concepts in the above definition's sense. From a practical point of view, this open question is not a real problem. However, the term "Conceptual Modeling" is widely used, and there has been an international conference with this name for 39 years. Therefore, it would be desirable to have a definition or at least a set of criteria at hand that would allow us to define more precisely what is and what is not a conceptual model. We will return to this question in section 4.

Recent initiatives (e.g. [DLPS18, GGM20, Tha18]) try to compile and analyze systematically existing definitions and opinions in order to filter out a better understanding of the nature of conceptual modeling. [Myl20] offers "*three complementary theses, answers to the question 'What is a conceptual model?'*". These theses essentially state that conceptual models are (1) computational because they are stored in computers and are analyzed and justified by computers, (2) artifacts, so they should have requirements dictated by Engineering, and (3) social artifacts, because they must capture the common conceptualization of a group.

In fact, from the very beginning, conceptual modeling was propagated as a means to improve the design and implementation of whatsoever software system, especially with regard to a

[1] http://www.webster-dictionary.org/definition/conceptualisation (accessed on August 3rd 2020): "*A conceptualisation is an abstract, simplified view of the world that we wish to represent*"

comprehensive and as clear as possible elicitation and analysis of system requirements. Until now, however, the practical use has mostly taken place at the level of mere drawings, which do not play a major role in the further development process and are rarely adapted to changes. Consequently, the developed software (nota bene: again a model) usually deviates considerably from what was originally modeled. The situation in Business Process Modeling is not much different, independent of the modeling method used (like BPMN, Adonis, Event Chains etc.). This means that the potential of conceptual modeling is far from being exploited.

MDA/MDSD approaches [KWB03] and models@runtime [BG18] are enforcedly more aligned with the system life cycle as they use the models for generating or driving the targeted software. Usually, they work with well-defined subsets or variants of known modeling methods [FR07, GR19, and PR18]. However, also these approaches do not have a breakthrough in practice.

Worse still, university graduates who highly motivated join a company often quickly lose their enthusiasm when they are told that modeling is too expensive in terms of effort and cost, not paid for by the customer, and has no impact on the quality of the software development process, since “agile developers” know what they are doing.

We assume that all this is mainly due to the fact that inventors and propagandists of conceptual modeling languages like ourselves have so far failed to make the anatomy of conceptual modeling and its benefits transparent to users. Instead, we invented hundreds of variants of “modeling languages” always believing that it should be a must for the targeted user to acknowledge and happily exploit the miracle we presented to her/him. Moreover, uncountable papers present what they call “ontologies” and expect the readers to internalize and share these without contradiction. Others implicitly equate conceptual modeling with "graphical modeling" (KM20) and thus not only add to the confusion but also distort the view of the essential.

We believe, therefore, that in order to make conceptual modeling more attractive for practitioners, we have (1) to provide a clear conception of what we are speaking about, (2) to make the anatomy of conceptual modeling transparent with its principles, paradigms, postulates, assumptions, particularities, specifics, potential, capacity and limitations, and (3) to allow the modelers to easily create and use their own domain and culture tailored modeling language and method instead of forcing them to learn and deal with ours.

With this paper we would like to make a contribution to these To-do’s. We offer here our understanding of what modeling, in particular, conceptual modeling is about, and how we can clearly distinguish it from other modeling approaches. The perspective presented reflects four decades of dealing with conceptual modeling in research and practice, countless discussions with colleagues and practitioners, the rich body of knowledge published up to now, as well as long and intensive working meetings the authors had over the last two years. But we have neither the intention to improve the world nor to provide an n+1st definition of what “conceptual modeling” is. Rather, we present a "signature" of conceptual modeling in the sense of a framework of characteristics by which conceptual modeling can be categorized. In other words, we will offer an explanatory framework that could help to better understand the nature of conceptual modeling.

The paper is structured as follows: Section 2 explores specifics of models and introduces six characteristics that can be observed for models. Section 3 refines these six characteristics in terms of a list of criteria that can be used to determine the nature of CM. In section 4 we summarize the two previous sections and discuss the first conclusions that can be drawn from them. This will provide the basis for section 5, where we present the core message of the paper: the triptych paradigm of conceptual modeling together with its dimensions and model/language hierarchies. The paper ends with a conclusion and an outlook on open research challenges in section 6.

We will reference related work where appropriate but, intentionally, there will be no separate section on related work. Instead, we refer to the rather comprehensive overview given by Thalheim in [Tha18], to [Wik17] and to attempts to define the term "model" [TN15a].

Finally, we would like to point out that, for the sake of readability, we also adopt the usual homonymous use of the term "model" in this paper: From an epistemological point of view, a model is a mental object. In practice, however, the representation of a model introduced into the perceivable world is also referred to as a model, like, e.g. an Entity-Relationship diagram. We adopt this homonymy because the particular meaning will result from the respective context.

2 Characteristics of Models

Across disciplines, the number of publications dealing with models, modeling and abstraction are unmanageable. Even for the notion of "conceptual model" more than 60 different definitions can easily be found [Wik17, Tha18, Myl20]. None of these, however, allows for a robust and unequivocal differentiation between conceptual and non-conceptual models. This is also true for an interesting definition that recently emerged in a side-piece discussion at ER 2017: "*A conceptual model is a partial representation of a domain that can answer a question*". For, it only highlights one aspect.
We, therefore, try to elaborate the essence of conceptual modeling in the form of a taxonomy of characteristics that may help to better delimit the semantics of the term "conceptual model".

Before we can do this, we first need to take a closer look at the terms "concept", "notion" and "term". The reader will have noticed that we have avoided their use as much as possible so far. The reason for this is that the meanings of these terms in literature and in encyclopedias are not sharply delineated, so that there are overlapping or synonymous definitions. For the purposes of this paper, however, we need a more precise distinction (which will be further specified in section 5). We, therefore, assume the following meanings in the subsequent sections:

- A ***Concept*** is a mental construct formed by mentally combining characteristics of general or abstract ideas gained by cognition. It is seen as a pair of an intension and its extension. The intension describes the concept as such, the extension consists of all objects that might be used as an example for the intension. This definition is based on [We20] (a concept is "*something conceived in the mind*"), Wordnet [MBF90] ("*an abstract or general idea inferred or derived from specific instances*"), the Stanford Encyclopedia of Philosophy[2] ("*concepts are constituents of thoughts*"), [BMS86, Kan15] and [Mur01].
- A ***Notion*** is a general inclusive concept in which some confidence is placed; i.e. a notion is a specific kind of concept[3]. This definition is based on Wordnet ("*a notion is a general understanding, vague idea or a general inclusive concept in which some confidence is placed*") and [We20] who propose "*arriving at the notion of law*" as an example for the interpretation of notion as a general inclusive concept.

[2] https://plato.stanford.edu/entries/concepts/

[3] With this interpretation *notion* corresponds to the German "*abstrakter Begriff*" [KB71] as "*mental and abstract reflection of a class of individuals or classes on the basis of their invariant characteristics ... i.e. specific concepts as abstract essences ... (ideas)*".

- A ***Term*** is an item of a (possibly formal) language formed for denoting, designating, or naming something. "Language" is understood here in a very broad sense, i.e. it can be textual, graphical but also material. Terms can refer to concepts and/or represent them for recognition by linguistic perception processes, i.e. processes mapping a term/symbol to a mental object. Note that this interpretation of term is inspired by one of the definitions given in [We20] ("*a pronounceable series of letters having a distinct meaning especially in a particular field*"), but differs from others. We use it here to clearly separate "term" from "concept" and "notion". Also, for the sake of clarity, we will not use any other word with a similar meaning throughout the paper (such as for example "sign").

Second, as conceptual models are models, we have to agree on the key characteristics of models before specializing and extending these to determine what the characteristics of conceptual models are. For this purpose, we adopt the main criteria provided by [Mah05] that may be summarized by "*A model is the synthesis of a conceptual idea, a form of expression and the assumption of a role through which it fulfils a function*"[4].

Model Characteristic 1: Models are related to (a collection) of "origins" or "originals". A model is a model of something[5], i.e. it is a proxy of a natural, artificial or mental original; in particular, the original of a model may be a model itself. As originals may change in time, the model/original relationship may change in time as well [Sta73]. Models are results of cognitive processes (perception) [vF72]. The mission of a model is that of transporting a "*cargo*", namely the perceived properties of the original that are considered to be relevant within the perception's context. Mahr sees this function as the key criterion for a "*model being a model*" [Mah15]. The transport occurs with the usage of the model, precision and transport warranties distinguish models and metaphors [Mah08].

Model Characteristic 2: **Concern and Usage**. We distinguish three different main concerns that are coupled to most kinds of modeling: (1) understanding, (2) communicating, and (3) agreeing as a process of consolidation, manifestation, and consensus. With the usage, a model unfolds its power: "*We place models between ourselves as perceiving, recognizing, understanding, judging or acting subjects and the world as perceptible, observable, effective, to be judged or produced exterior. The impact of models results from the role that models play through their transport function in work processes, cognitive processes, business processes. The power of models is the result of their power to act"*[6][7] [Mah05]. In general, the usage of a model will be directed by its initial concern. However, this is not mandatory, because the using individual can do what she/he wants with a model.

Model Characteristic 3: **Purpose and Function**. Given its concern and usage, a model serves a particular purpose: to understand/analyze/assess the origin, to plan/design a new original, to explain or predict properties of the original, to communicate about perceptions and ideas, and

[4] Original quotation in German: „*Ein Modell ist die Synthese einer begrifflichen Vorstellung, einer Ausdrucksform und einer Einnahme einer Rolle, durch die es eine Funktion erfüllt*." Bernd Mahr cites here George A. Millers work „ *The science of words* ", which was not accessible for us directly [Mil91].

[5] "*Every mental phenomenon has an object towards it is directed*" [Bre74].

[6] Original quotation in German: „*Wir stellen Modelle zwischen uns als wahrnehmende, erkennende, verstehende, urteilende oder handelnde Subjekte und die Welt als wahrnehmbares, beobachtbares, wirkendes, zu beurteilendes oder herzustellendes Äußeres. Die Wirkungsmacht von Modellen ergibt sich aus der Rolle, die Modelle durch ihre Transportfunktion in Werkprozessen, Erkenntnisprozessen, Unternehmensprozessen spielen. Die Macht von Modellen ist das Ergebnis ihrer Wirkungsmacht".*

[7] Translated with www.DeepL.com/Translator

similar. The usage determines the function(s) of a model, for example to support explanation. It therefore makes sense to see a model's function as that of an "*instrument*" [Tha19, TN15b].

Model Characteristic 4: **Domain and Context**. For the concerns of modeling, we distinguish the following three domains:

a. the domain of interest, experience, and perspective of a human,
b. the application domain or world domain to which a community of practice refers,
c. the domain of discourse among some particular people.

The first domain is concerned with understanding and thinking. So is the second one which additionally is concerned with realization (in the sense of implementation). The third domain is concerned with communication.

A model is created, modified or refined in particular contexts: The personal context of the modeler, the environmental context in which the modeling process takes place, the social context, i.e. the particular community of practice, and the spatio-temporal context (time, duration, location, and movement etc.) [MM13, MM16].

Clearly, a model's cargo as well as its interpretation depends on the given concern, purpose, domain and context.

Model Characteristic 5: Focus. A model reflects, for a given purpose, the "relevant" but not all aspects of its origin(al). In particular, "*The objectual properties may recede behind the consideration of their rational-functional relationships*" [Wol96]. Note, that this is a more general view than that of [Sta73], who emphasizes on reduction, i.e. differentiates between "modeled attributes" and "neglected attributes" of the origin(al).

Model Characteristic 6: Representation. For communication/transportation purposes, a model needs an associated "physical" representation; examples are an acoustic signal, a toy railroad, a diagram, a XML statement, an OWL file, a spoken/written natural language text, and so forth. These representations allow models to be recognized and understood by communication partners; in the case of a human partner, recognition is enhanced by "linguistic perception" [vF72]. The representations should be dependable, understandable by the involved actors (humans and/or systems), and thus be agreed within the community of practice. George A. Miller explained the relationship between a model and its representation as follows: "*To have a model means to be able to produce or recognize a physical symbol carrier that represents a model, and to understand the meaning of the model*" [Mil91].

Clearly, this taxonomy is not complete, as the literature addresses many more characteristics. For example, see the "Kiel house of modeling" [TN15a][8]. However, it should not be a problem to classify most of them in relation to the characteristics presented.

3 Characteristics of Conceptual Models

Conceptual models are models, conceptual modeling is (a kind of) modeling. Consequently, the characteristics described in section 2 also apply to conceptual models. So we need to identify what constitutes the specialization "conceptual". To this end, we will now, wherever possible, specialize

[8] http://bernhard-thalheim.de/ModellingToProgram/

the above model characteristics and introduce two more that we believe are specific to conceptual models.

CM Characteristic 1: Conceptual models are related to (a collection of) origins or originals. As there is no restriction on the entirety of origin(al)s conceptual models may relate to, this characteristic does not provide an indication for differentiation.

CM Characteristic 2: **Concern and Usage**. In the discipline of Informatics the term "Conceptual Modeling" has been initially used for a database design method, later on for requirements modeling and since the 90ies for business process modeling and software specification. In all cases, the mapping from conceptual models (represented using languages like the ERM, UML, BPMN, SysML etc.) to an implemented system language (SQL DDL, programming languages, workflow languages etc.) has been a key issue until today. Model Driven Software Development (MDSD) [SK03], Model Driven Architecture [KWB03] as well as models@runtime [BG18] all start from conceptual models and aim at materializing and automating that mapping. Model Centered Architecture (MCA) [MMR17] advocates, for any aspect of a system under development, the use of Domain Specific Modeling Languages (DSML), i.e. focuses on models (and their metamodels) in any design and development step up to the running system. In summary, conceptual modeling has a strong (although not mandatory) orientation to a subsequent implementation/realization of artifacts or products. It, therefore, is widely used as a means for requirements modeling and analysis. Consequently, we may add the **concern (4) "specifying"** to the list of concerns. Unlike [Myl20], however, we do not claim that conceptual models are 'computational' - and that they have only existed since computers have existed.

CM Characteristic 3: **Purpose and Function**. As a consequence of extending the concern we supplement the purpose "plan/design a new original" with "plan/design/realize".

CM Characteristic 4: **Domain and Context**. There is no principal limitation regarding domains and contexts of conceptual modeling so that also this characteristic provides no hard criterion for differentiation. However, in practice, conceptual modeling has been mainly used so far in domains and contexts that deal with discrete objects (things, actions), their properties and relationships. Again, however, this is no strong criterion for differentiation.

CM Characteristic 5: Focus. Conceptual models have no noteworthy peculiarity regarding this general model characteristic. In practice, however, their focus has been mainly on aspects that can be realized or implemented.

CM Characteristic 6: Representation. Conceptual models transport semantics by terms that denote concepts. As terms are elements of languages, conceptual modeling uses linguistic representations in the broadest sense: these may originate from a diagrammatic language (e.g. ER diagrams), a natural language, an artificial language (e.g. XML), a mathematical or formal language (e.g. Petri nets in the sense of algebraic structures). Such languages provide a set of literals and a set of rules for composing literals to terms, terms to phrases, and phrases to sentences and so on. If the members of a certain community of practice have agreed on the meaning of terms or patterns and their combination, they can infer from these to the transported concepts.

CM Characteristic 7: Concept Space. From its beginnings, conceptual modeling had a strong relation to semantics. Partly, "semantic modeling" using terms that are associated with concepts from a "concept space" is even used as a synonym [FGH92] of conceptual modeling: A community of practice agrees on the terms and concepts, which it will consider, as well as on the association between these terms and concepts, and thus establishes an instrument for communication. The terms used for representing models thus have a meaning, the “a priori semantic” [vF91].

So we can consider conceptual models as models that are “enhanced” by concepts from a concept space. I.e., the decision to compile and accept a set of concepts and to use its elements for relating them to models opens the entrance into the world of conceptual modeling (we will discuss this in detail within the next chapters). This characteristic governs all others and reminds to the definition “*Conceptual Modeling is modeling with concept*s” cited in the introduction. Reflecting the considerations presented so far, a more apposite description could be “*Conceptual Modeling is modeling with concepts from an associated concept space*”.

Such association provides a semantical basis supporting understanding communicated models within a community of practice (see CM characteristic 2): as a prerequisite, this community agrees in advance on a set of concepts to be used for modeling, their meaning and representation (controlled vocabulary). Usually this is done informally in natural language, i.e., relating an explaining natural language phrase to the given concept. As an example think of Peter Chen’s explanation of the concepts “entity” and “relation” [Che76]: “*An entity is a “thing” which can be distinctly identified. A specific person, company, or event is an example of an entity. A relationship is an association among entities. For instance, “father-son” is a relationship between two* ‘person’ entities”. I.e., the semantics of natural language - and thus its intrinsic a priori [Pla03, Kan81, Lat17] knowledge - are used to determine (the meaning of) concepts.

Therefore, the degree of common understanding of the elements of a concept space by the members of a community of practice depends on the degree of equivalence of their understanding of the natural language used. As such equivalence cannot be formally derived or proven without a reference mechanism like an ontology or a set of axioms, the “a priori semantics” [vF72] of conceptual models provide a practically useful but formally inaccurate means for communication. Some people, therefore, call conceptual models “semiformal” as has been mentioned in the introduction.

CM Characteristic 8: Concept Relationship. Concepts can be related to each other. Typical concept relationships are the “abstractions” [SS77] Mereology (Aggregation), Generalization, and Intension, each of them having an inverse: Disassembly (into components), Specialization (by additional concept attributes), Extension (denominating the elements characterized by their intension concept) [LMN93, LMW79]. Other concept relationships are, e.g., synonymy, homonymy, troponymy, hyponymy; however, these only concern the level of assigning linguistic denoters to concepts. Therefore, such relationships can be found in thesauri and encyclopedias, since they occur at the linguistic level.

Note that a conceptual modeling language that offers explicit means for modeling the intension/extension relationship[9], supports “multi-level” modeling. For, models then are not mere extensions of a given meta-model but may consist themselves of intension/extension concept hierarchies. In the field of Domain Specific Modeling this possibility is often neglected: Metametamodel abstraction relations are just used for relating metamodel concepts, but metamodels often do not explicitly provide such relations for allowing the same on the modeling level, i.e., the relationships are not introduced as part of the concept space in question.

[9] Intension/Extension are the concept relationships establishing model hierarchies with levels like metametamodel |metamodel | model | instance as provided, e.g., by the OMG MetaObject Facility [OMG] or the ISO Information Resource Dictionary System [ISO90.]

Table 1 summarizes these considerations on model characteristics.

Characteristics	**Model**	**Conceptual Model**
Relation to origins	A (conceptual) model is a model of something	
Concern and Usage	(1) understanding (2) communicating (3) agreeing	(1) - (3) + (4) specifying
Purpose and Function	(1) understand, analyze, assess (2) plan, design (3) explain, explore, predict, use	(1) - (3) + (4) realize
Domain and Context	Domain: (1) domain of interest, experience, and perspective of a human (2) application domain or world domain accepted by a community of practice (3) domain of discourse among some people Context: (1) personal context of the modeler (2) environmental context in which the modeling process takes place (3) social context, i.e. the particular community of practice, and (4) spatio-temporal context	
Focus	aspects of the origin(al) that are "relevant" for a given purpose	not mandatory but lived practice: aspects that can be realized or implemented
Representation	by physical symbol carriers	lived practice:-(in the broadest sense) linguistic terms
Concept Space		is associated with concepts from a concept space: a-priori semantics
Concept Relationship		semantic relationships between concepts induce semantic relationships between conceptual models.

Table 1: Characteristics of Models and Conceptual Models

4 Some initial results of using the characteristics

In this section, we use some examples to show how the previously introduced CM characteristics can be used to decide for a given model whether it is conceptual or not.

First, however, we note that models based on model hierarchy frameworks such as the Information Resource Dictionary IRDS [ISO90]) or the MetaObject Facility MOF [OMG] are not conceptual per se, although the model hierarchies are induced by concept relationships according to CM characteristic 8. For, a metamodel (on a hierarchy level H^{n+2}, e.g. on MOF Level M2) specifies modeling elements ("modeling concepts" in [LM78]) and their relationships but neither automatically nor explicitly associates these with a concept space in the sense of CM Characteristic 7. This would also not change if we cast the whole thing in languages, i.e. create linguistic means of expression for the formulation/representation of models on level H^{n+1} or model extensions on level H^{n} by defining corresponding grammars[10].

[10] Please note that we use a more general form of model hierarchy here, which can have any number of levels, possibly even nested ones. For practical purposes, especially for system design, the MOF or IRDS levels are of course sufficient, i.e. n=0 in this case.

The **Entity-Relationship-Model** (in the sense of a metamodel) thus only becomes "conceptual" when the meaning of the terms “entity set”, “relationship set”, “attribute” etc. is at least colloquially explained (a priori semantics) in an associated concept space. The same applies to the metamodel of the **UML**. In other words, modeling with UML is conceptual if the a priori semantic explanation of what is meant by "class", "relation" etc. is associated with the models. Pure drawing of diagrams or "graphical modeling" is not conceptual.

But what about the question we raised in the introduction: “*Is the **Relational Data Model** a conceptual one?*” Traditionally, answering this question was avoided by introducing the notion of „logical” model, i.e. a representation that is based on a “logical” language and the semantics defined by usage (or implementation). [Myl20] makes a more specific statement on this: “*Relational schemas are not conceptual as well, because they say nothing about the meaning of data in a database, only about its structure*.” At first sight, one seems to be able to get along with this statement, but we do not want to accept it as a generally valid one. For, the situation is identical to the situation described above regarding UML: if there is (within the given community of practice) a common accepted view on the meaning of the terms “relation”, “column”, “row”, “attribute” and so forth, the Relational Data Model (a metamodel) is a conceptual model, as it’s elements have an associated concept space. A relational schema represented in SQL DDL (on level H^1) and introducing common concepts (e.g. a table called „client“) then allows us to infer that client is an extension of the concept relation and has attributes (columns) describing clients’ properties, and rows describing particular clients.

If, at this stage, we chose the names of the columns from denominators that are well-known in our natural language NL (e.g., name, birthday …) then we can exploit the NL a-priori knowledge in order to intuitively interpret the tables. This was already recognized in the year 1977 by John and Diane Smith [SS77]: “*Since databases are usually designed to model the real world as we understand it, we can safely require that all object names in a relation definition be natural language nouns. These nouns then provide the bridge between our intuitive understanding of the real world and its intended reflection in the relation definition. If natural language nouns are not used, any discussion of the meaningfulness of a relation definition seems moot.*” Formally, however, these H^1 level concepts become related to the respective Universe of Discourse only, if the respective denominators („client“, „name”, etc.) and their a priori semantics are added to the concept space. This is often achieved by use of a data dictionary or by establishing an ontology.

Another question that sometimes gives our students headaches is: “*What is a **balance sheet** from a modeling perspective?*” Of course, our students first would ask us to specify more precisely, what we mean by "balance sheet":

(1) The usual components (concepts) of a balance sheet and their interrelationships as taught in a lecture on business administration, i.e. something that could be considered a metamodel, hierarchy level H^2?

(2) The balancing scheme of a company C, on the basis of which balance sheets for C can be drawn up at any reporting date, i.e. something that could be regarded as an extension of (1) and thus as (the representation of) a model, hierarchy level H^1?

(3) The balance sheet of enterprise C as at 31.12.2019, i.e. something that could be regarded as an extension of (2) and therefore as (the representation of) a model, hierarchy level H^0?

So far so good. But now we ask: Is a balancing scheme according to (2) a conceptual model? Again, this question can be answered with the help of the characteristics: Yes, it is a conceptual model, if the underlying metamodel associates with its elements the concept space of business administration with concepts like "assets", "liabilities", "cash on hand", equity capital, borrowed capital etc. and their relationships.

A somewhat easier to answer question than the previous one concerns the **Petri nets** already mentioned in the introduction: are they conceptual models? Again, for answering the question we have to specify more precisely, what we are concretely referring to by the term "Petri net". So let's restrict ourselves to the classical "marked Petri Net"[11]. Given the definition in the footnote, we are dealing here with a purely formal structure with which no semantics are associated. However, if we associate, as Petri suggested, a "net interpretation", i.e. a concept space, with (the elements of) the marked Petri Net it becomes a conceptual model. This, by the way, on hierarchy level H^n, since a change of marking leads to a new extension. On H^{n+1}, one could, as an example, consider (P,T,I,O) together with the set of all possible mappings M as the intension of the H^n model.

Comparable considerations can also be made about **circuit diagrams** in electrical engineering. It should be clear that such a diagram is not a pure drawing or formal graph structure but the graphical representation of a planned circuit or the description of a realized circuit. In the German term "Schaltplan" (literally translated as "circuit plan") this model character is clearly expressed. The diagram for a concrete circuit is located on hierarchy level H^0, but of course H^1 plans are also common as intensions, namely when they are generic, so that several concrete extensions can be derived from them (see Figure 1). But are circuit diagrams also conceptual models? We can answer this question clearly with yes: The metamodel uses concepts exclusively from the concept space of electrical engineering (power source, resistor, line, switch, lamp, etc.), the (graph-)grammatical composition rules for the symbols of the model representation language correspond to the physical rules of concept space's universe of discourse. Thus, the model in Figure 1 represents a conceptual model for circuits in which a lamp and a switch are connected in series to a Battery as a power source.

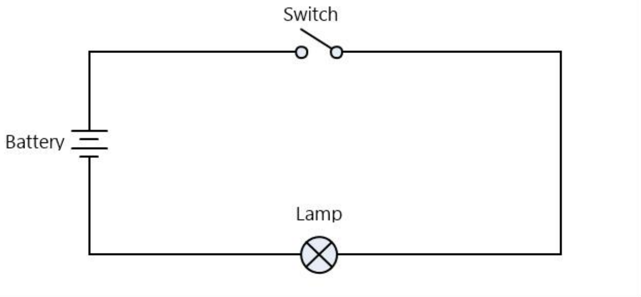


Figure 1: Simple circuit diagram

Another example concerns a **temple stele** (Figure 2) that is over 3000 years old, about which historians say the following: Basically, this is the "material manifestation" (a representation) of the contexts of a religious imagination. The stele represents a religious concept that was new at the time, namely a God who hears and answers to prayers (see the ears on the right side of the picture), with the king acting as "mediator". In addition, social conditions are described: In the group of adorants the stele founder comes first, then his wife, then their children, ranked by age (importance). Clearly, the stele represents a model of abstract and concrete originals (ch 1). It's concern and usage is communicating the new religious concept to viewers (ch 2) with the purpose (ch 3) that these understand the concept. The focus (ch 5) is on the idea of a listening God, the representation is graphical with some symbols referencing concepts. We interpret this stele as the representation of a hierarchy H^1 level model, as at least God and king may have various extensions.

[11] A *Marked Petri Net* is a quintuple (P,T,I,O,M), where (P,T,I,O) is a bipartite graph with disjoint node sets P and T and two relations $I,O \subseteq P \times T$ such that $0 < |P \cup T| < \infty$ and $(P \cup T, I \cup O)$ is a connected graph; $M: P \rightarrow N_0$ is a mapping called *marking* (N_0 denoting the set of natural numbers including 0).

If this model is a conceptual one, cannot be said with a hundred per cent certainty. For there is no explicitly assigned concept space with corresponding concept relations handed down with. However, it is not impossible that artist and viewers were aware of such a concept space at the time.

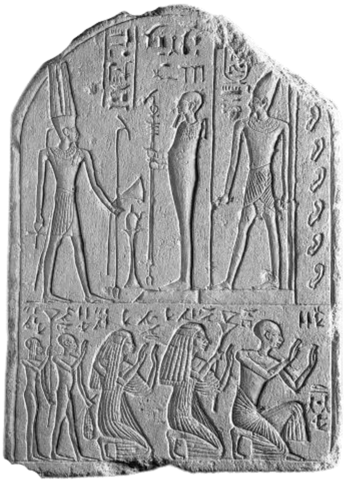


Figure 2: Stele of Seth-er-neheh, Roemer- und Pelizaeus-Museum, Hildesheim

For our last example, think of a musical score. This is a linguistic representation of the (complex) model created by a composer using a common musical language. Members of the musical community that have agreed on the meaning of terms or patterns and their combination of the musical language can infer from the score elements and their sequence to the transported concepts. For instance, think of "The Art of the Fugue" by Johann Sebastian Bach. In this case, the score represents a conceptual model. This model in turn may have many different extensions (created, e.g. through performances).

To sum up: the existence of a concept space is a precondition for models being conceptual ones; the degree to which a component of a conceptual model has UoD related, interpretable semantics depends on its associated vocabulary elements. For comparison, consider the difference between WEB 1.0 and WEB 3.0 („Semantic Web“ [W3C]): In WEB 1.0 we can interpret website content based on the natural language terms used exploiting NL's a priori knowledge. WEB 3.0 pages are intended to provide a vocabulary defining the semantics of the page content; consequently WEB 3.0 pages are representations of conceptual models (mainly on hierarchy level H^0), the concept space being defined, e.g., exploiting schema.org [Sch19].

5 The Triptych: Dimensions of Conceptual Modeling

Our considerations as presented so far have inspired us to create a paradigm for conceptual modeling, namely the triptych[12]: For with this paradigm the transition from the linguistic description of phenomena to modeling and then to conceptual modeling can be described vividly through the successive opening of wings. In this section, we first explain the paradigm and then go into detail about the three dimensions that we attribute to conceptual modeling with this paradigm.

[12] A triptych is a piece of art made of three (panel) paintings connected to each other in a way that allows the two outer ones to fold in towards the larger central one (see https://dictionary.cambridge.org). I.e., when folded, the inner panel is not visible.

5.1 *The Paradigm*

The Closed Triptych: The intuitive perspective

Let's start with the closed triptych as depicted in Figure 4[13] : we see the backs of the two outer wings, which in this state cover the middle tableau. Let's associate this situation with the everyday situation of dealing with information without explicit conceptualization and modeling**:** humans reason on the basis of their observations on the perceivable world due to their senses, feelings and beliefs. They build their mental worlds based on their perceptions which typically differ. They live in their social worlds with their agreements. On the other side, humans use a variety of languages as an instrument for narrative representations. The "*enabling language tableau*" on the right hand side shows us, that we can use very different languages. The "*sensing, mental and social tableau*" on the left hand side symbolizes the diversity of aspects and things that can be grasped and communicated through language: (i) observations, (ii) beliefs, perspectives, trust, and cognition, and (iii) agreements.

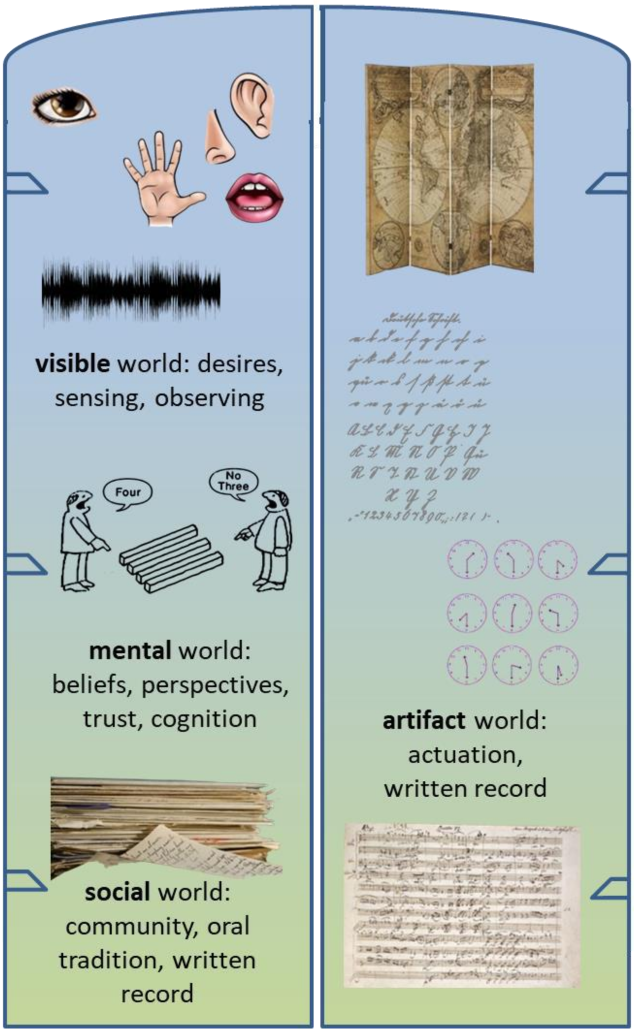

visible world: desires, sensing, observing
Four
No Three
mental world: beliefs, perspectives, trust, cognition
social world: community, oral tradition, written record
artifact world: actuation, written record

Figure 3: The two tableaus of the closed triptych: Languages (right) enabling the narrative representation of observations, mental reasoning and social agreement (left)

[13] We are well aware of the fact that we cannot match the artistic skills of the painters of real triptychs.

The Triptych with it's right wing opened: The model perspective

When we open the right wing, we see its front side and at the same time the right half of the previously hidden middle tableau as is depicted in Figure 4. This opens the way to modeling: on the now visible part of the middle tableau we see models of different levels of abstraction (meta^n-models and their extensions down to the (lowest) meta^0-level, the instance level). Their origins are the elements on the backside of the closed left (sensing, mental and social) tableau. The message of the right tableau remains the same, except that we are now dealing not only with natural languages and traditional symbolic languages but also with modeling languages or model representation languages. The choice of language is a matter of preferences, education, and practices within the community of practice. Usually, ortho-normalized languages are used for this purpose, such as an entity-relationship language based on a common language foundation.

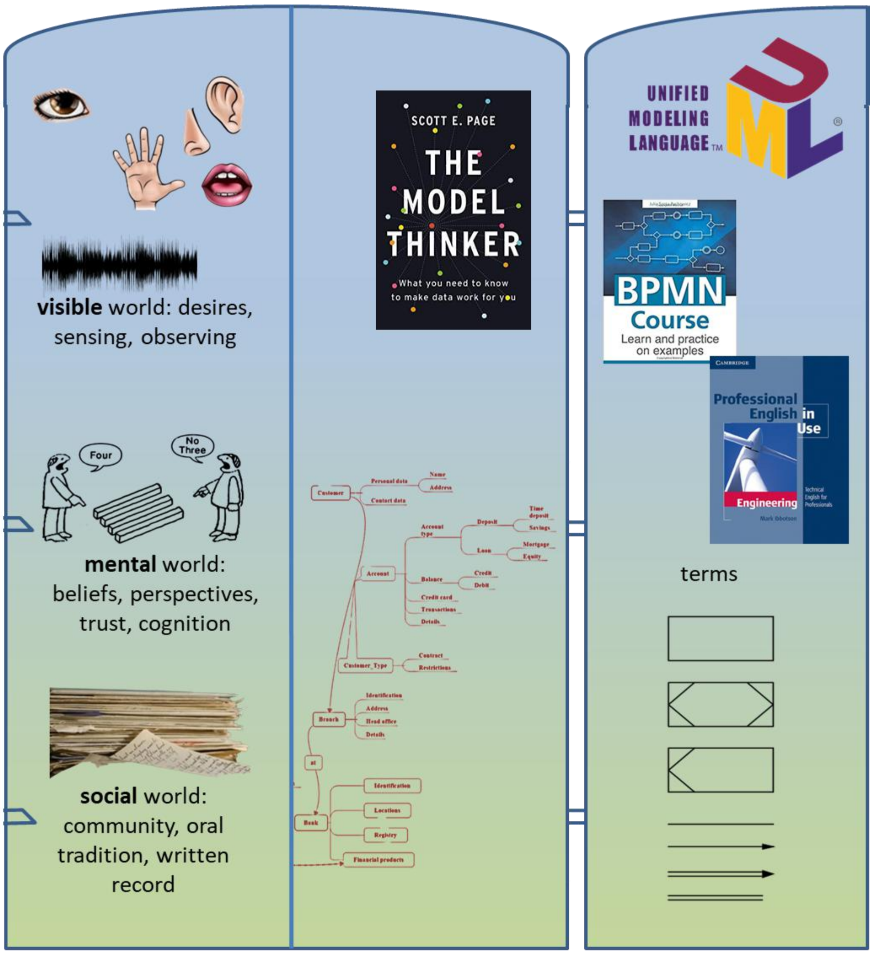


Figure 4: The triptych with its right wing opened and left wing closed

The Triptych with both wings opened: The conceptual model perspective

Opening the left wing makes the world of conceptual modeling shine in all its beauty, as the left tableau now shows us a concept space whose elements are assigned to the models on the now fully opened middle tableau (see Figure 5). The concept space brings order and structure to the world of observations, beliefs, agreements, etc., which we know are located on the back of the left wing. It may be organized by ontologies, thesauri or other kind of encyclopedias supporting conceptualization. We call this tableau the "encyclopedic tableau". It allows us to define the semantics and pragmatics of conceptual models and to relate the models to the human world.

Within this setting, the linguistic tableau supports conceptualization based on terms. We note that, apart from conceptual modeling, modeling does not need the encyclopedic tableau.

The triptych paradigm illustrates that conceptual modeling has three essential dimensions:

- The linguistic dimension: Conceptual modeling is made possible by a language that is generally accepted in a community of practice and that is semantically based on the perception and understanding of the members of the community of practice;
- The „encyclopedic” dimension: Conceptual models codify notions from the “user’s encyclopedia” and express those through linguistic terms.
- The model dimension: Conceptual modeling connects the two other dimensions.

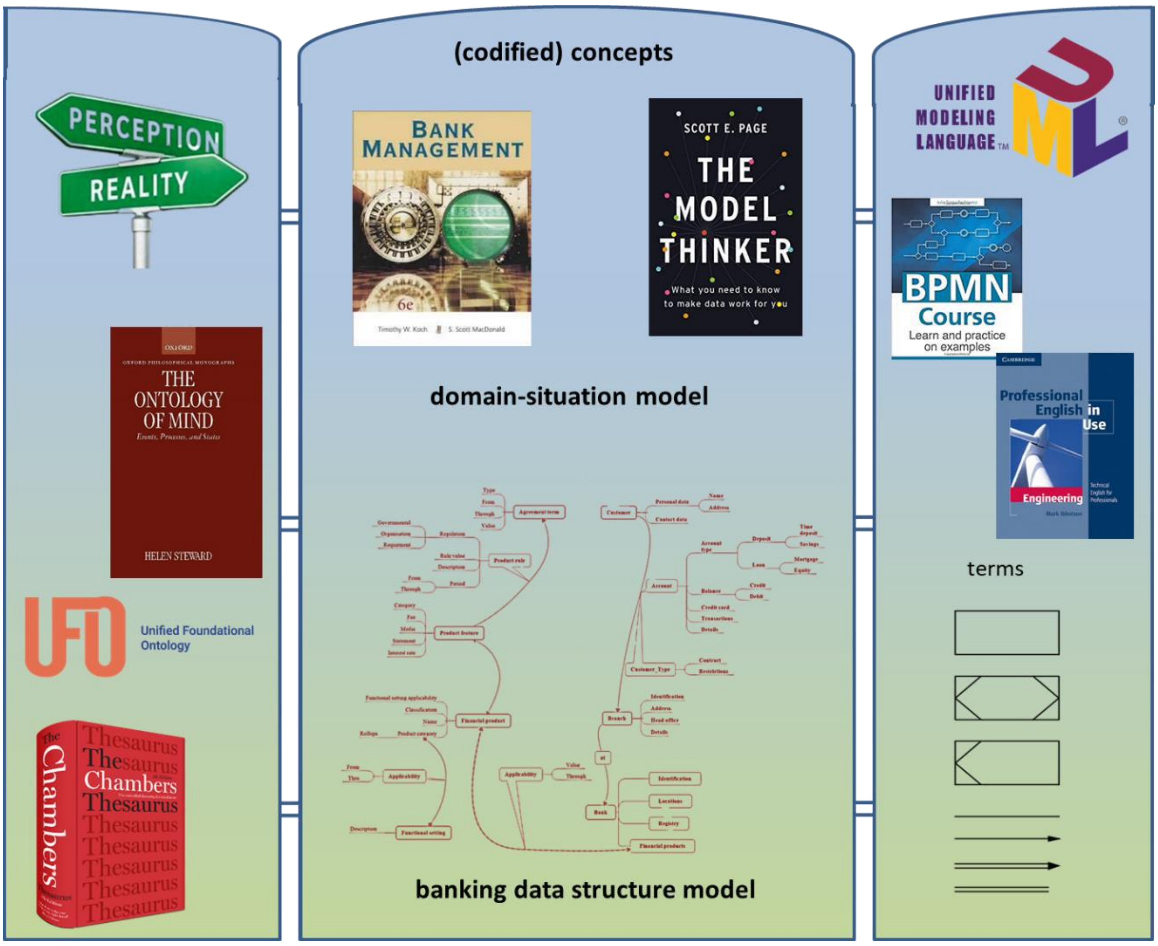

(codified) concepts
PERCEPTION
REALITY
BANK MANAGEMENT
SCOTT E. PAGE
THE MODEL THINKER
What you need to know to make data work for you
UNIFIED MODELING LANGUAGE
BPMN Course
Learn and practice on examples
THE ONTOLOGY OF MIND
HELEN STEWARD
domain-situation model
Professional English in Use
Engineering
UFO
Unified Foundational Ontology
terms
Chambers Thesaurus
banking data structure model

Figure 5: The open triptych
Left: The encyclopedic dimension for grounding models and their representation languages in concepts
Center: The conceptual model dimension
Right: The language dimension

In other words: we base our consideration of conceptual models on a separation of (i) language, (ii) knowledge, personal perception, and (iii) modeling as a separate activity. This separation allows us to distinguish between terms from certain languages and notions used for expressing perceptions or knowledge in the encyclopedic tableau and for enhancing models by concepts. It is thus the key to our distinction between models and conceptual models.

In the following subsections 5.2-5.4, we discuss the three dimensions in detail.

5.2 *The Linguistic Dimension: The Term Space*

All forms of communication take place by means of terms[14], which are exchanged via a carrier medium (apart from metaphysical phenomena). Terms can be images or image sequences, sounds or tone sequences, texts or elements of a formal language, etc. and of course mixed forms. Term sets often consist of a set of basic forms (literals) from which more complex terms can be constructed based on grown or defined composition rules, i.e. a grammar. Think, for example, of natural languages, in which words, phrases, sentences and texts are formed from the letters of the respective alphabet and some special characters. Or think of the score of a classical symphony: it consists of notes and special characters arranged along staves. The same applies to any formal language, such as the characters used in graph theory or UML.

What all languages have in common is that they can only be properly used as a means of communication if

- the communication partners know the grammar, i.e. the literals and the composition rules: a person who can't read notes can't do anything with a score except perhaps admire it because it is calligraphically appealing;
- the terms used for content description have a relative similar meaning for all partners;
- the information content transported by a term is conscious and intended by the sender and can be accessed by the receiver [vF72, FGH92].

Communication thus requires that a community willing to communicate explicitly or implicitly agrees on the literals and character set rules used as well as on the assignment of meaning to terms. For example, with the first language acquisition of our mother tongue we implicitly accept it as a means of communication and successively learn the available terms, how they are composed and what meaning is usually attributed to them. If such a means of communication comprises definitional elements, it can be used to create new language elements (e.g. new literals, new rules) and to define or propose their meaning, so that an agreement process can take place in the community: This corresponds to Gruber's original definition of ontology development ("shared conceptualization"). But we do not need full agreement.

If we now consider the model characteristics discussed in section 2, we can conclude that the essence of communication is the exchange of models based on terms. If there are rules about the composition and permissibility of the terms used, and if the represented models are associated with elements in the encyclopedic tableau, we speak of a controlled vocabulary.

5.3 *The Encyclopedic Dimension: The Notion Space*

People form a certain consolidated understanding of the world on the basis of their own cognition. Cognitive Scientists speak of a '*cognitive structure*'[15] that is created by '*cognitive processes*' [Kol07, vF03]: observation and perception, and activities of thinking like comparison, reflection, idealization, context expansion, abstraction, and separation. Consequently, the main ingredients of a person's cognitive structure are ideas[16] that are usually strongly interlinked.

[14] As announced at the beginning of section 2, we only use the word term instead of "sign" to avoid misunderstandings.-

[15] Note that the term "cognitive structure" is used in the literature with different meanings. We use it here to denote the outcome of cognitive processes but not the structure of the processes. [Kol07] calls this interpretation "conceptual structure".

[16] We use 'idea' here in the sense of 'conception', which best relates to the German word 'Vorstellung' [Bol37, Bre74, Twa94], respectively to 'mental concepts' as used in [Kol07].-

The first level of the encyclopedic dimension is thus the cognitive structure of a person. In order to communicate about it, ideas and their connections must be represented and conveyed by terms of any language. A person who perceives such terms then interprets them according to her/his personal cognitive structure. This makes it clear that with this (traditional) form of communication, a complete agreement is not possible and cannot be proven.

However, the situation can be improved by externalizing the encyclopedic dimension and formalizing it in the form of explicit thesauri, lexicons, or ontologies. Explicating the encyclopedic dimension corresponds to the opening of the left wing of our triptych. For us, therefore, this externalization is the moment when ideas become concepts. I.e. we can now sharpen our understanding of "concept" and "notion" from section 2 as follows:

1. A *concept* (in the encyclopedic dimension) is a mental construct
    - that is formed by combining characteristics of general or abstract ideas gained by cognitive processes (see section 2) and
    - that is externalized and explicated in an encyclopedic structure.[17]
2. Consequently, also a *notion* (being a general inclusive concept) is externalized and explicated in an encyclopedic structure.

If communication partners agree on the common use of such encyclopedic structures, for example a shared [Gru93] ontology, the probability of communication free of misunderstandings increases. However, it is of course still not possible to prove that the mutual understanding is identical.

5.4 The Conceptual Model Dimension: The Link between Term and Notion Spaces

Usually, a concept space is specific to a certain area of application and is based on an understanding of the perceptions of things and coherences in that area. The utilization, exploration and application of concepts depend on the user and her/his community of practice (e.g. users' education profile), usage and context.

Based on what has been said so far, we can now formulate somewhat sloppily: A conceptual model selects, uses, reconsiders, orders, and integrates parts of a notion and a term space and thus establishes a structured view on that notion space (analogously e.g. to views on databases).

Composition and structure of such a view correspond (according to CM7 and CM8, see section 3) to the given conceptual model, which in turn is an extension of a certain metamodel (more precisely: of the metamodel elements and relationships) like the ER Model. The relationships may satisfy a number of axioms and lead to poly-hierarchically ordered concept structures, typically with layers. Such poly-hierarchical structures arise in particular if the metamodel in question contains abstraction relations like generalization/specialization, aggregation/decomposition, clustering, and intension/extension, which can be instantiated on the model level.

The view, in turn, consolidates the meaning of the elements of the model and determines the linguistic meaning of terms (designators and annotations) which is an inherent but hidden aspect of the concept space.

Even though we have declared at the beginning not to present a (new) definition of "conceptual model", we can't hold back from formulating our understanding of it here: *A conceptual model is a*

[17] Note that for externalization, a concept is assigned one term or a construct of terms.

concise and purposeful consolidation of a set of concepts that are presented by means of terms in a predefined linguistic format. As such it establishes a view of a given notion space.

Let's explain all this using a very simple example, in which – for further simplification - we use a graphical grammar as known from UML for representation purposes on all levels except the lowest one, and have omitted an explicit definition of roles and multiplicities:

a) For creating a model, first of all, we have to define which modeling elements we want to use to build that model. This is done with the help of a metamodel. However, for metamodeling we need modeling elements again, so we have to create these on a Metameta level, i.e. hierarchy level H^3.
b) Figure 6 shows such very simple Metametamodel: it provides us with the possibility to define (in a metamodel) model elements that may linked by IS-A, part-of and relation connectors.
c) Figure 7 shows a metamodel that is an extension of the metametamodel of Figure 6: it introduces the modeling elements Class, Association, IS-A, Attribute and Type as well as some relationships between these elements.
d) Figure 8 shows a model that refers to the retail sector as an application area. It is an extension of the metamodel given in Figure 7. The model introduces classes Person and Organization, defines these as specializations (IS-A) of class Client which is associated with class Article in an m:n relationship. The classes have some attributes that come with type specifications for their extensions.
e) Figure 9 shows some extensions of the model given in Figure 8 on the next lower level which is usually called object or data level. Here we used a self-explanatory text-oriented grammar for representation.

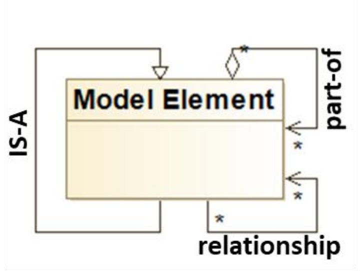


Figure 6: A simple metametamodel[18]

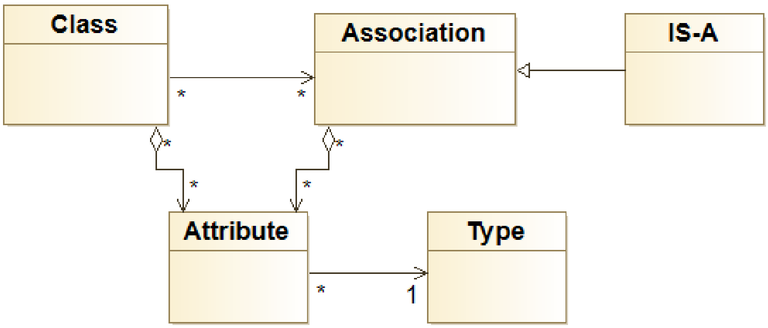


Figure 7: A metamodel created as an extension of the metametamodel given in Fig. 6

[18] Figures 6-8 were created using the Modelio tool. https://www.modelio.org/

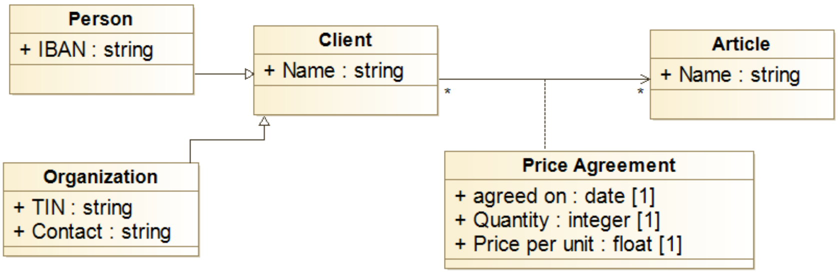


Figure 8: A Model created as an extension of the metamodel given in Fig. 7

```
Person(<Name: Frank Muller>, <IBAN: DE99 0909 9090 0909 9090
09>)

Organization(<Name:  Buyers  Ltd.>,  <TIN:  ATU99999999>,
<Contact: Frank Bourbaki>)

Article(<Name: Mouth and nose protection mask>)

Price-Agreement(<client:  Frank  Muller><article:  Mouth  and
nose protection mask>, <agreed on: 05.08.2020>, <Quantity:
10>, <Price per unit: 2,50>)

Price-Agreement(<client:  Buyers  Ltd.><article:  Mouth  and
nose protection mask>, <agreed on: 02.09.2020>, <Quantity:
1000>, <Price per unit: 0,82>)
```

Figure 9: Some extensions of the model given in Fig. 8

Please note that nothing we have presented so far in our example can be called a conceptual model. Rather, we find ourselves - metaphorically speaking - in front of the triptych with open right and closed left wing. In other words, we are dealing with models

- that are structured,
- are in intension/extension relationships, and
- have a certain intuitive meaning since we have used words from natural language and from the environment of UML.

However, the use of words, symbols and structuring mechanisms that we know from conceptual modeling does not automatically lead to the creation of conceptual models. E.g., drawing an UML class diagram is not conceptual modeling per se.

To make the (meta-)models of Figure 6-9 conceptual ones, we therefore have to associate concepts with each of their components. I.e., we need a notion space that explicates and explains the meaning of all elements (from "Model Element" down to "Price Agreement") on all levels including the connections/relationships.

It then becomes clear, that for instance if we associate with the components of the model given in Figure 8 notions commonly used in the retail sector, this model defines a specific view on this sector in terms of its structure and the selection of what is considered relevant. We can also say that the conceptual model "codifies" the respective concepts of the application domain.

5.5 Model and Language Hierarchies

Model hierarchies are based on the duality of intension and extension [BMS86, Kan15] and thus reflect levels of abstraction. They are well known through the considerations of Information Resource Dictionary systems [ISO90] or the MetaObject Facility [OMG].

We therefore do not want to go into further details of model hierarchies in this paper. On the other hand, the model representation languages to be defined for this purpose deserve a closer look. They have to provide suitable syntactic artifacts to represent the semantic artifacts (the models).

These representation languages in turn form a hierarchy, which, however, is not isomorphic to the model hierarchy. Rather, we distinguish three levels as shown in Figure 10 [MMR17], [MMS18]:

(1) Grammar definition level (top level): contains the means of defining the language grammars. In our research, we use a specific version of the extended Backus Naur Form EBMF, compatible with the ANTLR grammar definition language [Par13].

(2) Language definition level: defines grammars for the representation languages (RL) related to the (possibly domain specific) modeling languages under consideration: meta-metamodel RLs, metamodel RLs, model RLs and instance/data RLs.

(3) Language usage level: representations of the models of all levels. For example, it is possible to use OWL 2 as a representation language this level.

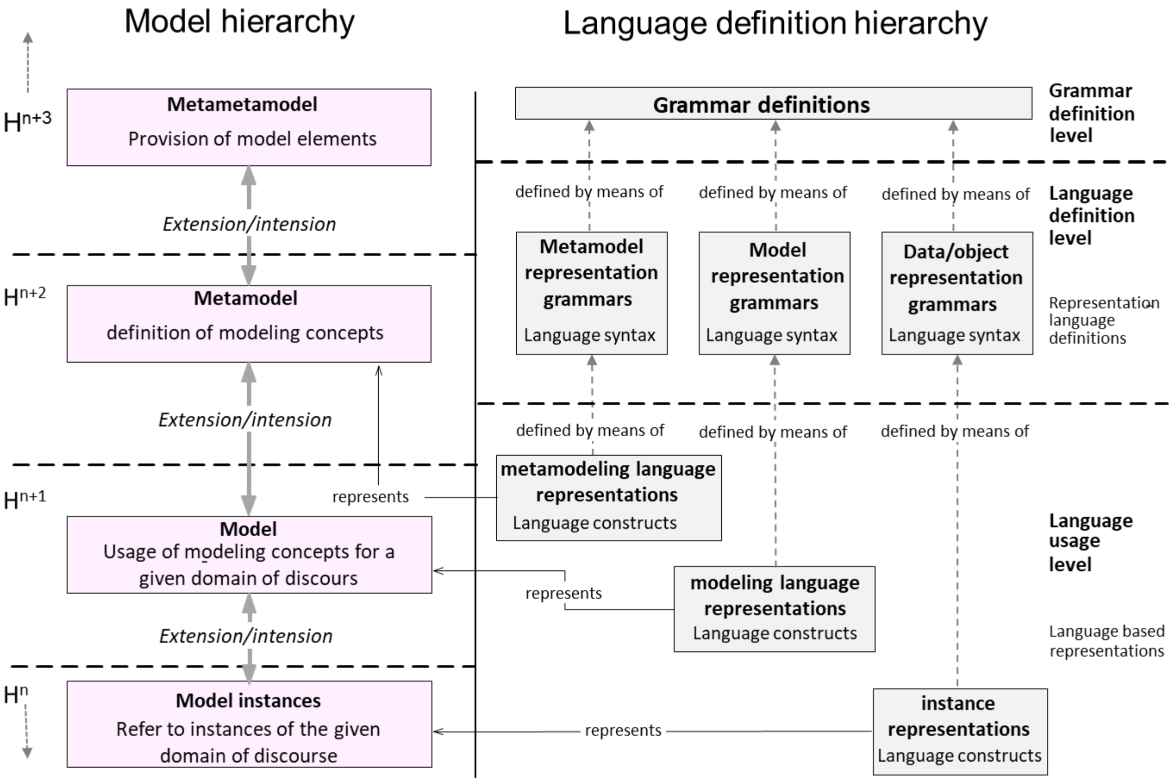


Figure 10: Language hierarchy and its connections to the model hierarchy (A^i: Abstraction Level i)

6 Conclusion and Future Research

Modeling is as old as human advanced civilisation. The bible remembers models already in the Chronicles (1, 28, 11)[19] or Torah/Genesis (2 Moses, 25, 9 or 1 Moses, 1, 27). Around the same time, Heraclitus developed the tenet of logos that consists of concepts.

Conceptual modeling is one of the kernel activities in information systems engineering. For instance, conceptual schemata are widely used since the advent of database technology and explicitly named as such with the introduction of the entity-relationship modeling language. The first model we know that might be called a conceptual one relates back to the myth of Ptah who builds the world according to his worldview and doctrine. As far as we know for sciences, simulation explicitly uses the term "conceptual model" since 1950 [RAB15]. Other disciplines use the wording "conceptual model" with quite different meaning, see for instance [SNW13].

Today, the term "conceptual model" is widely used and needs a proper systematisation. We start this systematisation with eight characteristics for a signature of conceptual model. The first six characteristics (related to origins; concern and usage; purpose and function; domain and context; focus; representation) already belong to the signature of models. These characteristics are extended for conceptual models by two additional characteristics (concept space; concept relationship). The second and the third characteristics are extended for conceptual models.

We differentiate between notions as general inclusive concepts, terms as representations referring to concepts, and concepts in the narrow sense as codified abstract ideas. In systems engineering, concepts are those in the narrow sense. The histories of notions, concepts, and terms are different. Notions have been used as reasoning instruments. Terms are bound to languages. Concepts have a history of at least 3-4000 years. The separation into notions, concepts, and terms can be used for a proper introduction of a paradigm of conceptual modeling: the triptych that consists of three wings which represent

(1) the notion or "encyclopaedic" dimension as the supporting foundation of concepts,

(2) the term or linguistic dimension as the enabler for model specification, and

(3) the concept and model dimension as the result of modeling.

The outer two wings can be used without the third one. They form then a closed triptych, i.e. a diptych. The middle part of the triptych – the model dimension – is supported by the "encyclopaedic" dimension and is enabled by the linguistic dimension.

Modeling has been systematised by abstraction levels. The separation by abstraction is typical for artificial languages. The ground H^n level represents things of interest. The H^{n+1} level is used for models; the H^{n+2} level for metamodels (i.e. essentially the structure of the modeling approach), and the H^{n+3} for metametamodels (i.e. essentially the framework of the modeling approach). This strict separation by abstraction is blurred in the linguistic dimension and almost not existing in the "encyclopaedic" dimension. Conceptual modeling thus supports defining properly structured views on the encyclopaedic dimension.

We did not plan to propose a new definition of the term "conceptual model", especially as more than 60 such definitions exist to our knowledge. The introduced signature, however, together with the Triptych paradigm provides a means and explanation of the essence of conceptual modeling.

[19] The Zwingli bible translation explicitly uses the word „Modell", whereas the Luther translation uses "Vorbild" (antetype, archetype).

We understand this as a first step towards a general theory of conceptual modeling. We encounter a good number of problems to be solved in forthcoming research. The list below is ordered according to our plans for the future; collaboration and contribution from the community are more than welcome:

(1) Modeling is based on abstraction, modularity, and other *modeling principles.* The selection of promising and useful principles is still art that compromises between model capacity, cognitive economy for the community of practice, and inferential utility. We need a proper systematisation of principles. Abstraction goes beyond structural abstraction (e.g. [SS77]) and considers advanced hierarchies beyond IsA relationships.

(2) Conceptual model characteristic 7 relates models to their *concept space*. The theory of concept spaces distinguishes concepts, notions, and terms. The corresponding spaces need a deeper exploration. Classical intension-extension Galois lattices are too strict for terms and notions.

(3) Conceptual models use *languages* as enablers. In natural languages, words or terms have their language specific semantic (or word) fields, i.e., a lexical set of words that share a common semantic property [Bri00]. These fields can be different for different languages. Conceptual modeling is not bound to a singleton language. Multi-language modeling can be based on synset approaches commonly used for WordNet.

(4) Similar to *generic models* which allow specialisation of models to more appropriate ones, concept spaces can also be based on generic concept spaces with a specialization theory that allows to adapt the concept space to a specific application, context, and community of practice.

(5) Conceptual model characteristic 8 is based on *concept relationship*. There may be various types of relationship such as one-to-one or many-to-one. Concept spaces are typically structured. The impact of this variability is an open issue.

(6) Modeling is steered by the *purpose* and *function* of the model. We know so far a good variety of model functions in different scenarios where models are used on purpose. Functions can be categorised. This categorisation can be used for categorisation of conceptual models and for stereotyping of models. These stereotypes have then a common grounding and basis that is inherited by most models of such a stereotype.

(7) We typically use a number of models of the same origins in a coherent manner. Some models are derived from other models in such *model ensemble*. Conceptual model transformation, model coexistence, and model coevolution need a theoretical underpinning.

(8) The *focus* of a conceptual model is based on the directed and concentrated attention that is steered from one side by the model's function and purpose and from the other side by the potential and capacity of the encyclopaedic support and enabling language. The impact of these governing dimensions need a proper exploration.

(9) We considered so far the four most important *concerns and usages.* There are further concerns and usages which result in different kinds of conceptual models, specific quality requirements to conceptual models, and specific variability of the model. A model is also serving a weighted overlaying combination of concerns and usages.

(10) Conceptual models do not reflect all potential *origins* of a given universe of discourse but only most likely or most typical ones. Whether the selection of such set of origins is the

most appropriate *for a given modeling target* is a difficult question. Models can also be origins of models, e.g. mental models and domain models. The plasticity and stability of a model against the selection of origins is a difficult research issue.

(11) Applications, infrastructures, origins, and user communities continuously evolve. *Evolution* of models needs a proper modernisation strategy, evolution tactics, and a realisation approach including handling of heritage (legacy) models. Models will become adaptable and self-adapting.

(12) The *context* of (conceptual) models and of (conceptual) modeling includes aspects of time, disciplines, (thought) schools, applications, experience, education, and in general of cultures. Models differ in dependence on this context. We need powerful transformation techniques that allow to become partially context-independent.

(13) Can recommendations for the development of domain-specific conceptual modeling methods be derived from all this?

This list is far from being complete. It demonstrates, however, the potential of the signature approach by systematic treatment of open issues in (conceptual) model research.

References

[BG18] T. Brand and H. Giese. *Towards software architecture runtime models for continuous adaptive monitoring*. In *Proc. MODELS 2018 Workshops,* vol. 2245 of *CEUR Workshop Proceedings*, pp. 72-77, 2018.

[Bri00] L.J. Brinton. *The structure of modern English: a linguistic introduction*. John Benjamins Publishing Company, 2000.

[BMS86] M. Brodie, J. Mylopoulos, and J.W. Schmidt (eds.). *On conceptual modeling*. Springer, Heidelberg, 1986.

[Bol37] B. Bolzano. *Wissenschaftslehre: Versuch einer ausführlichen und größtenteils neunen Darstellung der Logik*. Sulzbach: Seidel, 1837.

[Bre74] F. Brentano. *Psychologie vom empirischen Standpunkte*. Leipzig: Dunker & Humblot, 1874.

[Che76] P.P. Chen. *The entity-relationship model: Toward a unified view of data*. *ACM TODS*, vol. 1, no. 1, pp. 9-36, 1976.

[DLPS18] L. M. L. Delcambre, S. W. Liddle, O. Pastor, and V. C. Storey. *A reference framework for conceptual modeling*. In *Proc. ER 2018*, volume 11157 of *Lecture Notes in Computer Science*, pp. 27-42. Springer, 2018.

[FGH92] H. von Foerster, E. von Glasersfeld, and P. Hejl. *Einführung in den Konstruktivismus*. Piper, München, 1992.

[FR07] R. France and B. Rumpe. *Model-driven development of complex software: A research roadmap. Future of Software Engineering*, 2007, 37-54, IEEE Computer Society.

[GGM20] N. Guarino, G. Guizzardi, and J. Mylopoulos. *On philosophical foundations of conceptual models*. In *Information Modeling and Knowledge Bases XXXI*, *Frontiers in Artificial Intelligence and Applications*, forthcoming. IOS Press, 2020.

[GR19] J. Gray and B. Rumpe. *Models as the subject of research*. *Software and Systems Modeling*, 18(6): pp. 3189-3191, 2019.

[Gru93] T.R. Gruber. *Toward Principles for the Design of Ontologies Used for Knowledge Sharing*. International Journal Human-Computer Studies 43, pp. 907-928, 1993.

[ISO90] ISO/IEC 10027:1990(en). *Information technology — Information Resource Dictionary System (IRDS) framework*, 1990.

[Kan15] H. Kangassalo. *Definitional conceptual schemata – The core for thinking, learning, and communication. Keynote given at 25th EJC Conference*, Maribor, Slovenia, June, 2017.

[Kan81] I. Kant. *Kritik der reinen Vernunft*. Verlag von Johann Friedrich Hartknoch, Riga 1781/87.

[KB71] G. Klaus and M. Buhr (eds). Philosophisches Wörterbuch. Bibliographisches Institut, Leipzig 1971.

[KM20] KEA-Mod: *Kompetenzorientiertes E-Assessment für die grafische Modellierung – Projektvorstellung*. http://butler.aifb.kit.edu/MoHoL2020/KEA-Mod.pdf (accessed August 3rd, 2020)

[Kol07] M.E. Koltko-Rivera. *What Are Cognitive Structures? Are Worldviews Cognitive Structures?* Proc. 115th-Annual Convention of the American Psychological Association, San Francisco, 2007.

[KWB03] A.G. Kleppe, J.B. Warmer, W. Bast. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Longman Publishing Co., Inc. (2003).

[Lat17] C. Lattmann. *Vom Dreieck zu Pyramiden - Mathematische Modellierung bei Platon zwischen Thales und Euklid*. Habilitationsschrift, CAU Kiel, 2017.

[LM78] P.C. Lockemann and H. C. Mayr. *Rechnergestützte Informationssysteme*. Springer, 1978.

[LMN93] P.C. Lockemann, G. Moerkotte, A. Neufeld, K. Radermacher, and N. Runge. *Database design with user-definable modeling concepts*. *Data Knowledge Eng*ineering, 10, pp. 229-257, 1993.

[LMW79] P. C. Lockemann, H. C. Mayr, W. H. Weil, and W. H. Wohlleber. *Data Abstractions for Database Systems*. *ACM Trans. Database Syst.,* vol.-4, No. 1, pp. 60-75, 1979.

[Mah05] B. Mahr, Was ist ein Modell. *Der Modellbegriff in Natur-und Ingenieurwissenschaften"*. TU Berlin Summer term 2005, http://pdv.cs.tu-berlin.de/gk-magsi/RV05Mahr.pdf, accessed on August 3rd, 2020.

[Mah08] B. Mahr. *Cargo. Zum Verhältnis von Bild und Modell*. In *Visuelle Modelle,* 17--40. Wilhelm Fink Verlag, München, 2008.

[Mah15] B. Mahr. *Modelle und ihre Befragbarkeit - Grundlagen einer allgemeinen Modelltheorie. Erwägen-Wissen-Ethik (EWE),* Vol. 26, Issue 3, pp. 329-342, 2015.

[MMR17] H. C. Mayr, J. Michael, S. Ranasinghe, V.A. Shekhovtsov and C. Steinberger. *Model Centered Architecture*. In: Cabot J., Gómez C., Pastor O., Sancho M., Teniente E. (eds) *Conceptual Modeling Perspectives*, pp. 85-104, Springer, 2017. https://doi.org/10.1007/978-3-319-67271-7_7

[MMS18] H. C. Mayr, J. Michael, V. A. Shekhovtsov , S. Ranasinghe and C.-Steinberger. *A Model Centered Perspective on Software-Intensive Systems.* In Proc. 9th Int. Workshop on Enterprise Modeling and Information Systems Architectures, CEUR Vol. 2097, pp. 58-64, 2018.

[MBF90] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. *Wordnet: An on-line lexical database*. *International Journal of Lexicography*, 3, pp. 235-244, 1990.

[Mil91] G.A. Miller. *The science of words*. Scientific American Library, 1991.

[MM13] J. Michael and H. C. Mayr. *Conceptual modeling for ambient assistance*. In *Proc. ER 2013*, LNCS Vol. 8217, pp. 403-413. Springer, 2013.

[MM16] J. Michael and H. C. Mayr. *The process of creating a domain specific modeling method* (extended abstract). In *Proc. {EMISA} 2016*, vol. 1701 of *CEUR* Workshop Proceedings, pp. 40-43. CEUR-WS.org, 2016.

[Mur01] G. L. Murphy. *The big book of concep*ts. MIT Press, 2001.

[Myl20] J. Mylopoulos. *Philosophical Foundations of Conceptual Modeling: What is a Conceptual Model*? ER Online Summer Seminars. https://eross2020.inf.unibz.it/wp-content/uploads/2020/07/UniBZ-John.pdf, accessed August 3rd, 2020.

-[OMG] MetaObject Facility Specification. https://www.omg.org/mof/, accessed August 10th, 2020.

[Par13] T. Parr. *The Definitive ANTLR 4 Reference*. The Pragmatic Bookshelf, 2013.

[Pet62] C. A. Petri. *Fundamentals of a theory of asynchronous information flow*. In *Proc. 2nd {IFIP Congress 1962*, pp. 386-390. North-Holland, 1962.

[Pla03] R. Platon. *Platonis rempublicam recognovit brevique adnotatione critica instruxit*. Oxford 2003.

[PP19] D. Plexousakis and T. Patkos. *Modeling knowledge action and time: Action theories and their application in dynamic domains*. *NEMO Summer School 2019*, University of Vienna, Faculty of Computer Science, Vienna, 2019.

[PR18] O. Pastor and M. Ruiz. *From requirements to code: A conceptual model-based approach for automating the software production process. EMISA International Journal on Conceptual Modeling*, pp 274-280, 2018.

[RAB15] S. Robinson, G. Arbez, L.G. Birta, A. Tolk and G. Wagner. Conceptual modeling: Definition, purpose and benefits. Proc. of the 2015 Winter Simulation School, IEEE, pp. 2812-2826, 2015.

[Sch19] W3C *Schema.Org Community Group*. Welcome to schema.org. https://schema.org/, 2019.

[SK03] S. Sendall and W. Kozaczynski. *Model transformation: the heart and soul of model-driven software development*. IEEE Software, vol. 20, no. 5, pp. 42-45, Sept.-Oct. 2003, doi: 10.1109/MS.2003.1231150.

[SNW13] J. Shreffler-Grant, E. Nichols, C. Weinert*. The Montana State University Conceptual Model of Complementary and Alternative Medicine Health Literacy. Available from: https://www.researchgate.net/publication/253331927_The_Montana_State_University_Conceptual_Model_of_Complementary_and_Alternative_Medicine_Health_Literacy [accessed September 3, 2020].*

[SS77] J. M. Smith and D. C. P. Smith. *Database abstractions: Aggregation and generalization*. *ACM TODS* 2(2), pp. 105-133, 1977.

[Sta73] H. Stachowiak. *Allgemeine Modelltheorie*. Springer, Berlin, 1973.

[Tha18] B. Thalheim. *Conceptual model notions - a matter of controversy; conceptual modeling and its lacunas*. *EMISA International Journal on Conceptual Modeling*, pp. 9-27, 2018.

[Tha19] B. Thalheim. *Conceptual modeling foundations: The notion of a model in conceptual modeling*. In *Encyclopedia of Database Systems*. Springer US, 2019.

[TN15a] B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*. De Gruyter, Boston, 2015.

[TN15b] B. Thalheim and I. Nissen. *Ein neuer Modellbegriff*. In *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*, pp. 491-548. De Gruyter, Boston, 2015.

[Twa94] K. Twardowski. *Zur Lehre vom Inhalt und Gegenstand der Vorstellungen: Eine psychologische Untersuchung*. Wien: Hölder, 1894.

[vF72] H. von Foerster. *Perception of the future and future of perception*. *Instructional Science*, vol. 1, no. 1, pp. 31-43, 1972.

[vF91] H. von Foerster. *Zur Konstruktion der Wirklichkeit. Die erfundene Wirklichkeit. Wie wissen wir was wir zu wissen glauben*, pp. 39-60, 1991.

[vF03] H. von Foerster. *Understanding Understanding: Essays on Cybernetics and Cognition*. Springer-Verlag New York, Inc., 2003.

[W3C] W3C. *Semantic Web*. https://www.w3.org/standards/semanticweb/; accessed on August 13, 2020.

[We20] Merrian-Webster Dictionary, https://www.merriam-webster.com/dictionary/; accessed on August 3rd, 2020

[Wik17] Wikiquote. *Conceptual model*. https://en.wikiquote.org/wiki/Conceptual_model, 2017. Accessed Nov. 21, 2017.

[Wol96] G. Wolters. *Modell.* In J. Mittelstraß, editor, *Enzyklopädie Philosophie und Wissenschaftstheorie*, Vol. 2, pp. 911-913, J.B. Metzler, Mannheim,1995.

EMISA International Journal on Conceptual Modeling, February 2018, pp. 9-27

Conceptual Model Notions - A Matter of Controversy

Conceptual Modelling and its Lacunas

Bernhard Thalheim
Christian-Albrechts University at Kiel, Department of Computer Science, D-24098 Kiel

January 12, 2022

Abstract

The conception of a conceptual model is differently defined in Computer Science and Engineering as well as in other sciences. There is no common notion of this conception yet. The same is valid for the understanding of the notion of model. One notion is: *A model is a well-formed, adequate, and dependable instrument that represents origins and functions in some utilisation scenario.* The *conceptual model of an information system* consists of a conceptual schema and of a collection of conceptual views that are associated (in most cases tightly by a mapping facility) to the conceptual schema. In a nutshell, a *conceptual model* is an enhancement of a model by concepts from a concept(ion) space.

The variety of notions for conceptual model is rather broad. We analyse some of the notions, systematise these notions, and discuss essential ingredients of conceptual models. This discussion allows to derive a research program in our area.

Keywords: Model, Conceptual model, Concept and notion of a model, Art of modelling.

1 What is a Conceptual Model

Modelling is a topic that has already been in the center of research in computer engineering and computer science since its beginnings. It is an old subdiscipline of most natural sciences with a history of more than 2.500 years. It is often restricted to Mathematics and mathematical models what is however to much limiting the focus and the scope. Meanwhile it became a branch in the Philosophy of Science. The number of papers devoted to modelling doubles each year since the early 2000's.

It is often claimed that there cannot be a common notion of model that can be used in sciences, engineering, and daily life. The following notion covers all known so far notions in agriculture, archaeology, arts, biology, chemistry, computer science, economics, electrotechnics, environmental sciences, farming, geosciences, historical sciences, languages, mathematics, medicine, ocean sciences, pedagogical science, philosophy, physics, political sciences, sociology, and sports. The models used in these disciplines are instruments that are deployed in certain scenarios (see [39]). A commonly acceptable statement for a general model notion is the following one[1]:

A model is a *well-formed*, *adequate*, and *dependable* instrument that represents *origins* and *functions* in some utilisation *scenario*. Its criteria of well-formedness, adequacy, and dependability must be commonly accepted by its *community of practice* within some *context* and correspond to the *functions* that a model fulfills in *utilisation scenarios*. The function determines the purposes and goals.

CS-conceptual modelling[2] is often related back to the introduction of the entity-relationship model(ling language) for information systems development. It surprises nowadays that there is no commonly accepted notion of conceptual model yet. There have been several trials but none of them was sufficient and was able to cover the idea of the conceptual model.

The database and information systems research communities are extensively using the term "conceptual model"[3]. The notion of conceptual model still needs some clarification: what is a conceptual model and what not; which application scenario use which kind of conceptual model; is conceptual modelling only database modelling; do we need to have an understanding of modelling; is a conceptual database model only a reflection of a logical database model; is a conceptual model a model or not; etc. Let us illustrate the wide spread and understanding of conceptual models, the activity of conceptual modelling, and the modelling as a scientific and engineering process by some examples[4,5]:

Reality and world description: Conceptual modelling is the activity of formally describing some aspects of

[1] We refer to the model-to_model-modelling compendium (see [39]) for notions that are not introduced in this paper.

[2] In the paper we restrict ourselves to this kind of conceptual model and thus omit the CS acronym. In general, a conceptual model is a representation of a system in its widest sense on the basis of concept(ion)s that allow people to consciously act and being guided in certain situations of their systems.

[3] Facetted search for the term "conceptual model" in DBLP results in more than 5.000 hits for titles in papers (normal DBLP search also above 3.400 titles).

[4] The notion of conceptualisation, conceptual models, and concepts are far older than considered in computer science. The earliest contribution to models and conceptualisations we are aware of is pre-socratic philosophy.

[5] Wikiquote (see [44]) lists almost 40 notions. We add our list to this list.

the physical and social world around us for purposes of understanding and communication. Such descriptions, often referred as conceptual schemata, require the adoption of a formal notation, a conceptual model in our terminology[6]. (see [25])

Community description : Conceptual modeling is about describing the semantics of software applications at a high level of abstraction[7].
Specifically, conceptual modelers (1) describe structure models in terms of entities, relationships, and constraints; (2) describe behavior or functional models in terms of states, transitions among states, and actions performed in states and transitions; and (3) describe interactions and user interfaces in terms of messages sent and received and information exchanged. In their typical usage, conceptual-model diagrams are high-level abstractions that enable clients and analysts to understand one another, enable analysts to communicate successfully with application programmers, and in some cases automatically generate (parts of) the software application. (see [12])

Conceptual database modelling : A data model is a collection of concepts that can be used to describe a set of data and operations to manipulate the data. When a data model describes a set of concepts from a given reality, we call it a conceptual model. (see [2, 10][8])

Instance-integrating conceptual modelling: A conceptual model consists of a conceptual schema and an information base. A conceptual schema provides a language for reasoning about an object system, and it specifies rules for the structure and the behaviour of the system. A description of a particular state is given in an information base, which is a set of type and attribute statements expressed in the language of the conceptual schema. (see [4])

System-representation models: A conceptual model is a descriptive model of a system based on qualitative assumptions about its elements, their interrelationships, and system boundaries. (see [7])

Representational models: A conceptual model is a type of diagram which shows of a set of relationships between factors that are believed to impact or lead to a target condition; a diagram that defines theoretical entities, objects, or conditions of a system and the relationships between them. (see [8])

Enterprise modelling and conceptual modelling : A conceptual is a model which represents a conceptual understanding (i.e. conceptualisation) of some domain for a particular purpose. A model is an artefact acknowledged by the observer as representing some domain for a particular purpose. (see [3])

Holistic view : In most cases, a model is also a conceptual model [9]. (see [28])

Conceptual models as a result of an activity: We use the name of conceptual modeling for the activity that elicits and describes general knowledge a particular information system needs to know. The main objective of conceptual modeling is to obtain that description, which is called a conceptual schema. (see [26])

Purpose-oriented modelling: Conceptual modelling is about abstracting a model that is fit-for-purpose and by this we mean a model that is valid, credible, feasible and useful. (see [31])

Documentation-oriented conceptual model: A conceptual data model is a summary-level data model that is most often used on strategic data projects. It typically describes an entire enterprise. Due to its highly abstract nature, it may be referred to as a conceptual model. (see [17])

Semiotics viewpoint: Conceptual modeling is about describing syntax, and semantics (potentially also pragmatics) of software applications at a high level of abstraction. (see [11])

Documentation and understanding viewpoint: A conceptual model of an application is the model of the application that the designers want users to understand. By using the application, talking with other users, and reading the documentation, users build a model in their minds of how to use the application. Hopefully, the model that users build in their minds is close to the one the designers intended. (see [18])

[6] And continuing: These terms are introduced by analogy to data models and database schemata. The reader may want to think of data models as special conceptual models where the intended matter consists of data structures and associated operations.

[7] Some research challenges in conceptual modeling: Provide the right set of modeling constructs at the right level of abstraction to enable successfully communication among clients, analysts, and application programmers. Formalize conceptual-modeling abstractions so that they retain their ease-of-communication property and yet are able to (partially or even fully) generate functioning application software. Make conceptual modeling serve as analysis and development tools for exotic applications such as: modeling the computational features of DNA-level life to improve human genome understanding, annotating text conceptually in order to superimpose a web of knowledge over document collections, leveraging conceptual models to integrate data (virtually or actually) providing users with a unified view of a collection of data, extending conceptual-modeling to support geometric and spatial modeling, and managing the evolution and migration information systems. Develop a theory of conceptual models and conceptual modeling and establish a formal foundation of conceptual modeling.

[8] Another version is the following one: The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. A conceptual schema hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints. A high-level data model or an implementation data model can be used at this level.

[9] The slides of the keynote talk state: A conceptual model is a simplification of a system built with an intended goal in mind.
An abstraction of a system to reason about it (either a physical system or a real or language-based system). A description of specification of a system and its environment for some purpose. One main conclusion that we can reach is that the distinction between "model" and "conceptual model" is not always as precise as it should be.

Conceptualisations of models: Conceptual models are nothing else as models that incorporate concepts and conceptions which are denoted by names in a given name space. A concept space [10] consists of concepts (see [24]) as basic elements, constructors for inductive construction of complex elements called conceptions, a number of relations among elements that satisfy a number of axioms, and functions defined on elements. (see [38])

At the ER'2017 conference a special brainstorming and discussion session has been organised with the task to coin the notion of a conceptual model. It seems to be surprising that there is no commonly accepted notion of a conceptual model after more than 40 years of introduction of this concept into database research. One proposal of the brainstorming discussion was:

ER 2017 discussion proposal: A conceptual model is a partial representation of a domain that can answer a question.

As for a model, the purpose dimension determines the quality characteristics and the properties of a model.

In a nutshell, a *conceptual model* is an enhancement of a model by concepts from a concept(ion) space. It is formulated in a language that allows well-structured formulations, is based on mental/perception/domain-situation models with their embedded concept(ion)s, and is oriented on a modelling matrix that is a common consensus within its community of practice.

We thus meet a good number of challenges, e.g. the following ones: is there any acceptable and general notion of conceptual model; do conceptual models really provide an added and sustainable value; what are the differences between conceptual models and models; what is a model; what means conceptualisation; how to support language-based conceptual modelling; etc. This paper is oriented on these questions and tries to develop an answer to them. We restrict the investigation to conceptual models in computer science and computer engineering and thus do not consider conceptual modelling for product design, service design, other system's design, natural and social sciences. Physical conceptual models are also left out of scope.

2 Revisiting Conceptual Modelling

2.1 State-Of-Art and State-Of-Needs

Modelling offers the benefit of producing better and understandable systems. It is based on a higher level of abstraction compared to most programming languages. Whether a model must be formal is an open question. The best approach is to consider model suites (or ensembles) that consist of a coherent collection of models which are representing different points of view and attention. We observe a resurgence in domain specific approaches that are challenged by technical, organisational and especially language design problems. UML is not the solution yet because UML Models aren't executable but MDA needs them to be. The vast majority of UML models we have seen in industrial project are mere sketches and are informal and incomplete. They are not yet a viable basis for precise and executable models. Without precise models, no formal checking can take place. Therefore, these issues must be addressed either if modelling is well-accepted and gains significant presence in applications.

From the other side, the large body of knowledge on conceptual modelling in computer science is a results of hundreds of research papers over the last three-score years although different names have been used for it. Modelling is often based on a finalised-model-of-the-real-world paradigm despite the constant change in applications. Model quality has already been considered in a dozen papers. Modelling literacy is rarely addressed in education. Models must however be reliable, refinable, and translatable artifacts in software processes.

Conceptual modelling is supported by a large variety of tools. e.g. (see [21]). However, few of them support executable models. Of that few, far fewer still are actually rewarding to use. Conceptual models are acknowledged as mediators in the software development process. However, they are used and then not evolving with the evolution of the software. Reuse, migration, adaptation, and integration of models is still a lacuna. The lack of robust, evolution-prone and convenient translators is one reason. An environment as a constituent part for modelling and translation into a consistent, easy-to-use and -revise, seamless, and industry-quality tools is still on the agenda. Information and software systems become eco-systems. Modelling eco-systems are not yet properly addressed.

Models are also used for communication based on some injection of a name space while the community of practice uses a wealth of terms and terminology with which they express their nuances of viewpoints. So, we need a number of representation models beside the singleton graphical representation. At the same time, models must be properly formal and based on rules strictly to be followed or else having a risk of making illogical statements. Modelling must thus be based on methodologies.

2.2 Myths of (Conceptual) Modelling

Modelling and especially conceptual modelling is not yet well understood and misinterpreted in a variety of ways. It has brought a good number of myths similar to those known for software development (see [1]):

1. *Modelling is mainly for documentation.* The introduction of the conceptual modelling for database systems has been motivated by documentation scenario. A conclusion might be that modelling is a superfluous

[10] We follow R.T. White (see [37, 42]) and distinguish between concepts, conceptual, conceptional, and conceptions.

activity, especially in the case that documentation is not an issue.

2. *Modelling is finished with the use of the model and an initial phase*. Historic development of software started with requirements which were frozen afterwards and with modelling and specifications that were complete and became frozen before realisation begins.

3. *Modelling is only useful for heavyweight V-style software development*. Modelling and especially conceptual modelling is abandoned due to its burden and the discovery of the complexity of the software that is targeted.

4. *The collection of origins must be "frozen" before starting with modelling*. Models should be plastic and stable (one of the justification and thus dependability properties), i.e. the collection of origins to be modelled could change.

5. *The model is carved in stone and changes only from time to time if at all*. The realisation becomes 'alive' and thus meets continuous change requests. The model can have some faults, errors, misconceptions, misses etc. Extensions and additional services are common for systems. So, the model has to change as well.

6. *Modelling is starts with selecting and accommodating a CASE tool*. Although CASE tools are useful they impose their own philosophy, language, and treatment. Moreover, CASE tools allow to become too detailed. Instead, conceptual modelling should allow to create the model that is simple as possible and as detailed as necessary.

7. *Conceptual modelling is a waste of time*. Developers are interested in quick success and have their own perception model in mind. It seems to be superfluous to model and better to focus solely on how to write the code.

8. *Conceptual data modelling is a primary concern*. Data- and structure-driven development without consideration of the usage of the data in applications results in 'optimal' or 'normalised' data structure models and bad database performance. One must keep in mind the usage of the data, i.e. use a co-design method, e.g. (see [34]).

9. *The community of practice has a common understanding how to conceptually model*. Modelling skills evolve over years and are based on modelling practice and experience. Further, conceptual models are based on a common domain-situation model that has to be shared within the community of practice. So, the perception models of modellers should match.

10. *Modelling is independent on the language*. Modelling cannot be performed in any language environment. Language matters, enables, restricts and biases (see [43]).

Understanding these and other myths allows to better understand the modelling process and the models. One way to overcome them is the development of sophisticated and acknowledged frameworks. Model-centred development (see [23]) uses models as a kernel for development of systems. Conceptual modelling ist still taught as modelling in the small whereas modelling in the large is the real challenge.

2.3 Specifics of Notions

Let us return to the list of notions given in Section 1. Each of these notions has its graces, biases, orientations, applicability, acceptability, and specifics.

Scopes of conceptual models may vary from very general models to fine-grained models. General models allow to reason on system properties whereas fine-grained models serve as a blueprint for development.

Result-oriented viewpoint: Conceptual models can be seen as the final result and documentation of an activity that follows a certain development strategy such as agile, extreme, waterfall etc. methodologies.

Communication viewpoint: Conceptual models are a means for communication and negotiation among different stakeholders.

System construction orientation: Database, information and software system development is becoming more complex, more voluminous, requires higher variety, and changes with higher velocity. So a quick and parsimonious comprehension becomes essential and supports higher veracity and an added value for the system itself.

Perception and domain-situation models are specific mental models either of one member or of the community of practice within one application area. It is not the real world or the reality what is represented. It is the common consensus, world view and perception what is represented.

Conceptual models as documentation: Models provide also quality in use, i.e. they allow to survey, to understand, to negotiate, and to communicate.

Conceptual modelling with prototypes: Models can be enhanced by prototypes or sample populations. A typical approach is sample-based development (see [16]).

Visualisation issues: Conceptual models may be combined with representation models, e.g. visualisation models on the basis of diagrammatic languages.

Biased conceptual modelling approaches: Conceptual models are often models with a hidden background, especially hidden assumptions that are commonly accepted in a community of practice in a given context and utilisation scenario.

Semiotics and semiology of conceptual modelling: Conceptual models are often language-based. The language selection is predetermined and not a matter of consideration in the modelling process.

Quality models: Conceptual models should be well-formed and satisfy quality requirements depending on their function in utilisation scenarios.

Concepts, conceptions: The elements in a conceptual models are annotated by names from some name space. These names provide a reference to the meaning, i.e. a reference to concepts and conceptions in a concept space.

Conceptual model suites: Models can be holistic or consist of several associated models where in the latter case each of them represents different viewpoints. For instance, a conceptual database model consists of a schema and a number of derived views which represent viewpoints of business users.

Normal models: Conceptual models represent only certain aspects and are considered to be intentionally enhanced by elements that stem from commonsense, consensuses, and contexts.

A normal models (called 'lumped' model in [45]) is a part of the model that is considered to be essential and absolutely necessary. The *normal model* has a context, a community of practice that puts up with it, a utilisation scenario for which is is minimally sufficient, and a latent – or better deep – model on which it is based (see [45] for 'base' model). The *deep model* combines the unchangeable part of a model and is determined by the grounding for modelling (paradigms, postulates, restrictions, theories, culture, foundations, conventions, authorities), the outer directives (context and community of practice), and the basis (assumptions, general concept space, practices, language as carrier, thought community and thought style, methodology, pattern, routines, commonsense) of modelling. The (modelling) *matrix* consists of the deep model and the modelling scenarios. The last ones are typically stereotyped in dependence on the chosen *modelling method*.

This variety of viewpoints to conceptual models illustrates the different requirements and objectives of models. So, we might ask whether a common notion of a conceptual model exists or whether we should use different notions.

2.4 Problems and Challenges

Conceptual modelling techniques suffer from a number of weaknesses. These weaknesses are are mainly caused by concentration on database modelling and by non-consideration of application domain problems that must be solved by information systems. We follow the state-of-the-art analysis of A. van Lamsweerde (see [40, 41]) who gave a critical insight into software specification and arrive with the following general weaknesses for conceptual modelling of information and database systems:

Limited scope. The vast majority of techniques are limited to the specification of data structuring, that is, properties about what the schema oft he database system is expected to do. Classical functional and nonfunctional properties are in general left outside or delayed until coding.

Poor separation of concerns. Most modelling approaches provide no support for making a clear separation between (a) intended properties of the system considered, (b) assumptions about the environment of this system, and (c) properties of the application domain

Low-level schematology. The concepts in terms of which problems have to be structured and formalized are concepts of modelling in the small - most often, data types and some operations. It is time to raise the level of abstraction and conceptual richness found in application domains.

Isolation. Database modelling approaches are isolated from other software products and processes both vertically and horizontally. They neither pay attention to what upstream products in the software might provide or require nor pay attention to what companion products should support nor provide a link to application domain description.

Poor guidance. The main emphasis in the database modelling literature has been on suitable sets of notations and on a posteriori analysis of database schemata written using such notations. Constructive methods for building correct models for complex database or information systems in a safe, systematic, incremental way are by and large non-existent.

Cost. Many information systems modelling approaches require high expertise in database systems and in the white-box use of tools.

Poor tool feedback. Many database system development tools are effective at pointing out problems, but in general they do a poor job of (a) suggesting causes at the root of such problems, and (b) proposing better modelling solutions.

Modern modelling approaches must not start from scratch. We can reuse achievements of database modelling in a systematic form and thus maintain theories and technologies while supporting new paradigms.

Constructiveness. Models of information systems can be built incrementally from higher-level ones in a way that guarantees high quality by construction. A method, is typically made of a collection of model building strategies, paradigm and high-level solution selection rules, model refinement rules, guidelines, and heuristics. Some oft hem might be domain-independent, some others might be domain-specific.

Support for comparative analysis. Database models depend on the experience oft he developer, the background or reference solutions on hand, and on preferences of developers. Therefore, the results within a team of developers might need a revision or a transformation to a holistic solution. Beyond the modelling qualities we may develop precise criteria and measures for assessing models and comparing their relative merits.

Integration. Tomorrow's modelling should care for the vertical and horizontal integration of models within the entire analysis, design, development, deployment and maintenance life cycle - from high-level goals to be supported by appropriate architectures, from informal formulation of information system models to conceptual models , and from conceptual models to implementation models and their integration into deployment of information systems.

Higher level of abstraction. Information systems modelling should move from infological design to holistic co-design of structuring, functionality, interactivity and distribution. These techniques must additionally be error-prone due tot he complexity of modern information systems. These abstraction techniques may be combined with refinement techniques similar to those that have been developed fort he abstract state machines.

Richer structuring mechanisms. Most modelling paradigms of the modelling-in-the-small approach available so far for modularising large database schemata have been lifted from software engineering approaches, e.g., component development. Problem-oriented constructs be developed as well model suites that provide a means for handling a variety of models and viewpoints.

Extended scope. Information system development approaches need to be extended in order to cope with the co-design of structuring, functionality, interactivity and distribution despite an explicit treatment of quality or non-functional properties.

Separation of concerns. Information system modelling languages should enforce a strict separation between descriptive and prescriptive properties, to be exploited by analysis tools accordingly.

Lightweight techniques. The use of novel modelling paradigms should not require deep theoretical background or a deep insight into information systems technology. The results or models should be compiled to appropriate implementations.

Multi-paradigm modelling. Complex information systems have multiple facets. Since no single modelling paradigm or universal language will ever serve all purposes of a system. The various facets then need to be linked to each other in a coherent way.

Multilevel reasoning and analysis. A multi-paradigm framework should support different levels of modelling, analysis, design and development - from abstract and general to deep-level analysis and repairing of detected deficiencies.

Multi-format modelling. To enhance the communicability and collaboration within a development and support team the same model fragment must be provided in a number of formats in a coherent and consistent way.

Reasoning in spite of errors. Many modelling approaches require that the model must be complete before the analysis can start. We claim that is should be made possible to start analysis and model reasoning much earlier and incrementally.

Constructive feedback from tools. Instead of just pointing out problems, future tools should assist in resolving them.

Support for evolution. In general, applications keep evolving due to changes in the application domain, to changes of technology, changes in information systems purposes etc. A more constructive approach should also help managing the evolution of models.

Support for reuse. Problems in the application domain considered are more likely to be similar than solutions. Models reuse should therefore be even more promising than code reuse.

Measurability of modelling progress. To be more convincing, the benefits of using information models should be measurable as well as their deficiencies.

This list of theories, solutions and methodological approaches is not exhaustive. It demonstrates, however, that modelling in the large and modern information systems modelling require specific approaches beyond integration of architectures into the analysis, design and development process.

2.5 The Research Issue

Let us reconsider the notions presented in Section 1. Table 1 compares essential properties of models. Missing model elements are denoted by n(ot).g(iven).

We observe that dependability is often either implicit or not considered in the model notion. Implicitness is mainly based on the orientation to normal models. The model matrix and especially the deep model are considered to be agreed before developing the model.

The origin is too wide in most cases. Models are not oriented towards representing some reality or the world. They are typically based on some kind of agreement made within a community of practice and according to some context, i.e.

Table 1. Orientation of notions of conceptual models according model properties

version	adequate	dependable	origin	function	scenario	concepts
reality, world	reflection, truncation	formal, reflection	world	describe	communication, understanding	n.g.
community	abstraction, mapping	semantic invariance	software application	describe	construction	n.g.
conceptual database	mapping, homomorphy	n.g.	data, operations	describe	construction, documentation	reality concepts
system & instance	mapping, abstraction	n.g.	system, objects	n.g.	construction	n.g.
system representation	reflection,	qualitative assumptions	system	describe	representation	system concepts
represen-tional	mapping	n.g.	relation-ships	represent	visualisation	impact factors
enterprise	mapping, abstraction	faithful	domain	purpose-determined	understanding	concept space
result of activity	mapping,	n.g.	system knowledge	describe	acquisition, elicitation	domain knowledge
purpose-oriented	abstraction purposeful	viable, fit	any	elicitate	n.g.	n.g.
documen-tation	summary, abstraction	n.g.	data system	represent, survey	strategy development	n.g.
semiotics	syntax abstraction	semantics, pragmatics	software application	describe	representation	n.g.
document understand	mapping	closeness	application	understand by users	design	n.g.
conceptualise	formal representation	semantics	any	describe	representation	concept(ion) space
ad-hoc	selective mapping	n.g.	domain	consider problem	solving	n.g.

they reflect some domain-situation model[11] or more generally some mental model[12]. They might represent a perception model of some members of the community practice. They say what the phenomena in the given domain are like.

Table 1 directs to a conclusion that the function is mainly oriented towards description and partially prescription for systems development. The notion of the conceptual model has, however, mainly considered in system construction scenarios.

Concepts are often hidden behind the curtain of conceptual models. A conceptual model does not reflect the reality. Instead it reflects the mental understanding within its utilisation scenario.

These observations show now directly some open issues that should be solved within a theory and practice of conceptual modelling. Let us state some of them.

Research question 1. What are the *origins* for conceptual models? Are these mainly domain-situation and perception models from one side and systems on the other side?

Research question 2. How tightly conceptual models are bound to their *modelling matrix* and especially their *deep model*? To what extent conceptual models are normal models that are intentionally combined with their deep models?

Research question 3. Which functions have conceptual models in which utilisation scenarios? Which properties must be satisfied by conceptual models in these scenarios? Which purposes and goals can be derived?

Research question 4. What is the role of the *community of practice* in conceptual modelling? Which kind of model supports which community in which context?

Research question 5. Conceptual modelling is less automated and more human dependent than any other development, analysis, and design process for information systems. It is a highly creative process. Is there any formalisation and foundation for this process?

Research question 6. Since models must not be conceptual models (see models in [39]), we might ask whether there exists a set of characteristics or criteria that separate

[11] We restrict consideration to our field and thus to domain-oriented models. These models describe the application domain and more specifically the understanding, observation, and perception of an application domain that is accepted within a community of practice. In general, a situation model is a mental representation of a described or experienced situation in a real or imaginary world (see [30]).

[12] Mental models are out-of-scope in this paper. Those consist of an evolving model suite with small-scale and parsimonious models carried in human head (see [13, 19]). They support various kinds of observation, information acquisition and filtering, reasoning, storage and information (de)coding, and communication. They are dependent on the observations, imaginations, and comprehension a human has made. Unlike conceptual models, mental models must neither be accurate, nor complete, and not consistent.

a conceptual model from a model that is not conceptual. What are the concept space that can be used for an enhancement of a model by concepts or conceptions?

3 The Nature of Models

3.1 The Notion of a (Conceptual) Model

The model is an utterance and also an imagination. As already stated above (see also [39]), a model is a *well-formed*, *adequate*, and *dependable* instrument that represents *origins* and *functions* in some utilisation *scenario*. A model is a representation of some origins and may consist of many expressions such as sentences. *Adequacy* is based on satisfaction of the purpose or function or goal, analogy to the origins it represents and the focus under which the model is used. *Dependability* is based on a *justification* for its usage as a model and on a *quality certificate*. Models can be evaluated by one of the evaluation frameworks. A model is *functional* if methods for its development and for its deployment are given. A model is *effective* if it can be deployed according to its portfolio, i.e. according to the tasks assigned to the model. Deployment is often using some deployment macro-model, e.g. for explanation, exploration, construction, documentation, description and prescription.

Models *function* as *instruments* or tools. Typically, instruments come in a variety of forms and fulfill many different functions. Instruments are partially independent or autonomous of the thing they operate on. Models are however special instruments. They are used with a specific intention within a utilisation scenario. The quality of a model becomes apparent in the context of this scenario.

Model development is often targeted on normal models and implicitly accepts the deep model. A model is developed for some modelling scenarios and thus biased by its modelling matrix. The deep model and the matrix thus 'infect' the normal model.

Within the scope of this paper, we concentrate on representation models as proxies. So, a model of a collection of origins, within some context, for some utilisation scenario and corresponding functions within these scenarios, and for a community of practice is

- a relatively enduring,
- accessible
- but limited
- internal and at the same time external
- representation of the collection of origins.

The model becomes *conceptual* by incorporation of concepts and conceptions commonly accepted, of ideas provided by members from the community of practice, or of general well-understood language-like semiotic components. One main utilisation scenario for conceptual database model is system construction[13]. In this case, the conceptual model thus becomes predictively accurate for the system envisioned and technologically fruitful. The model is an utterance and also an imagination. Other scenarios for conceptual models are: system modernisation, explanation, exploration, communication, negotiation, problem solving, supplantation, documentation, and even theory development.

Conceptual models must not limited to representation of static aspects of systems. They can also be used for representation of dynamic aspects such as business stories, business processes, and system behaviour. The carrier of representation is often some language. In this case, a conceptual model can be considered to be an utterance with a collection speech acts. The model itself can be then build on well-formedness rules for its syntax, semantics, and pragmatics, or more general of semiotics and semiology. According to J. Searle (see [33]), a speech act consists of uttering elements, referring and predicating, requesting activities, and causing an effect. Whether at all and which language is going to be used is a matter of controversy too.

3.2 Facets of a Conceptual Model

1. The conceptual model is a result of a perception and negotiation process. The conceptual model represents mental models, especially domain-situation models or a number of perception models. Domain-situation models represent a settled perception within a context, especially an application. Perception models might differ from the domain-situation model. They are personal perceptions and judgements of a member of the community of practice. Maturity of conceptual models is reached after the community of practice negotiated different viewpoints and has found an agreement.

2. The conceptual model represents its collection of origins. Considerations about what to model and what not to model are expressed via the adequacy criteria, especially for analogy to its origins, for focusing on specifics of the origins, and also on well-formedness of the model. The conceptual model does not represent a real world or a problem domain. It is already based on perception models of users about this problem domain or on domain-situation models of a user community on this problem domain.

3. The conceptual model is an instrument. The conceptual model is used in some utilisation scenario by its users. So it functions in this utilisation scenario. It should describe in a more abstract way compared to the origins how the user conceives it and thus does not target on describing the origins.

4. The deep model underpins the conceptual model. The deep model consists of all elements that are taken for granted, are considered to be fixed, and are common within the context for the community of practice. Elements of this model are symbolic generalizations as formal or readily formalisable components or laws or law schemata, beliefs in particular heuristic and ontological models or analogies

[13] Notice however that the first introduction of conceptual data models has been oriented on a documentation scenario.

supplying the group with preferred or permissible analogies and metaphors, and values shared by the community of practice as an integral part and supporting the choice between incompatible ways of practicing their discipline. There is no need to redevelop this model. So, the normal model only display those elements that are additionally introduced for the model.

5. The conceptual modelling matrix. The modelling matrix combines the deep model with the typical utilisation scenarios that are accepted by a community of practice in a given context. It specifies a guiding question as a principal concern or scientific interest that motivates the development of a theory, and techniques as the methods an developer uses to persuade the members of the community of practice to his point of view. Although often not explicitly stated, the model matrix consists of a number of components: the objectives, inputs (or experimental factors), outputs (or responses), content requests, grounding, basis, and simplifications. The matrix sets a definitional frame for the normal model. It might support modelling by model stereotypes. The agenda of the modelling method is derived from the matrix. The matrix determines also a spefic treatment of adequacy and dependability for a model.

6. The performance and quality criteria. The model is a persistent and justified artifact that satisfy a number of conditions according to its function such as empirical corroboration according to modelling objectives, by rational coherence and conformity explicitly stated through conformity formulas or statements, by falsifiability, and by stability and plasticity within a collection of origins. The quality characteristics bound the model to be valid, credible, feasible, parsimonious, useful, and at the same time as simple as possible and as complex as necessary.

7. The model is the main ingredient of a modelling method. Sciences and technologies have developed their specific deployment of models within their investigation, analysis, development, design etc. processes. The deep model and the matrix are often agreed. The central element of all modelling methods is the model that is used as an instrument in scenarios which have been stereotyped for the given modelling method. The modelling method typically also includes design of a representation model (or a number of such). The representation model of the (conceptual) model may be based on approaches such as diagramming and visualisation. It uses a set of predefined signs: icons, symbols, or indexes in the sense of Peirce.

3.3 Sources for Conceptual Models: Domain-Situation and Perception Models

The domain-situation model is build by a community of practice on a semantical level. It refers to the world-as-described-and-conceived-by-the-deep-model. It thus forms the deep understanding behind the conceptual model. This deep internal structure of the conceptualisation is commonly shared in the community, abstracts from accidental origins, uses a partial interpretation, exhibits (structural) hidden similarities of all origins under consideration, and presents the common understanding in the community. It gives thus a literal meaning to the domain. The context for the conceptual model is typically governed by domain-situation models. The domain-situation model is thus one source for the conceptual model.

A domain-situation model might or might not exist. It shapes, however, what is seen in an application domain and how to reason about what is seen. They represent some common negotiated understanding in the application domain. It may represent the application domain as it is or the application domain as it makes sense to be characterised, categorised or classified in one way rather than another given certain interests and aptitudes or more generally given certain background.

The second source for conceptual models is a collection of perception models that are provided and acknowledged by members of this community of practice. A perception model is one kind of epistemological mental model with its verbal, visual and other information compiled on the basis of cognitive schemata. It organises, identifies, and interprets observations made by the member. It does not need to know the deep facts or essential properties of the origins in order to succeed in communicating about them or to reason. The perception model typically follows the situation that it represents. It is however often underdetermined and thus may also partially contradictious. So it parallels and imitates parts of the reality ('Gestalt' notion of the model). They provide a partial understanding, refer to some aspect, may use competing sub-models about the same stuff, and may set alternatives on meaning. It is build by intuitive, discursive and evidence-backed perception, by imagination, and by comprehension. It is shaped by learning, memorisation, expectation, and attention. Perception models serve as an add-on beyond domain-situation models.

Both model kinds represent observations and phenomena for the community of practice. Typical elements are classifications, categorisations, ontologies and catalogs, background especially the grounding, practices and principles, pattern and solutions, and a commonly accepted basis from the modelling background.

These models also reuse a commonly accepted basis from the modelling background such as potentially available constructions or conceptions as definitional knowledge, signs from a language (symbols, indexes and icons), language-based semiotics and semiology, commonly accepted methods and techniques, guidelines and development approaches, approaches to realisation of models.

These two sources for conceptual models depend on the community of practice. So, different communities might use different kinds of verbal and nonverbal representation. Although they provide a literal meaning to the conceptual model they must not be explicitly stated within the conceptual model. They serve as the origin for the conceptual model and thus might not be explicitly incorporated into the conceptual model. The conceptual model may have its deep background, i.e. its basis and especially its grounding.

Both origins are not complete. Typically the scope of both models is not explicit. There are unknown assumptions applied for description, unknown restrictions of the model, undocumented preferences and background of the community of practice, and unknown limitations of the modelling language. Classically we observe for members of a community of practice that

- they base their design decisions on a "partial reality", i.e. on a number of observed properties within a part of the application,
- they develop their models within a certain context,
- they reuse their experience gained in former projects and solutions known for their reference models, and
- they use a number of theories with a certain exactness and rigidity.

The conceptual model to be developed is deeply influenced by these four hidden factors.

4 Conceptualisation of Models

The domain-situation model and also partially the perception model are commonly using concepts. Conceptual models reuse such concepts from these origins and thus inherit semantics and pragmatics from these models. Further, conceptualisation may also be implicit and may use some kind of lexical semantics of these models, e.g. word semantics, within a commonly agreed name space.

4.1 Concepts and Conceptions

Various notions of concept has been introduced, for instance, by J. Akoka, P. Chen, H. Kangassalo, R. Kauppi, A. Paivio, and R. Wille (see [6, 14, 22, 20, 27]). Artificial intelligence and mathematical logics use concept frames. Ontologies combine lexicology and lexicography. Concepts are used in daily life as a communication vehicle and as a result of perception, reasoning, and comprehension. Concept definition can be given in a narrative informal form, in a formal way, by reference to some other definitions etc. Some version may be preferred over others, may be time-dependent, may have a level of rigidity, is typically usage-dependent, has levels of validity, and can only be used within certain restrictions. We also may use a large variety of semantics (see [32]), e.g., lexical or ontological, logical, or reflective.

We distinguish two different meanings of the word 'concept' (see [42]):

1. Concepts are general categories and thing of interest that are used for classification. Concepts thus have fuzzy boundaries. Additionally, classification depends on the context and deployment.
2. Concepts are all the knowledge that the person has, and associates with, the concept's name. They are reasonable complete in terms of the business.

Conceptions (see [42]) are systems of explanation. They are thus more difficult to describe.

The typical definition frame we observed is based on definition items. These items can also be classified by the kind of definition. Concepts may simultaneously have different descriptions. Competing description may differently represent the same concept depending on context (e.g. time, space), validity, usage, and preferences of members of the community of practice. A concept may have elements that are necessary or sufficient, that may be of certain rigidity, importance, relevance, typicality, or Fuzziness. Based on the generalisations of the approach that has been proposed by G.L. Murphy (see [24, 35]), concepts are defined in a more sophisticated form as a tree-structured structural expression.

SpecOrderedTree(StructuralTreeExpression
(DefinitionItem, Modality(Sufficiency, Necessity),
Fuzziness, Importance, Rigidity,
Relevance, GraduationWithinExpression, Category))) .

Concept may be regarded as the descriptive and epistemic core units of perception and domain-situation models. These origins govern the way how a concept can be understood, defined, and used in a conceptual model. The conceptual model inherits thus concepts and their structuring within a concept space, i.e. conceptions.

4.2 Conceptualise

Conceptualisation and semantification are orthogonal concerns in modelling. *Conceptual modelling* is based on concepts that are used for classification of things. Concepts have fuzzy boundaries. Additionally, classification depends on the context and deployment. Conceptual[14] modelling uses *conceptions* which are systems of explanation.

Semantification (see [9]) improves comprehensibility of models and explicit reasoning on elements used in models. It is based on name spaces or ontologies that are commonly accepted in the application domain. Conceptual models are models enhanced by concepts and integrated in a space of conceptions.

Conceptualisation injects concepts or conceptions into models. These enriched models reflect those concepts from commonly accepted concept space. The concept space consists of a system of conceptions (concepts, theoretical statements (axioms, laws, theorems, definitions), models, theories, and tools). A concept space also may include procedures, conceptual (knowledge) tools, and associated norms resp. rules. Is is based on paradigms which are corroborated.

[14] Conceptual modelling is performed by a modeller that directs the process based on his/her experience, education, understanding, intention and attitude. Conceptual models are using/incorporating/integrating concepts (see [42]) Conceptional modelling aims at development of concepts.

4.3 Dependability of Conceptual Models

Models must be dependable, i.e. justified from one side and and qualitatively certified from the other side. Justification can be based on the domain-situation and perception models and the relation of the conceptual models to these models. If however such models are not available or of low quality then justification will become an issue. Quality certification is an issue of pragmatism and of added value of the conceptual model. So, we target on a high quality conceptualisation. Conceptualisation may be based on the seven principles of Universal Design (see [29]). Typical mandatory principles are usefulness, flexibility, simplicity, realisability, and rationality. Optional conceptualisation principles are perceptability, error-proneness, and parsimony.

The *principle of conceptualisation* is considered to be one -if not the main - of the seven fundamental principles for conceptual modelling (see [15]). The other six principles are: Helsinki, Universe of discourse, searchlight, 100%, onion, and three level architecture principles. They can be questioned further. These principles can be enhanced by the principles of understanding, of abstraction, of definition, of refinement, evaluation, and of construction (see [36]). Conceptualisation can be considered to be completed if: A conceptual schema should only include conceptually relevant aspects, both static and dynamic, of the universe of discourse, thus excluding all aspects of (external or internal) data representation, physical data organization and access, as well as all aspects of particular external user representation such as message formats, data structures, etc.

Based on Section 3.3, the principle of conceptualisation can be stated as follows:

A conceptual model should only include conceptually relevant aspects of the domain-situation and perception models. It does not consider neither aspects of realisation nor of representation. It includes, however, different viewpoints of business users and concepts from the common concept space.

5 Conclusion: Towards a Notational Frame for Conceptual Models

Conceptual modelling is not yet a science or culture. It is rather a craft or even an art. It can be learned similar to craft learning. It is however based on understanding and abstraction throughout the perception and domain-situation models, i.e. of mental models in general. Perception is dependent on deep models and thus incomplete, revisable, time-restricted, activity-driven, and context-dependent.

5.1 Slim, Light, and Concise Versions for Conceptual Models

Conceptual models are widely used in system construction scenarios. They function as description of the phenomena of interest within the context for its community of practice. So, conceptual models are normal models with rather specific modelling matrices and deep models. A slim notion of a conceptual model is should only reflect such normal models and refer to a specific modelling matrix. A light version needs to refer to some elements of the basis and to some context. A concise version must explicitly represent all the hidden details of a model, especially its relationships to the concept space, to the perception of this space by members of the community of practice, and to the utilisation scenario.

5.2 A Proposal for a Light Version: Conceptual Model $\sqsupseteq$ Model $\bigoplus$ Concepts

Conceptual modelling is not yet a common method in science (see [31]). Systems can be build without any conceptual model. It seems that there is no need for a formal conceptual modelling process. It seems to be too restrictive to require a full conceptual model. Performance and quality criteria are not commonly agreed. The science of conceptual modelling is still missing.

The main bottleneck is however the missing notion of a conceptual model. The conceptual model is a specific model and is based on conceptualisation. It might be language-bound. It is probably the most important aspect of system construction in computer science and computer engineering. It is however the most difficult and least understood. Minimal justification characteristics of models are classical viability, i.e. corroboration, validity, credibility, rational coherent and conform, falsifiable, stability against origin collection change. Minimal quality characteristics of models are the one for quality in use (e.g. usability, aptness for the function and purpose, value for the utilisation scenario, feasibility). Minimal performance characteristics are timely, elegant and feasible usage within the given context for their community of practice according to their utilisation scenario and their competencies or more general their profiles.

So, we might conclude for a *light version*: A conceptual model is a well-formed, adequate and dependable instrument that functions within its specific utilisation scenario, that represents origins, and that is enhanced by concepts from a concept(ion) space.

Therefore, the incorporation of concepts and the conceptions is one main difference to the model.

5.3 Lacunas of Conceptual Modelling

Since conceptual modelling is still more an art than a science and a culture of conceptual modelling is still beyond the horizons, we need

- an understanding of the area of conceptual modelling;
- a theory, techniques, and engineering of conceptualisation;

- an integrated multi-view approach for the needs and the capabilities of the members of the community of practice;
- a refinable definition of the conceptual model with all three versions, i.e. a simplified version, a fully fledged version, and an assessable version;
- a working approach with intentional and thus latent matrices and deep models for daily practice; and
- an understanding of language use in conceptual modelling.

These lacunas do not limit usability, usefulness, and utility of conceptual models. Conceptual database models improve from one side system comprehension. They allow to indicate associations among system elements, reduce the effect of bad implementation, provide abstraction mechanisms, support prediction of system behaviour, provide an elegant and adequate overview of the system at various levels of abstraction, support the construction of different user views, and cross-reference multiple viewpoints. From the other side, the reduce the developers, maintainers and programmers overhead. They support a simple and free navigation through components of the database system, provide an easy deduction of various viewpoints that represent the needs of business users, support concentration and focusing in evolution and maintenance phases, display the decisions made during development, indicate opportunities for further development and system maintenance, reduce the effort by reuse of design and development decisions that have already been made, and use a comfortable and effective visualisation. So, conceptual models are not restricted to construction scenarios or to database modelling.

We realise that the development and the acceptance of a notion of conceptual model follows the 13 Commandments stated (see [5]):

1. Thou shalt choose an appropriate notation.
2. Thou shalt formalise but not overformalise.
3. Thou shalt estimate costs.
4. Thou shalt have a formal methods guru on call.
5. Thou shalt not abandon thy traditional development methods.
6. Thou shalt document sufficiently.
7. Thou shalt not compromise thy quality standards.
8. Thou shalt not be dogmatic.
9. Thou shalt test, test, and test again.
10. Thou shalt reuse.
11. Thou shall meet intentions of all members of the community of practice
12. Thou shall provide a usable notation, i.e. for verification, validation, explanation, elaboration, and evolution.
13. Thou shall be robust against misinterpretation, errors, etc.

References

[1] S.W. Ambler and P.J. Sadalage. *Refactoring databases - Evolutionary database design*. Addison-Wesley, 2006.

[2] C. Batini, S. Ceri, and S. Navathe. *Conceptual database design (an entity-relationship approach)*. Benjamin/Cummings, Redwood City, 1992.

[3] M. Bjeković. *Pragmatics of Enterprise Modelling Languages: A Framework for Understanding and Explaining*. PhD thesis, Radboud University Nijmegen, 2017.

[4] M. Boman, J.A. Bubenko Jr., P. Johannesson, and B. Wangler. *Conceptual modelling*. Prentice Hall, London, 1997.

[5] J. P. Bowen and M. G. Hinchey. Ten commandments ten years on: Lessons for asm, b, z and vsr-net. In *Rigorous Methods for Software Construction and Analysis*, volume 5115 of *Lecture Notes in Computer Science*, pages 219–233. Springer, 2009.

[6] P.P. Chen, J. Akoka, Kangassalo H, and B. Thalheim, editors. *Conceptual Modeling: Current Issues and Future Directions*, volume 1565 of *LNCS 1565*. Springer, 1998.

[7] Business dictionary. Conceptual model. http://www.businessdictionary.com/definition/conceptual-model.html, 2017. Assessed Nov. 21, 2011.

[8] WordNet dictionary. Conceptual model. http://www.dictionary.com/browse/conceptual-model, 2017. Assessed Nov. 21, 2011.

[9] M. Duží, A. Heimburger, T. Tokuda, P. Vojtas, and N. Yoshida. Multi-agent knowledge modelling. In *Information Modelling and Knowledge Bases XX*, Frontiers in Artificial Intelligence and Applications, 190, pages 411–428. IOS Press, 2009.

[10] R. Elmasri and S. Navathe. *Fundamentals of database systems*. Addison-Wesley, Reading, 2000.

[11] D. Embley and B. Thalheim. Preface. In *The Handbook of Conceptual Modeling: Its Usage and Its Challenges*, pages v–ix. Springer, Berlin, 2011.

[12] ER community. Homepage. http://conceptualmodeling.org, 2017. Accessed Oct. 29, 2017.

[13] J.W. Forrester. *Collected papers of J.W. Forrester*, chapter Counterintuitive behaviour of social systems, pages 211–244. Wright-Allen Press, Cambridge, 1971.

[14] B. Ganter and R. Wille. *Formal concept analysis - Mathematical foundations*. Springer, Berlin, 1998.

[15] J. J. Van Griethuysen. The Orange report ISO TR9007 (1982 - 1987) Grandparent of the business rules approach and sbvr part 2 - The seven very fundamental principles. *Business Rules Journal*, 10(8), August 2009. http://www.brcommunity.com/a2009/b495.html.

[16] T. A. Halpin. Object-role modeling. In *Encyclopedia of Database Systems*, pages 1941–1946. Springer US, 2009.

[17] InfoAdvisors. What are conceptual, logical and physical data models? http://www.datamodel.com/index.php/articles/what-are-conceptual-logical-and-physical-data-models/, 2017. Assessed Nov. 21, 2011.

[18] J. Johnson and A. Henderson. Conceptual modelling in a nutshell. http://boxesandarrows.com, 2013. Accessed Jan. 29, 2013.

[19] P.N. Johnson-Laird. *Mental models: Towards a cognitive science of language, inference, and consciousness.* Cambridge University Press, London, 1983.

[20] H. Kangassalo and J. Palomäki. Definitional conceptual schemata - the core for thinking, learning, and communication. Keynote given at 25th EJC Conference, Maribor, Slovenia, June 2015.

[21] D. Karagiannis, H. C. Mayr, and J. Mylopoulos, editors. *Domain-Specific Conceptual Modeling, Concepts, Methods and Tools*. Springer, 2016.

[22] R. Kauppi. Einführung in die Theorie der Begriffssysteme. *Acta Universitatis Tamperensis, Ser. A, Vol. 15, Tampereen yliopisto, Tampere*, 1967.

[23] H. C. Mayr, J. Michael, S. Ranasinghe, V. A. Shekhovtsov, and C. Steinberger. Model centered architecture. In *Conceptual Modeling Perspectives.*, pages 85–104. Springer, 2017.

[24] G. L. Murphy. *The big book of concepts*. MIT Press, 2001.

[25] J. Mylopoulos. Conceptual modeling and Telos. In *Conceptual modeling, databases, and CASE: An integrated view of information systems development*, LNCS 10509, pages 49–68, Chichester, 1992. Wiley.

[26] A. Olivé. *Conceptual modeling of information systems*. Springer, Berlin, 2007.

[27] A. Paivio. *Mental representations: A dual coding approach*. Oxford University Press, Oxford, 1986.

[28] O. Pastor. Conceptual modeling of life: Beyond the homo sapiens. http://er2016.cs.titech.ac.jp/assets/slides/ER2016-keynote2-slides.pdf, 2016. Keynote given at ER'2016 (Nov. 15).

[29] B. Patil, K. Maetzel, and E. J. Neuhold. Design of international textual languages: A universal design framework. In *IWIPS*, pages 41–56. Product & Systems Internationalisation, Inc., 2003.

[30] G.A. Radvansky and R.T. Zacks. *Cognitive models of memory*, chapter The retrieval of situation-specific information, pages 173–213. MIT Press, Cambridge, 1997.

[31] S. Robinson. Conceptual modelling: Who needs it? *SCS M & S Magazine*, (2):1–7, 2010.

[32] K.-D. Schewe and B. Thalheim. Semantics in data and knowledge bases. In *SDKB 2008*, LNCS 4925, pages 1–25, Berlin, 2008. Springer.

[33] J. Searle. *Speech Acts: An essay in the philosophy of language*. Cambridge University Press, Cambridge, England, 1969.

[34] B. Thalheim. *Entity-relationship modeling – Foundations of database technology*. Springer, Berlin, 2000.

[35] B. Thalheim. The conceptual framework to user-oriented content management. In *Information Modelling and Knowledge Bases*, volume XVIII of *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2007.

[36] B. Thalheim. Towards a theory of conceptual modelling. *Journal of Universal Computer Science*, 16(20):3102–3137, 2010. http://www.jucs.org/jucs_16_20/towards_a_theory_of.

[37] B. Thalheim. The conceptual model $\equiv$ an adequate and dependable artifact enhanced by concepts. In *Information Modelling and Knowledge Bases*, volume XXV of *Frontiers in Artificial Intelligence and Applications, 260*, pages 241–254. IOS Press, 2014.

[38] B. Thalheim. General and specific model notions. In *Proc. ADBIS'17*, LNCS 10509, pages 13–27, Cham, 2017. Springer.

[39] B. Thalheim and I. Nissen, editors. *Wissenschaft und Kunst der Modellierung: Modelle, Modellieren, Modellierung*. De Gruyter, Boston, 2015.

[40] A. van Lamsweerde. Formal specification: a roadmap. In *ICSE - Future of SE Track*, pages 147–159, 2000.

[41] A. van Lamsweerde. Requirements engineering: from craft to discipline. In *SIGSOFT FSE*, pages 238–249. ACM, 2008.

[42] R.T. White. Commentary: Conceptual and conceptional change. *Learning and instruction*, 4:117–121, 1994.

[43] B.L. Whorf. *Lost generation theories of mind, language, and religion*. Popular Culture Association, University Microfilms International, Ann Arbor, Mich., 1980.

[44] Wikiquote. Conceptual model. https://en.wikiquote.org/wiki/Conceptual_model, 2017. Assessed Nov. 21, 2017.

[45] B. P. Zeigler, T. G. Kim, and H. Praehofer. *Theory of Modeling and Simulation*. Elsevier Academic Press, 2000.