Data structures

Data in motion and data in storage

- A million years ago: social animals exchanged **messages** and processed them in the light of **memories** retained.
- 200,000 BC: humans formed message using a **verbal language** in which the words typify what is described.
- 5,000 BC: humans formed message in **writing**, which facilitated the conduct of regular business operations.

Before “The Information Age”

- Human actors created, processed and moved data using
  - clay tablets
  - pen and paper
  - typewriters and
  - snail-mail postal organisations.

- Business depended on business information being.
  - moved in messages (data flows) – often paper
  - stored in memories (data stores) – card indexes and filing cabinets

Digitisation led to “The Information Age”

- 1960 AD: IT empowered **high speed message** transmission and **high volume memory** retention. It massively increased the ability of a business to
  - capture, move, store,
  - process, and analyse business data.

- “Today’s CEOs know that the effective management and exploitation of information through IT is a key factor to business success.” (TOGAF 9.1)

- Enterprise architects recognize the importance of understanding
  - **Data in motion:** messages sent in business activities
  - **Data in storage:** memories retained
  - **Meta data:** the qualities of data in messages and memories.

4.5 Describe the data structures used by a business and/or its applications

- Data in storage.
- Data in motion.
- Data structures.
- Data items.
- Data lifecycle: create, store, use, archive, delete.

- As data is used throughout an organisation and throughout various applications, its state, structure and use changes.

- Candidates shall recognise the key terms listed and be able to describe how data moves through the data lifecycle, from creation to eventual deletion.

Data items and structures

- **Data item**
- An instance of a data type that holds a specific data value.
- Such as an argument value, a return value, or a field value in a database.

- **Data structure**
- [A data type] a structure that arranges data items in a group or related groups.

Data store/memory structures

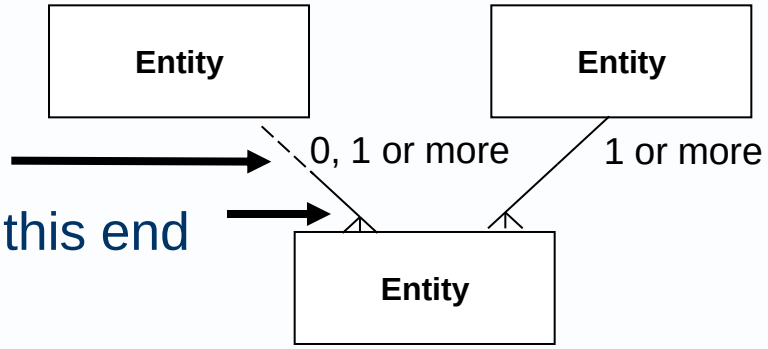
Data structures in data stores

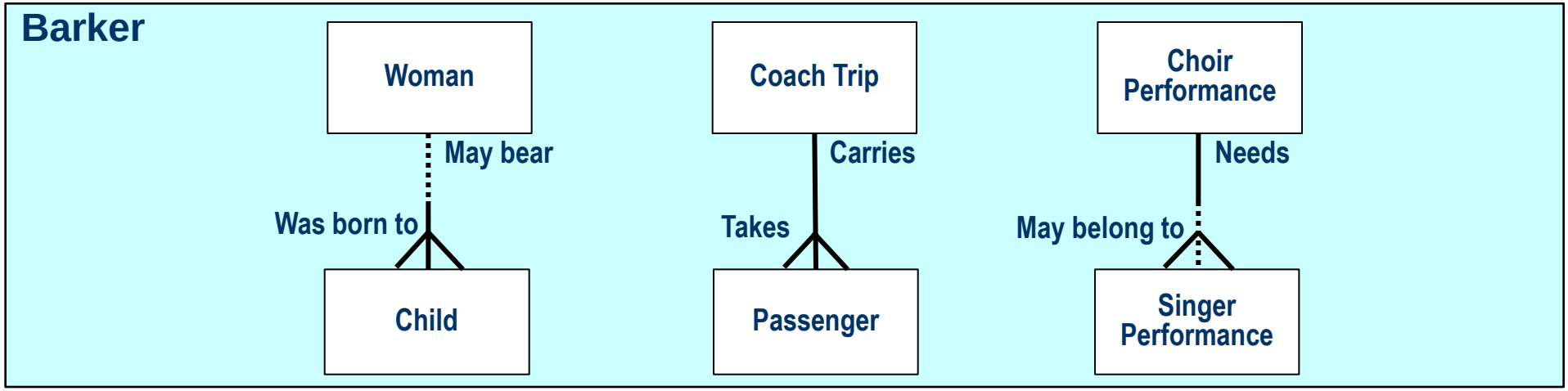
- **Data store**
- A container for persistent data structure, accessible by software.
- Sometimes on discs, increasingly on solid state drives.

- **Data store structures**
- An **entity-relationship diagram** is a concept map with cardinalities added to the relationship lines.

Barker data model notation

- Dashes mean *optional* at this end
- Crowsfoot means *more than one* at this end

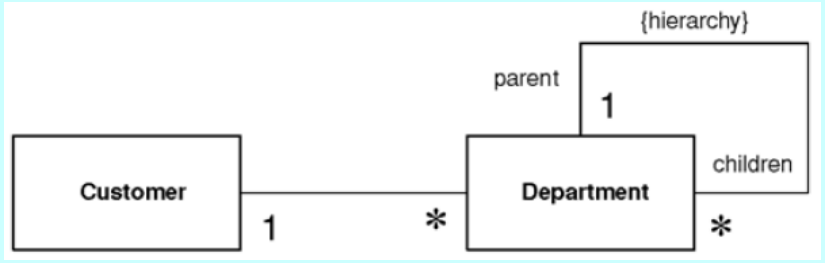




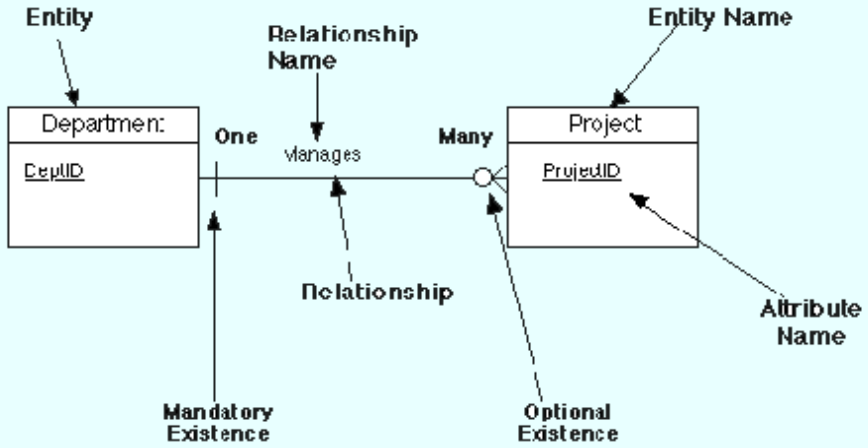
Other data model notations

Avancier

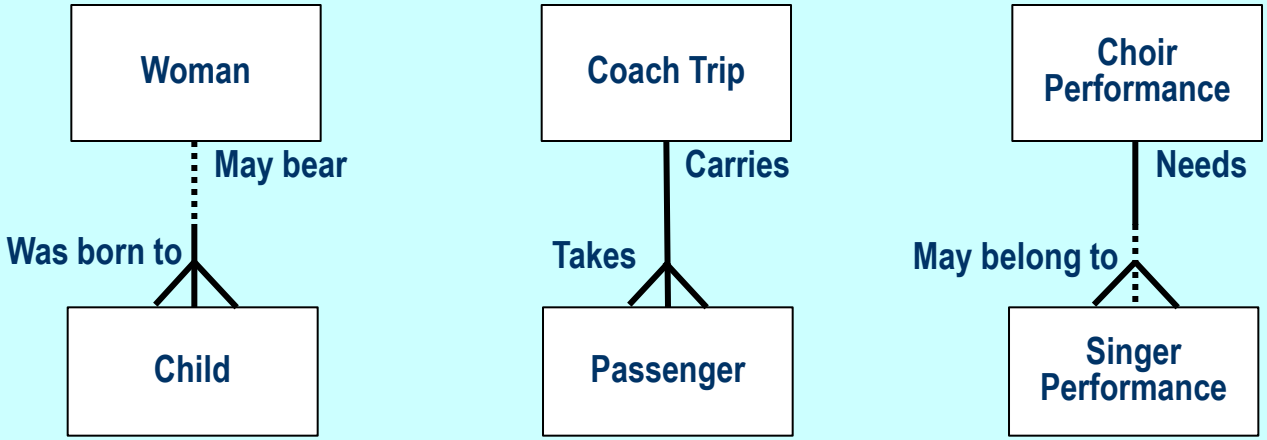
UML



IDEF



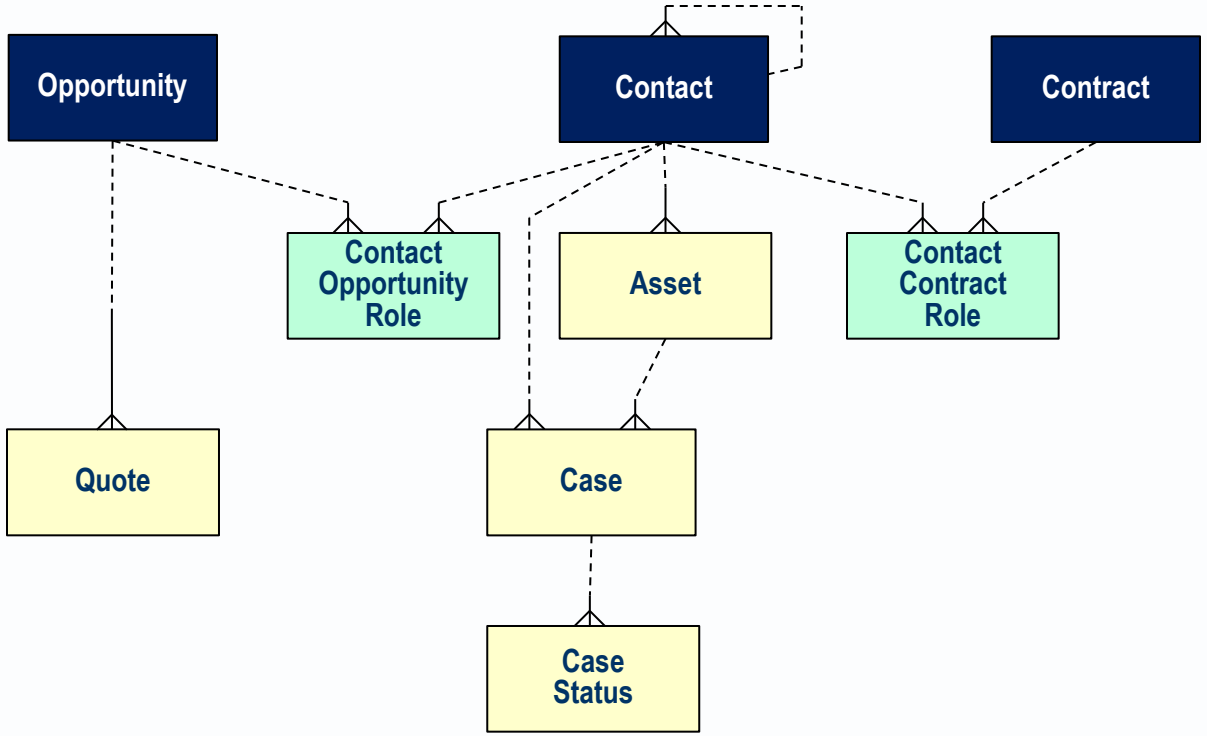
Barker



A naive classification of entity types

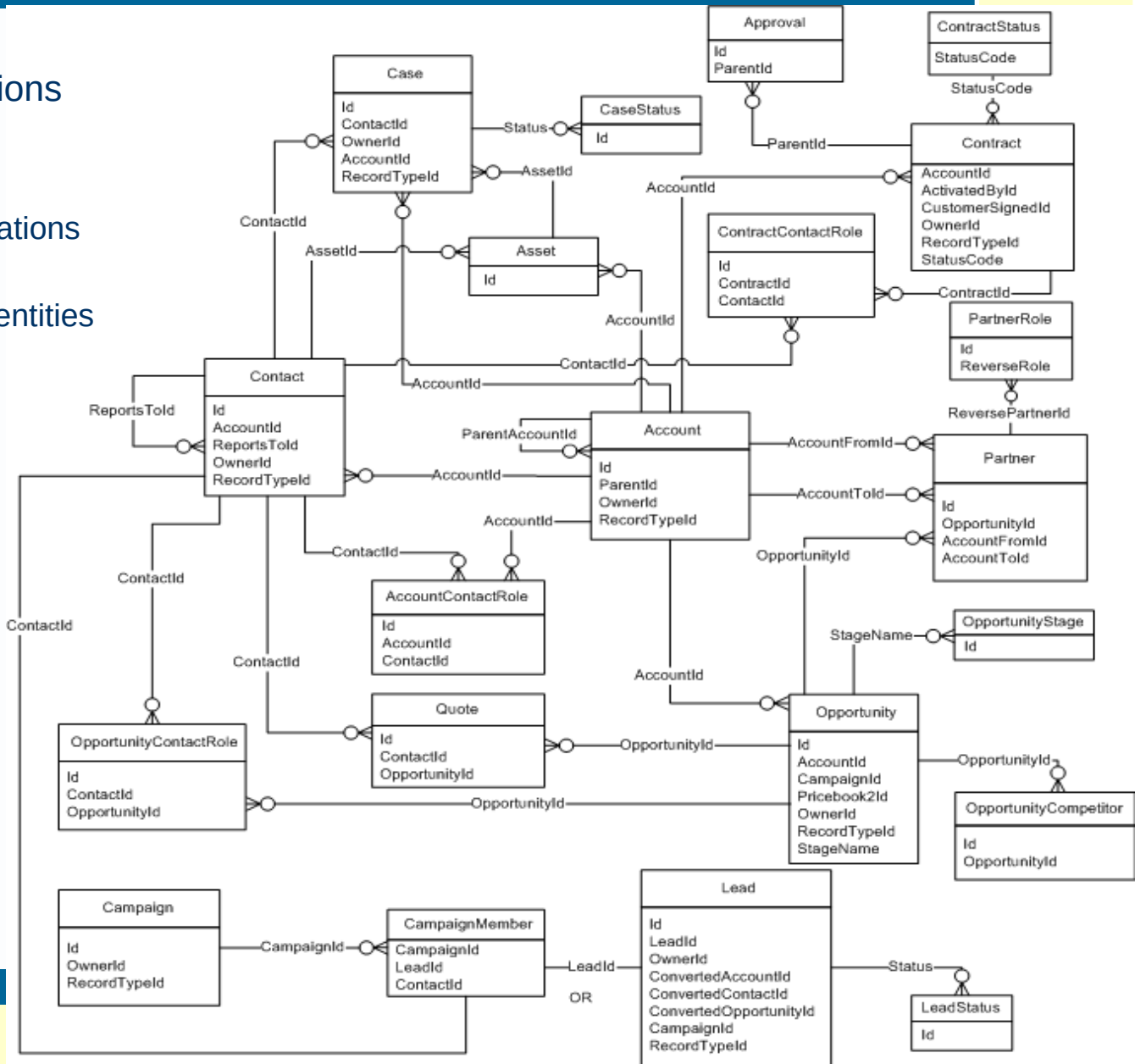
Avancier

Kernel	Kernel entities
Link	Relationships between them
Characteristic	Attributes and characteristic entities



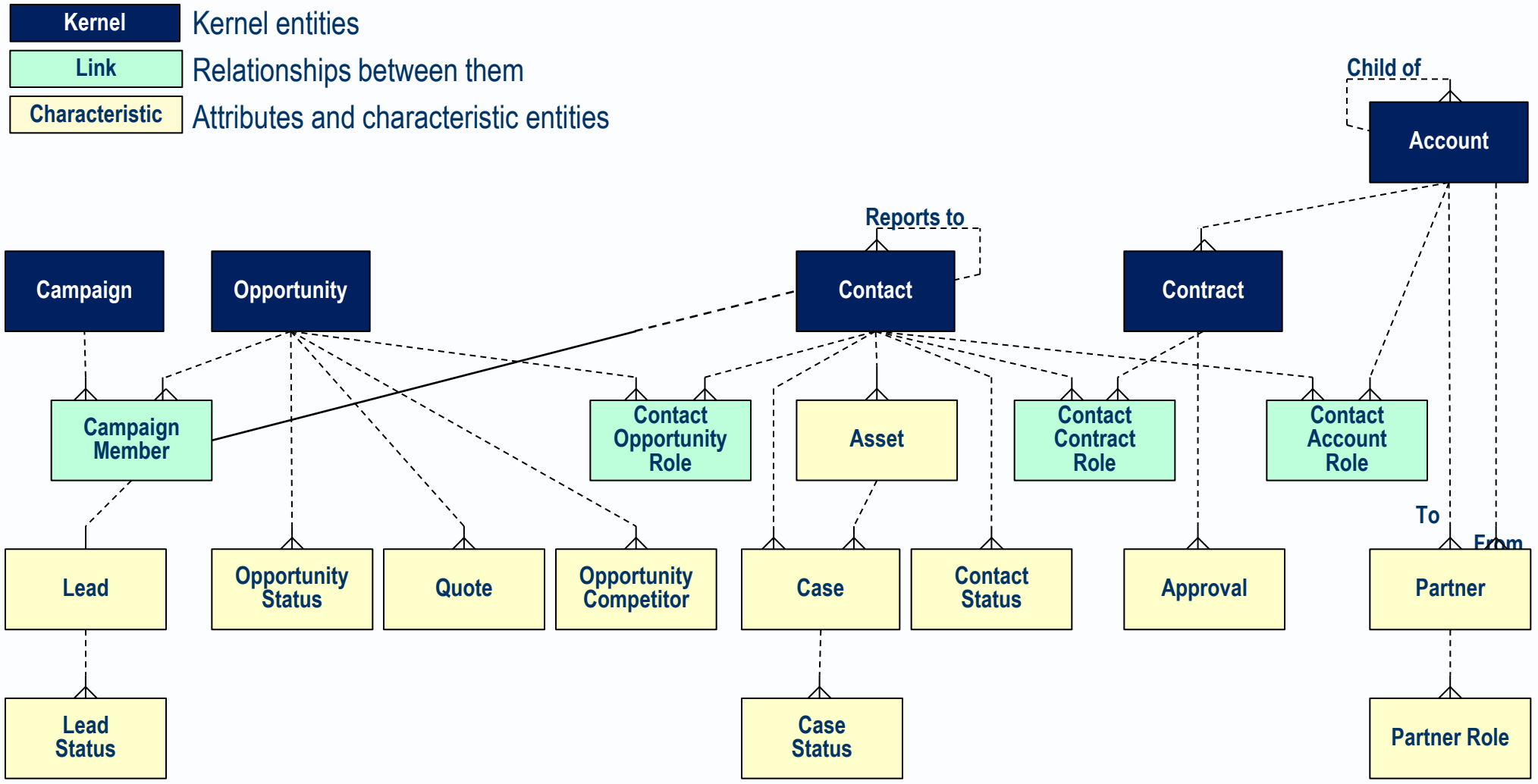
E.g. The core part of Salesforce.com's logical data model

- A network of 1-to-N associations
- Over time
  - Subtypes become roles
  - Aggregations become associations
  - 1-1 associations become1-N
  - 1 to N become N-to-N w link entities



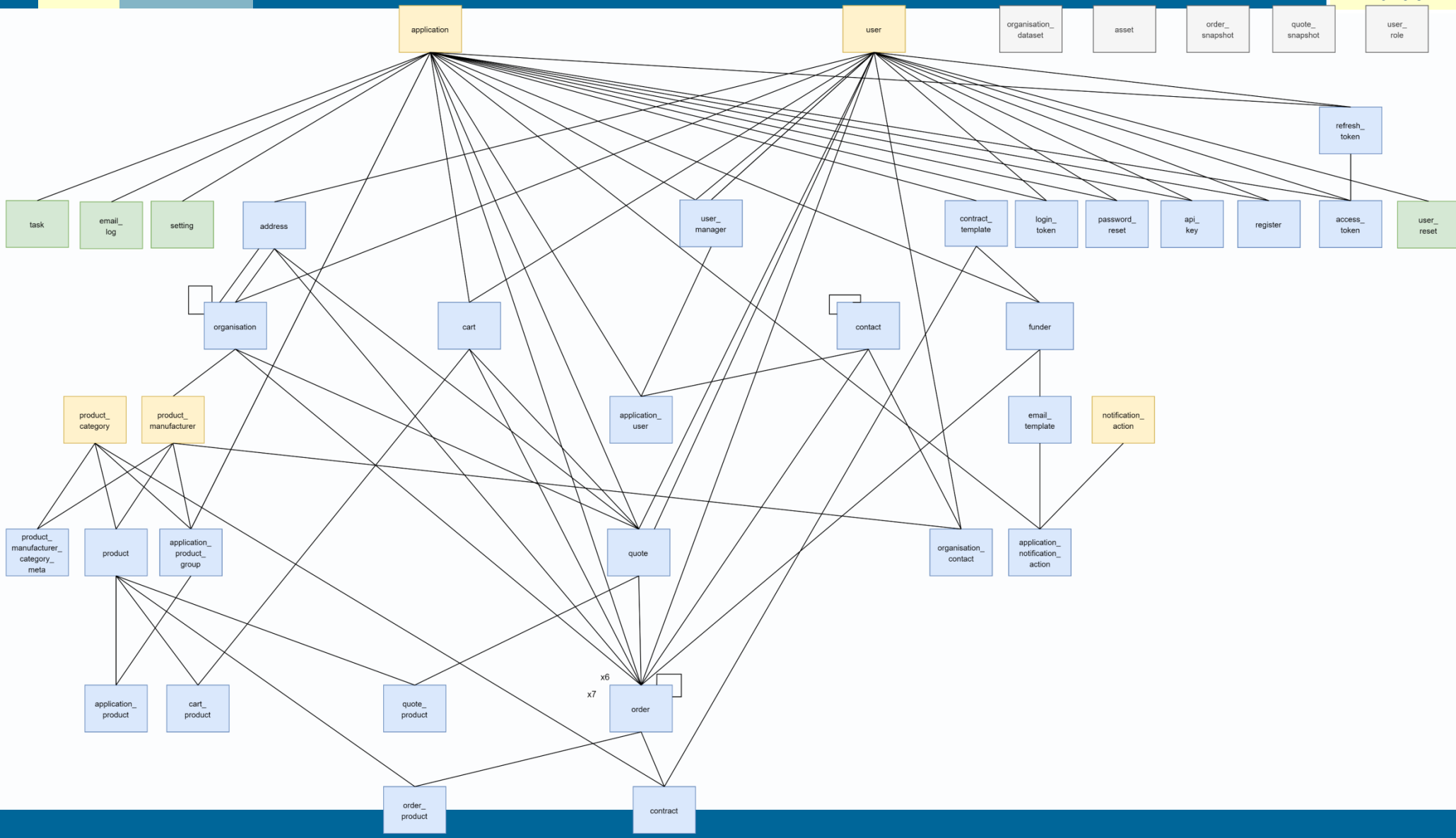
Salesforce.com domain model – redrawn hierarchically

Avancier



Another real world example

Avancier



Avancier

Data flow structures

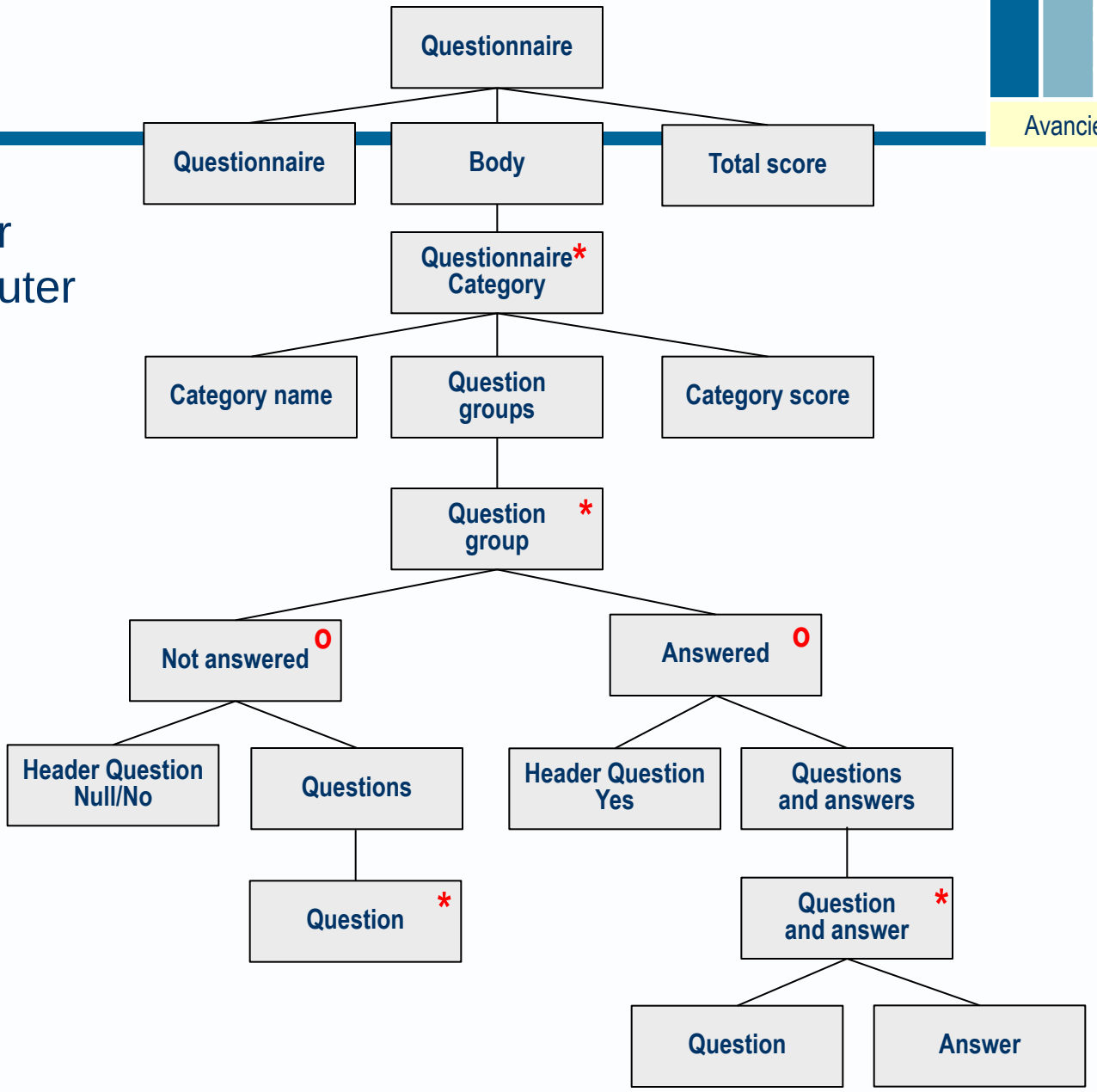
Data structures in data flows

- **Data in motion (data flow)**
- A message, file, form, report, or display in which data passes from a sender to a receiver.

- **Data flow (message) structures**
- A **regular expression** is a hierarchical sentence structure composed from sequence, selection and iteration components

A regular expression

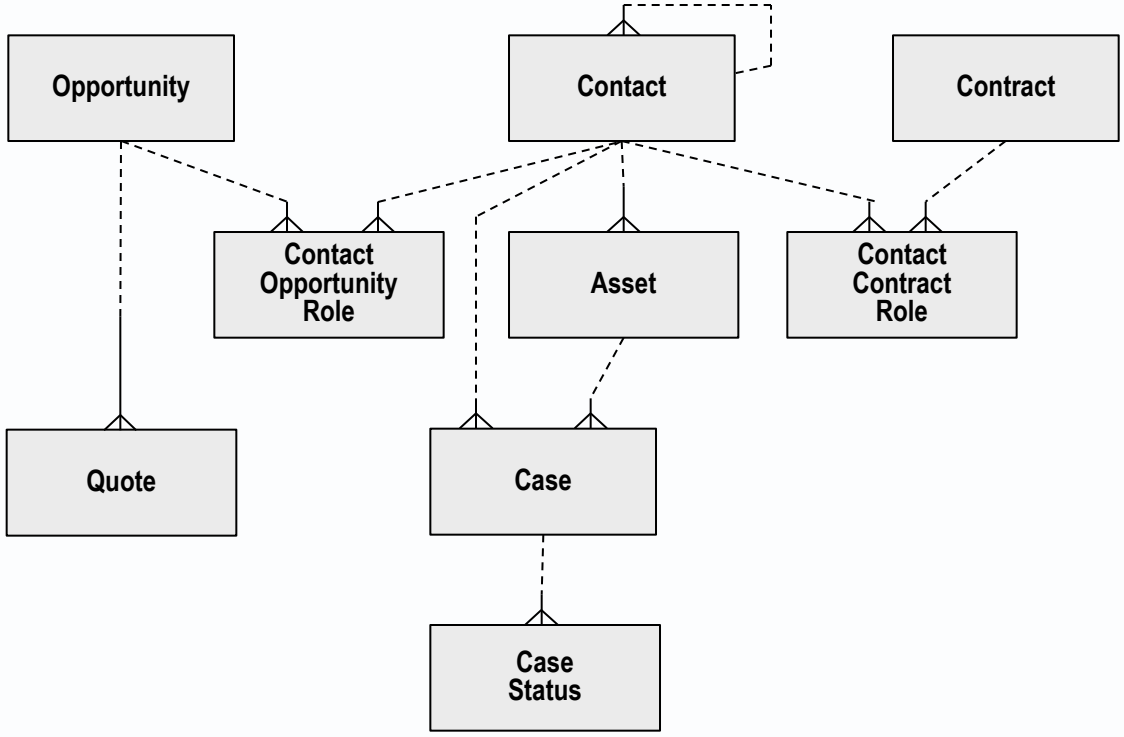
- The formal grammar for data structures a computer can read and write
- A hierarchical structure composed from
  - sequence,
  - selection and
  - iteration
- components



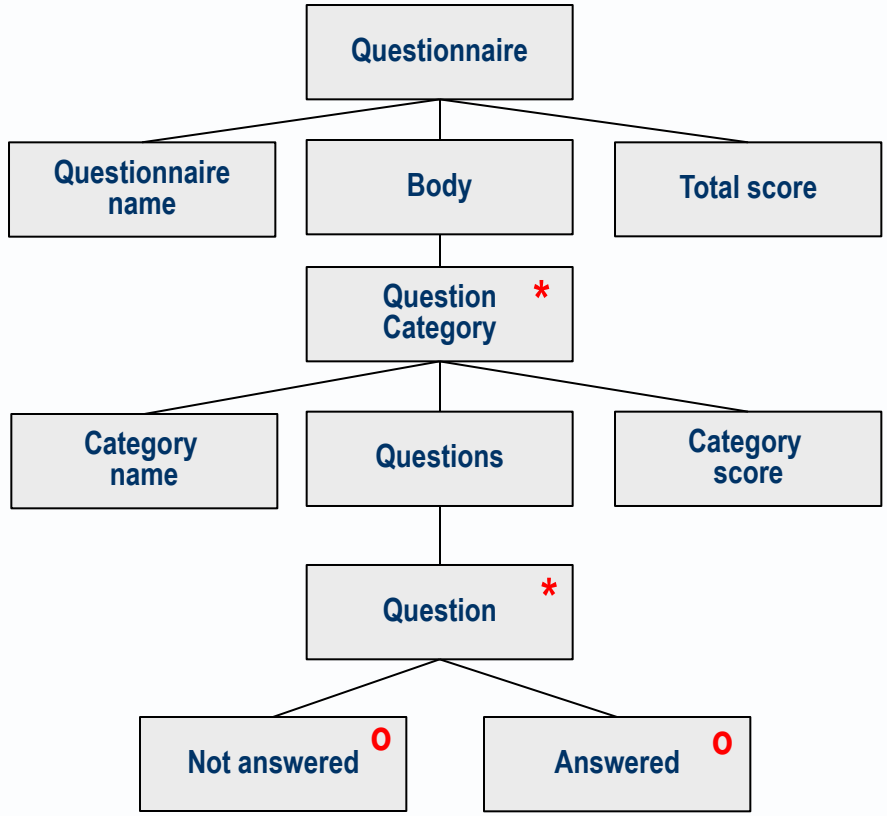
Data structures in stores and flows

Avancier

Data model (in store)



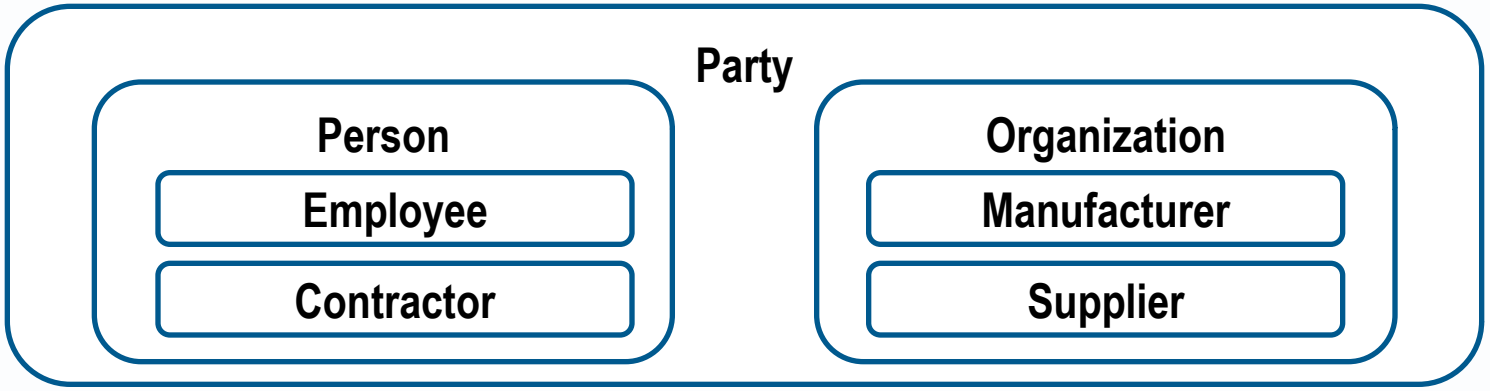
Regular expression (in flow)



Ontologies and Concept graphs

Types as set member definitions

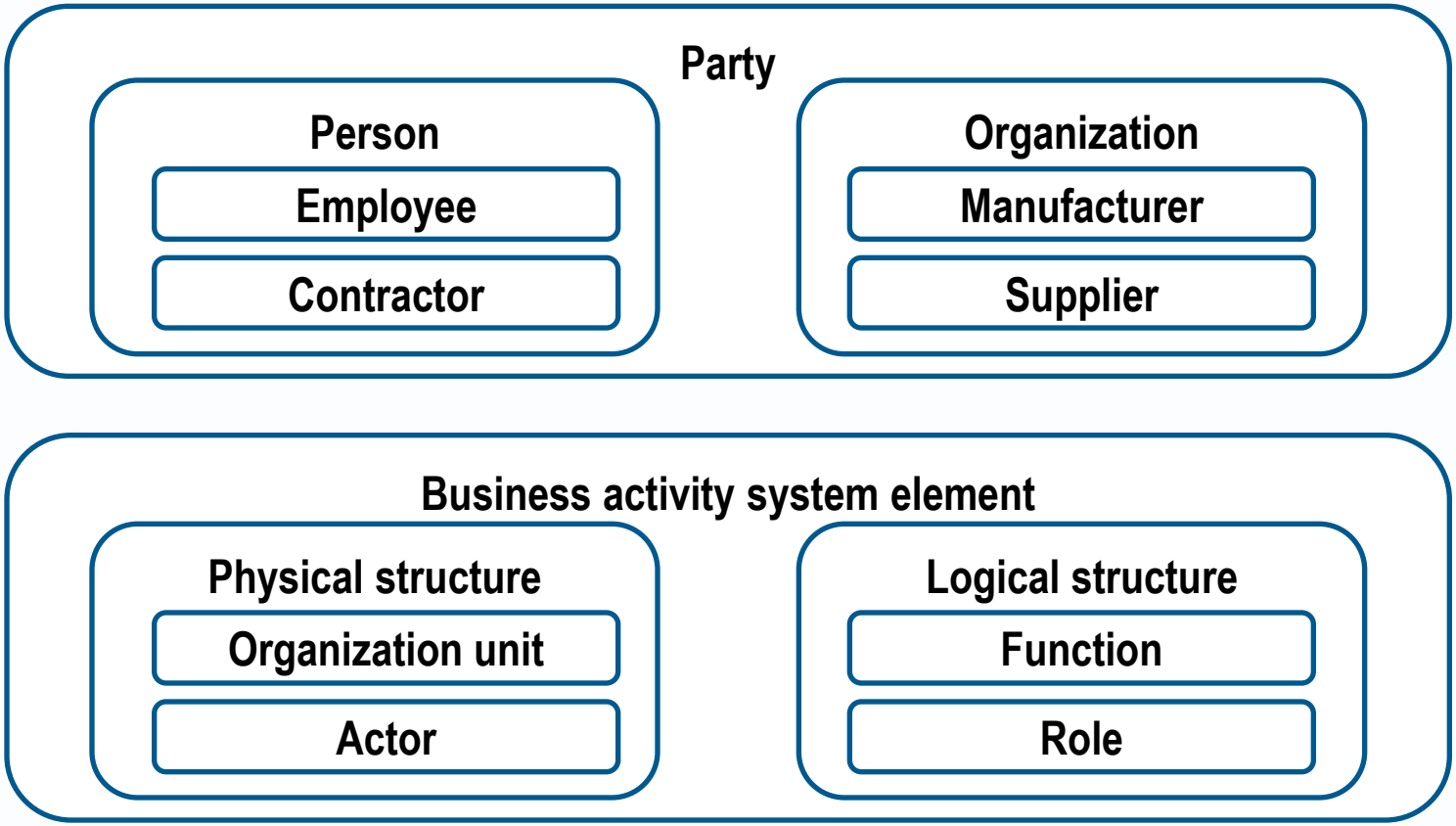
- A type is the "intensional definition" of a set member.
- Naming nested sets in nested boxes (in the manner of Venn diagrams).
  - “On the Diagrammatic and Mechanical Representation of Propositions and Reasonings” in the Philosophical Magazine and Journal of Science.” British logician John Venn.1890

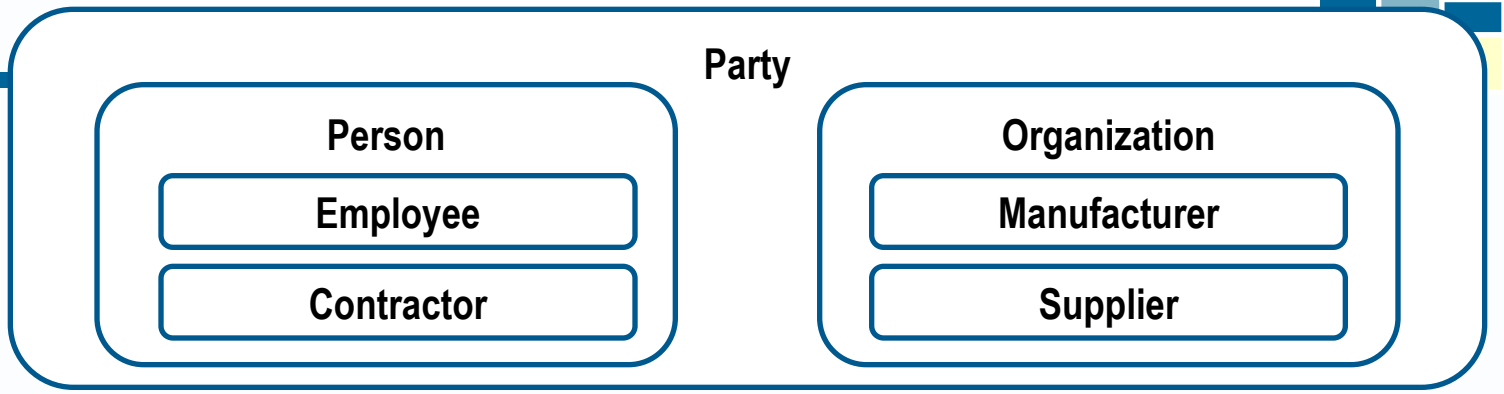


- The roots of this type of diagram go back at least 600 years.

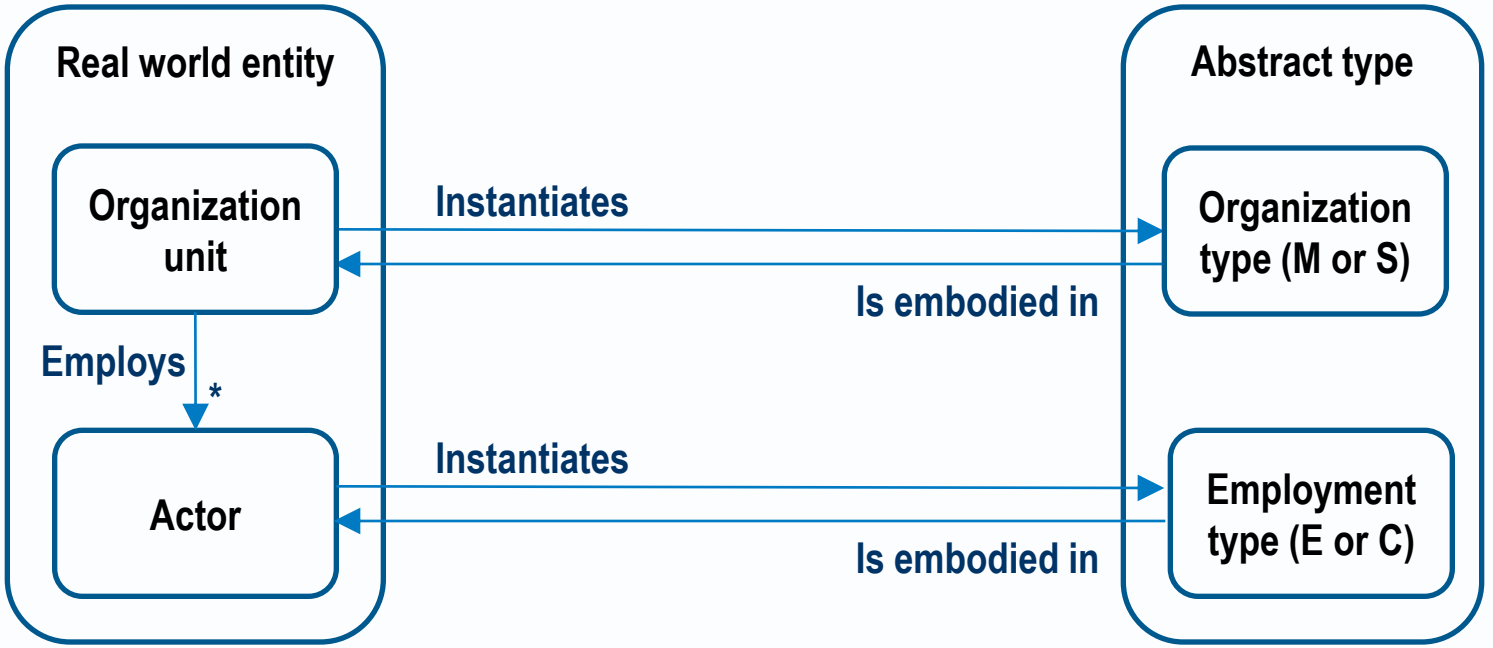
Ontology as a type hierarchy

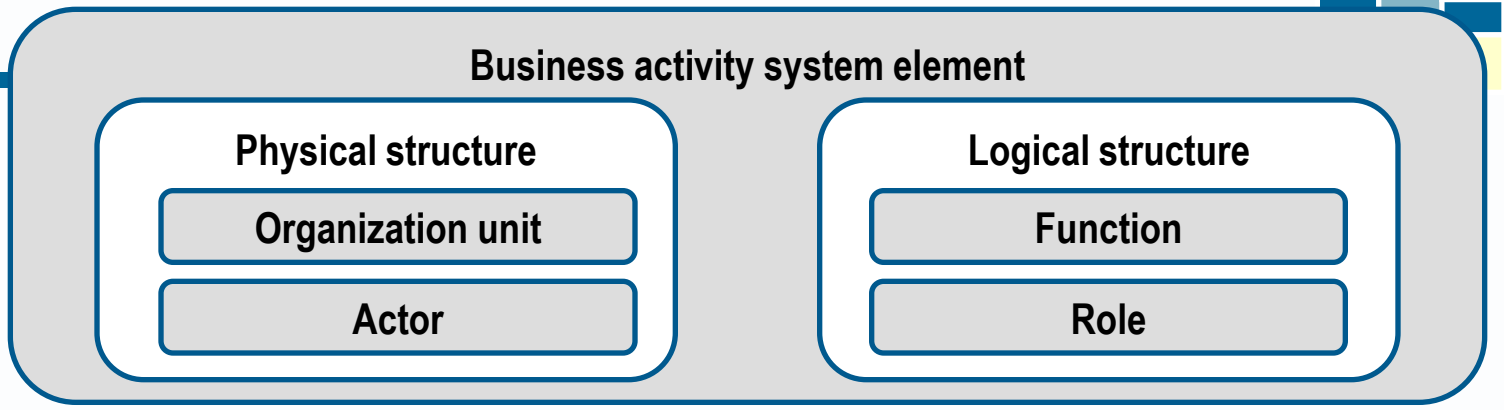
- Type hierarchies are popular, and seductively simple



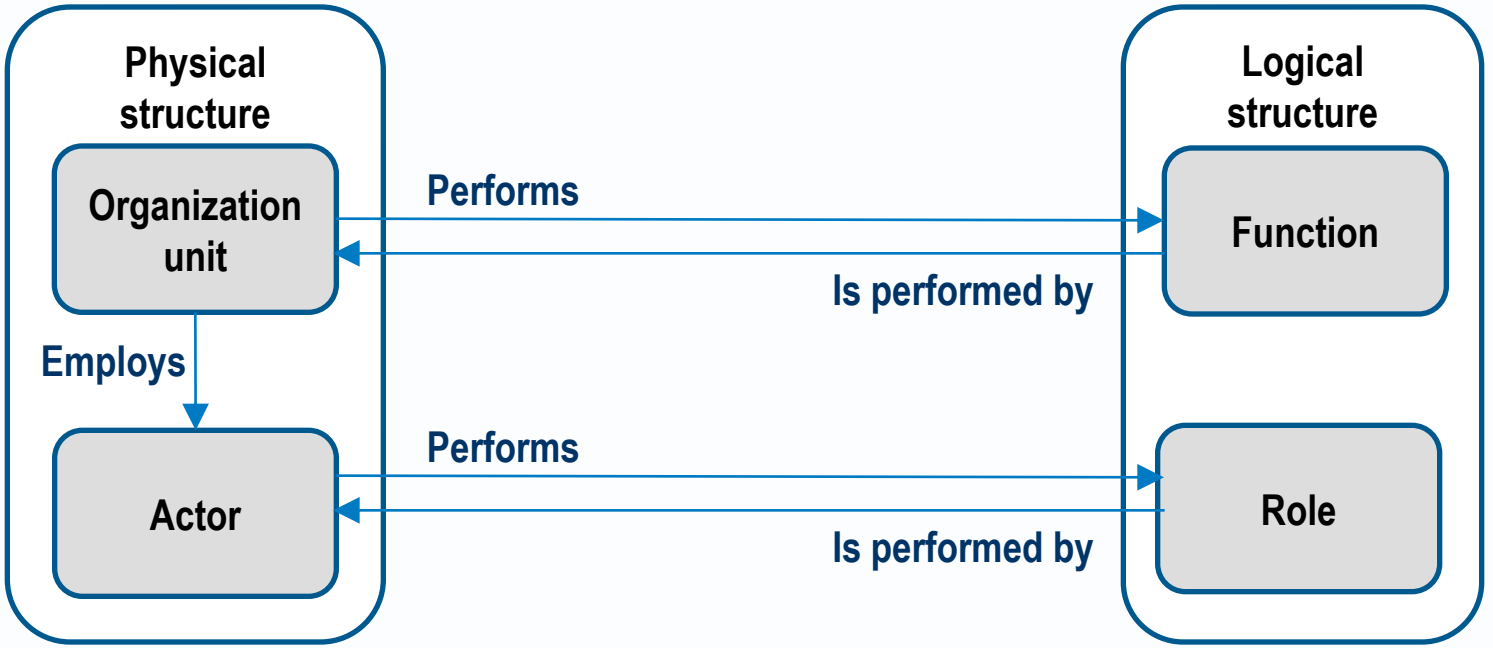


Like most such class/type hierarchies this model is more meaningfully drawn as a concept graph thus



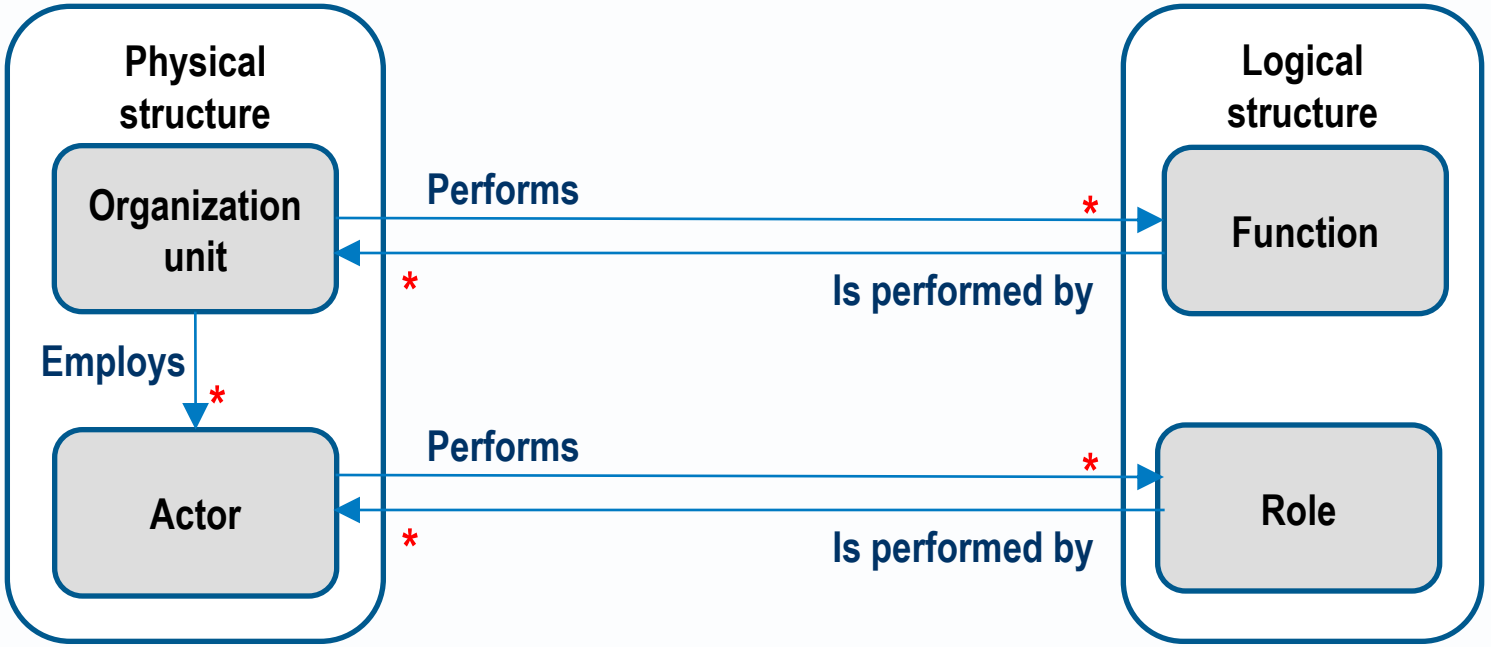


Similarly this class/type hierarchy may be redrawn as a concept graph thus



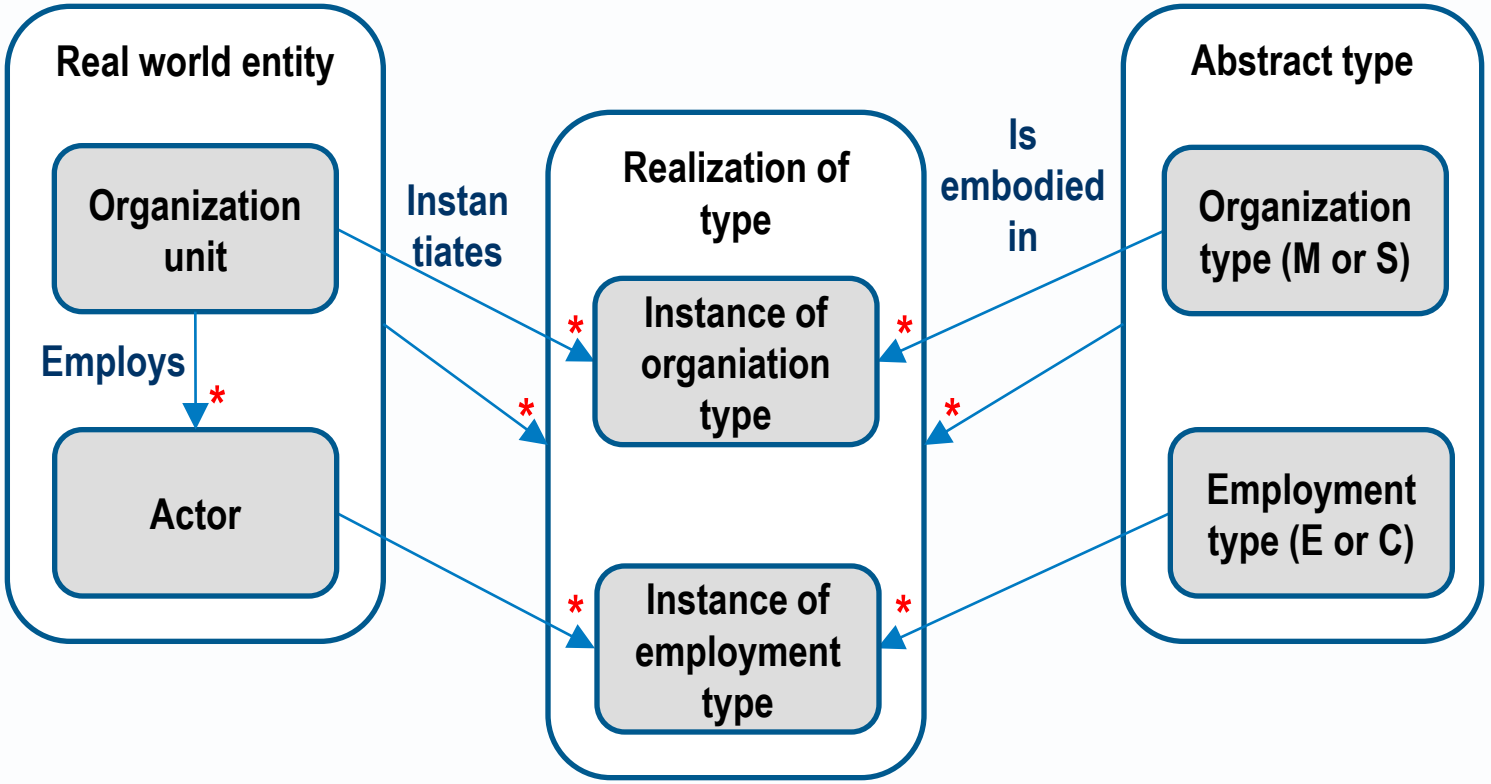
Data store (memory) structures

- An **entity-relationship diagram** is a concept graph with cardinalities added to the relationship lines.
- Note many-to-many relationships



Data store (memory) structures

- Many-to-many relationships are resolved by link entities.



Adding cardinality to relationships

Avancier

- Similarly