Data and Process Modelling

1.Introduction

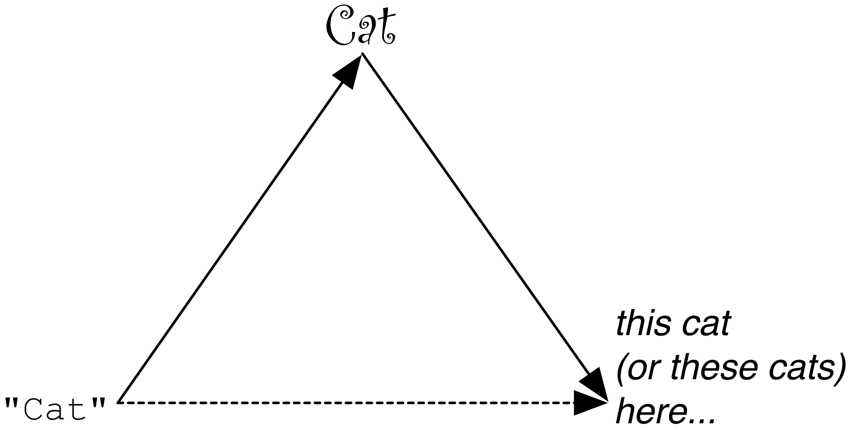
Marco Montali[1]

KRDB Research Centre for Knowledge and Data
Faculty of Computer Science
Free University of Bozen-Bolzano

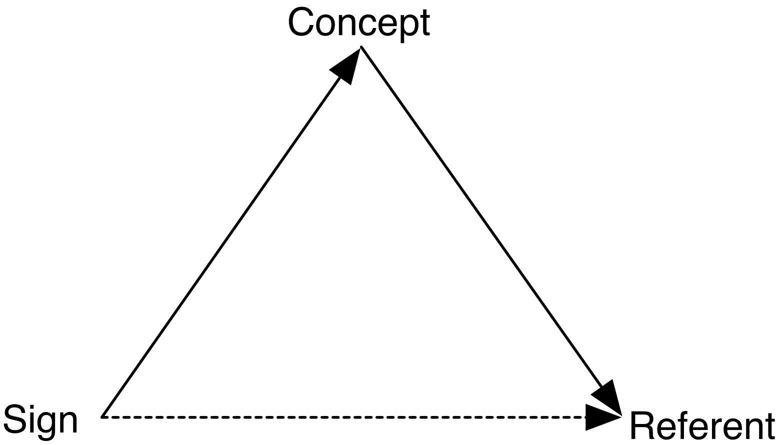
A.Y. 2014/2015

[1] credits to Nicola Guarino

Triangle of Meaning



Triangle of Meaning



Concepts

Concept - Intension - Class (latin *conceptum*: “something conceived”)

An abstract or general idea inferred or derived from specific instances. *(WordNet)*

- It is the part of meaning corresponding to general principles, rules to be used to determine reference.
- We use concepts to ascribe properties and relations to *objects*.

Object - Extension - Instance

Part of meaning corresponding to the effective reference.

Emergence of Concepts

A concept emerges as the result of a process of abstraction and generalization from experience, used by human beings to structure a perception of the domain and talk about it.

- Nietzsche:

  > *Every concept originates through our equating what is unequal. No leaf ever wholly equals another, and the concept 'leaf' is formed through an arbitrary abstraction from these individual differences, through forgetting the distinctions...*

- Called by Kant **a-posteriori concepts**: generated as a result of comparison, reflection, abstraction.

Experience and Conceptualization

Conceptualization

Piece of reality as perceived and organized by an agent, abstracting from a specific situation and the used vocabulary.

Humans isolate relevant invariances from physical reality, using perception, cognition, cultural experience, language.

Concepts in Space and Time

Synchronic level: spatial invariants.

- Unity properties are ascribed to input patterns.
- Emergence of topological and morphological wholes (percepts).

Diachronic level: temporal invariants.

- Objects: equivalence relationships among percepts belonging to different moments.
- Events: unity properties are ascribed to percept sequences belonging to different moments

More in general:

- topological wholes (a piece of coal);
- morphological wholes (a constellation);
- functional wholes (a laptop);
- social wholes (a soccer team).

On Ontology, Ontologies, and Conceptual Schemas

Ontology

The philosophical study of the **nature** and **structure** of **being**, or **reality**, as well as the basic categories of being and their relations.

Studies what there is, without even considering its actual existence.

Ontologies or Conceptual Schemas

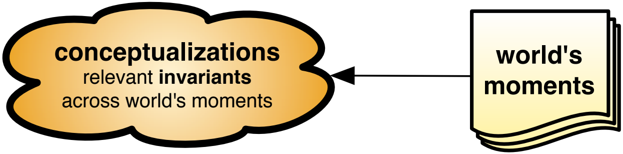
Specific artifacts expressing the **intended meaning** of a **vocabulary** in terms of **primitive** categories and relations describing the **nature** and **structure** of a domain of discourse.
(Guarino)

They are explicit and formal specifications of a **conceptualization** *(Gruber)*.

Conceptual Schema and Intended Meaning

world's moments

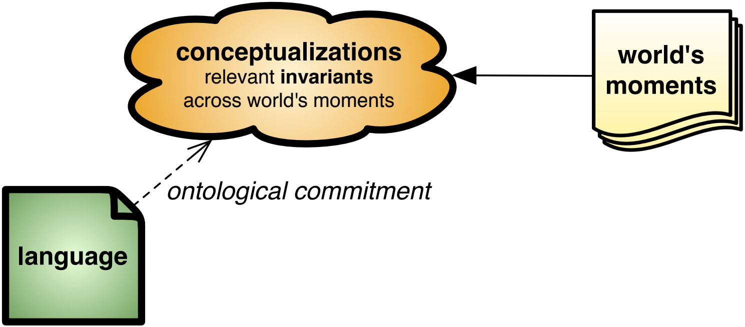
Conceptual Schema and Intended Meaning



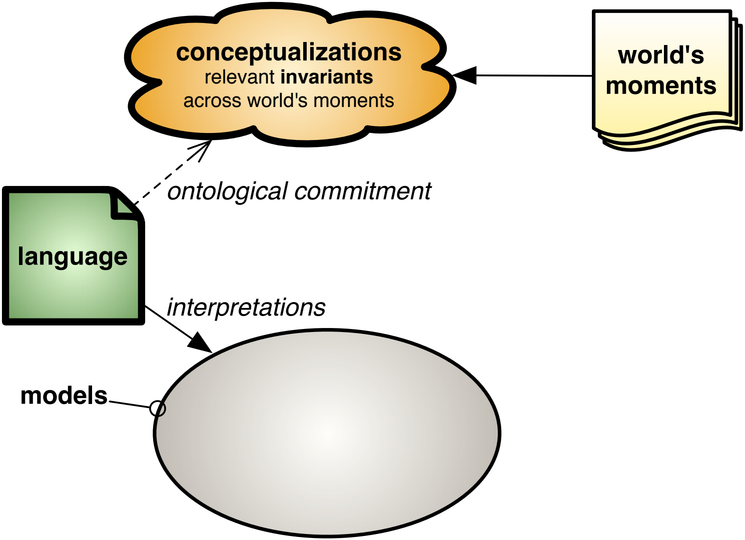
conceptualizations
relevant **invariants**
across world's moments

world's moments

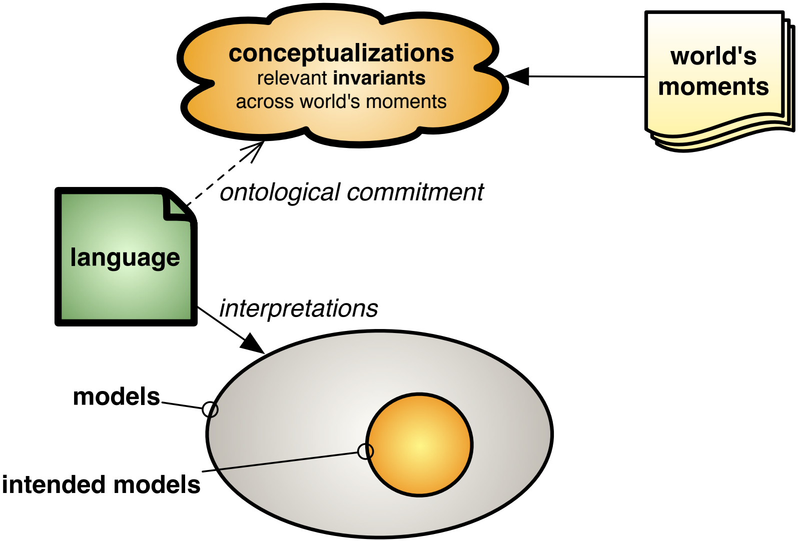
Conceptual Schema and Intended Meaning



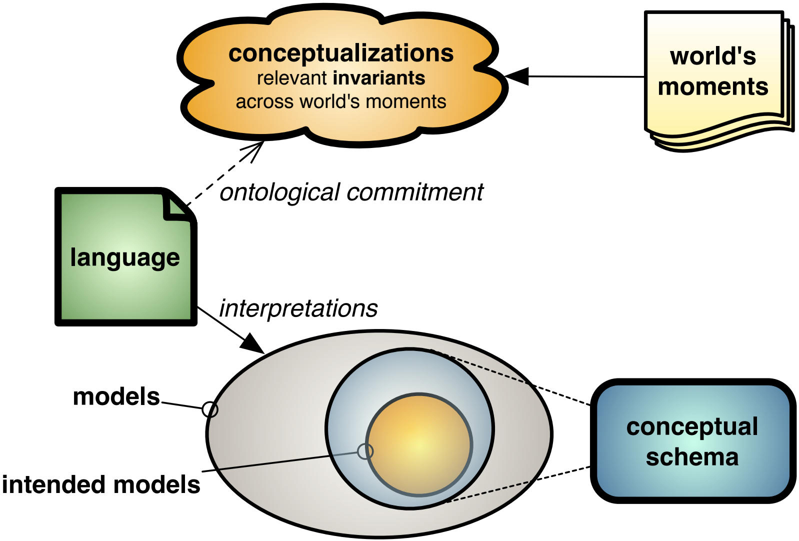
Conceptual Schema and Intended Meaning



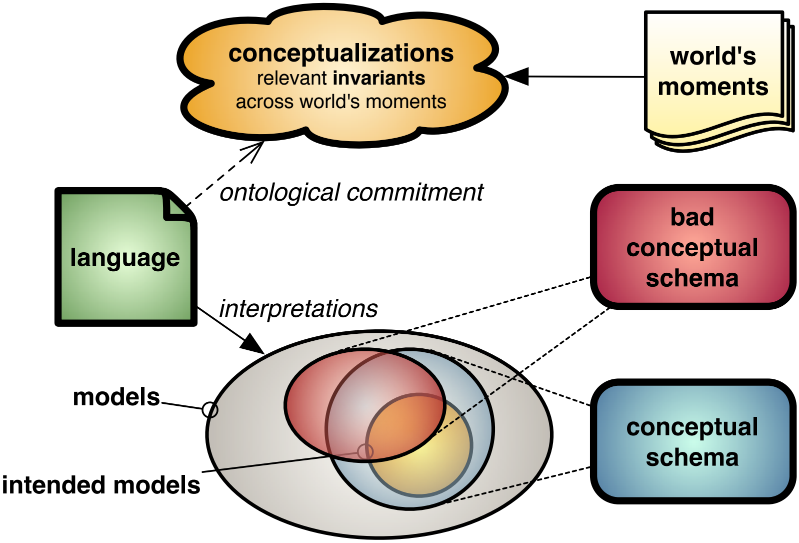
Conceptual Schema and Intended Meaning



Conceptual Schema and Intended Meaning



Conceptual Schema and Intended Meaning



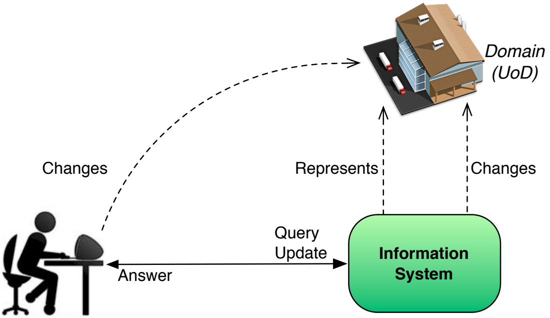
Information System

Information System

A system that collects, stores, processes, and distributes information about the state of a domain to facilitate planning, control, coordination, and decision making in an organization.

- The focus is on *designed* systems, resulting from an engineering activity.
- Refers to the state of a certain domain (UoD - Universe of Discourse).
  - Deals with the semantics of data!
  - Is a fax machine an information system?

Functions of an IS



Memory	to maintain a representation of the state of a domain
Informative	to provide information about the state of a domain
Active	to perform actions that change the state of a domain

Memory Function

IS maintains an internal representation of the state of the domain.

- Intensional level: concepts and constraints describing the structure of the domain.
- Extensional level: set of instances of the concepts described at the intensional level. Much more subject to *change*!

The extensional level is updated so as to reflect those changes that occur in the real world.

- Two update modes:
  1. On request - the users *inform* the system whenever the state changes.
     - ⋆ An operator responsible for the company's CRM
  2. Autonomous - the system *directly observes* the state of the domain and updates its internal state.
     - ⋆ A controller system equipped with environmental sensors.

Informative Function

- IS provides users with information about the state of the domain.
  - Sometimes the IS state mirrors a state that is explicitly present in the domain.
  - Sometimes the state is explicitly represented only in the IS, and it is difficult to observe in reality.
    - ⋆ What about counting the number of nails in a carpentry?
- Two modes:
  1. On request - a user poses a *query* to the IS and receives back an *answer*.
     - ⋆ A manager asking for the number of employees who earn more than 2K euros per month.
  2. Autonomous - a user (pre)defines a *condition* on the state maintained by the IS and is *notified* by the IS every time it holds in the actual, current state.
     - ⋆ An operator who needs to be alerted every time the CPU's temperature exceeds a given threshold.

Informative Function and Queries

- Queries are posed to the IS in order to get information from it.
- Queries and answers must obey to a unique, shared language.
  - Expressivity, complexity, understandability of query languages constitute an entire area of research in computer science.
- Two types of manipulated information:
  - Extensional - refers to the actual population contained in the state of the domain explicitly (o-123-bzFGH, Mario Rossi,. . .).
  - Intensional - refers to the state of the domain implicitly (orders, employees, . . .).
- Extensional queries: ask the IS for specific information about the state of the domain (Who is attending the Conceptual Modeling course? Who accumulated more than 100K purchase?).

  The IS can respond with
  - extensional information (Laura is taking the Conceptual Modeling course), or
  - intensional information (the gold customers).
- Intensional queries: ask for the type of information known by the information system (What is a student?)

Extensional and Intensional Queries

Extensional queries

Ask the IS for specific information about the state of the domain (Who is attending the Conceptual Modeling course? Who accumulated more than 100K purchase?).

The IS can respond with

- extensional information (Laura is taking the Conceptual Modeling course), or
- intensional information (the gold customers).

Intensional queries

Ask for the type of information known by the information system (What is a student?)

Active Function

- IS performs actions that modify the state of the domain.
  - Must be equipped with a description of the actions, their preconditions, and their effects.
  - Preconditions and effects must be defined in terms of concepts represented in the IS.
- Two modes:
  1. On request - a user *delegates* the execution of an action to the system.
     - A bank transaction related to an order's payment.
  2. Autonomous - the system is tuned so that when some condition holds in the state of the domain, the execution of an action is *triggered*.
     - Automatic replenishment of a store.

Functions of an IS and their Modes

	MODES	
FUNCTIONS	**On request**	**Autonomous**
Memory	Change customer address.	Measure temperature.
Informative	Who is the nearest customer considering my current position?	Signal when the temperature is too high.
Active	Notify of the change all consultants working with the customer.	Turn on the heating system when the temperature is too low.

Example of IS: Chess-Playing System

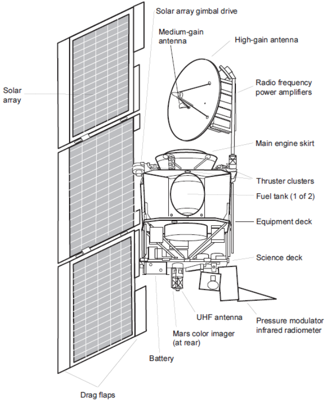
- Domain: board, pieces and their position on the board, legal moves, players, checkmate, . . .
- Memory: configuration of the board, with position of each piece.
- When the human player moves, she must communicate the move to the system, which updates the state of the domain.
- When the system moves, it updates the state and shows the new state to the user.
- The human player can get assistance from the system for the next move.
- When it is the system's turn, it analyzes the current state and decides how to move.

Example of IS: Chess-Playing System

- Domain: board, pieces and their position on the board, legal moves, players, checkmate, . . .
- Memory: configuration of the board, with position of each piece.
- When the human player moves, she must communicate the move to the system, which updates the state of the domain (on request informative function).
- When the system moves, it updates the state and shows the new state to the user (autonomous informative function).
- The human player can get assistance from the system for the next move (on request active function).
- When it is the system's turn, it analyzes the current state and decides how to move (autonomous active function).

Why Data is not Enough

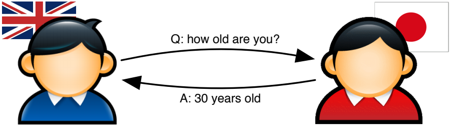
Mars Climate Orbiter



Mars Climate Orbiter spacecraft

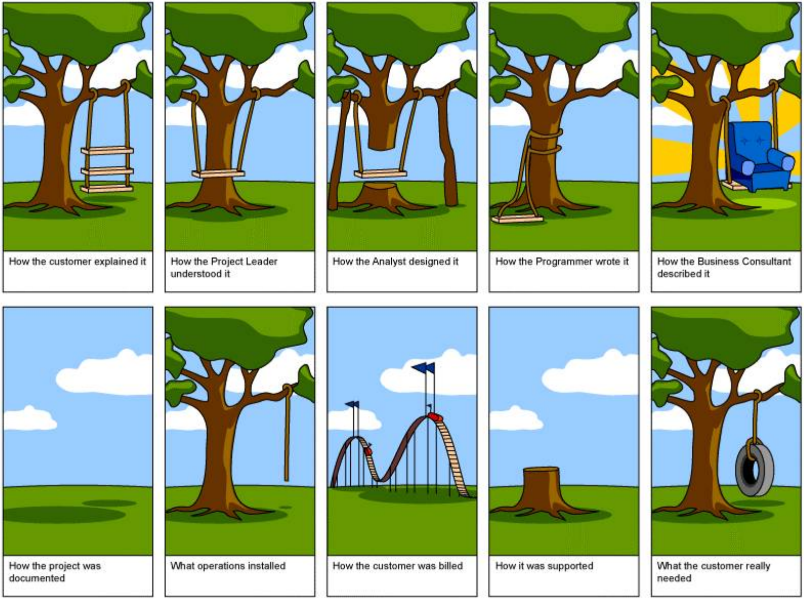
- Developed to study the martian climate and atmosphere.
- Mission cost: \$ 327.6M.
- During the orbital insertion maneuver, it went out of radio contact permanently.
- Why? Metric Mixup.
  - Software on orbiter: Newtons;
    Software on earth: Pound-force.
    Conversion factor: $\sim$ 4.5.
  - Same data, different interpretations.
  - Lack of testing (and budget). Danger of re-use.

Why a Shared Vocabulary is not Enough



- So... 29 or 30 years old?
    - In Japan newborns are considered to be 1 year old.
- Same data, different intended meanings.
- Need of a common languge that talks about a well-defined context (domain) with a well-defined, shared meaning.
- ISs must store information, i.e., data+semantics.

Lack of Conceptual Modeling


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Conceptual Models and Information Systems

To work properly, an IS requires knowledge about its domain and the functions it has to perform.

- It requires a representation of the domain and its state.
  - Conceptual schema and Information base.
- Typically, the state of the domain evolves over time: a representation of change is needed.
  - Conceptual schema contains static and dynamic aspects.
- The representation and evolution of the information maintained in the IS must be consistent with reality.
  - Conceptual schema contains constraints.
- Some information is not explicitly represented, but can be inferred from other information.
  - Conceptual schema contains primitive facts and derivation rules.

Open vs Closed World

Different assumptions on the relations between truth and knowledge.

Closed World

Every true statement is known to be true.
Lack of knowledge implies falsity.
Constraints interpreted as integrity checks over the data.

Databases.

Open World

What we know is only a subset of what is true.
Lack of knowledge does not imply falsity.
Constraints interpret as (intensional) knowledge: used to build "models" of reality starting from what is known, and to infer new information about the domain.

Ontologies, semantic web.

N.B.: also **inconsistency** may arise!

A General Language: Fact Types, . . .

Entity (type)

Concept whose instances are individual, identifiable, objects, possibly existing in the domain.

Relationship (type)

Entity whose instances are *tuples of two or more entities*.

In FOL: entity types → unary relations, relationship types → n-ary (n>1).

A General Language: Fact Types, . . .

Entity (type)

Concept whose instances are individual, identifiable, objects, possibly existing in the domain.

Relationship (type)

Entity whose instances are *tuples of two or more entities*.

In FOL: entity types → unary relations, relationship types → n-ary (n>1).

A General Language: Fact Types, . . .

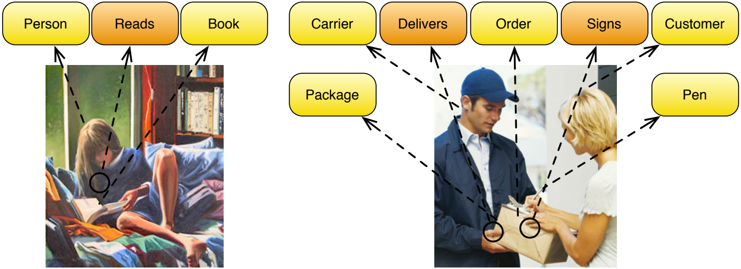
Entity (type)

Concept whose instances are individual, identifiable, objects, possibly existing in the domain.

Relationship (type)

Entity whose instances are *tuples of two or more entities*.

In FOL: entity types → unary relations, relationship types → n-ary (n>1).



. . . and Constraints

- The state of a domain consists of relevant properties that obey to the constraints of the domain.
- Two types of properties:
  - Primitive or elementary facts (the date of birth).
    - Atomic unit of information: cannot be split up into two or more simpler facts without loss of information.
    - No way of modifying or changing just a part of an elementary fact.
  - Derived facts (the age).
    - Obtained from elementary facts through (logical) inference.
- Three types of constraints:
  - Static constraints over the data contained in the state
    → structural conceptual schema
  - Temporal constraints over the allowed evolutions of data
  - Dynamic constraints on the way activities can be executed over time
    → behavioral schema
- We will **not** consider temporal constraints.

. . . and Constraints

- The state of a domain consists of relevant properties that obey to the constraints of the domain.
- Two types of properties:
  - Primitive or elementary facts (the date of birth).
    - Atomic unit of information: cannot be split up into two or more simpler facts without loss of information.
    - No way of modifying or changing just a part of an elementary fact.
  - Derived facts (the age).
    - Obtained from elementary facts through (logical) inference.
- Three types of constraints:
  - Static constraints over the data contained in the state
    → structural conceptual schema
  - Temporal constraints over the allowed evolutions of data
  - Dynamic constraints on the way activities can be executed over time
    → behavioral schema
- We will **not** consider temporal constraints.

A conceptual schema implicitly isolates all permitted states and transitions of the information base.

Static vs Dynamic Constraints

- Examples of static constraints:
    - Functional dependencies (each employee has a fixed salary).
    - (Primary) keys (each employee is identified by its SSN).
    - Multiplicity constraints (a car has exactly four wheels, each square can have at most one chess piece).
- Examples of dynamic constraints:
    - An accepted order cannot be rejected afterwards.
    - To access the cart, the user must successfully log-in.
    - A chess piece can be moved in a square if the move is legal w.r.t. the piece type and does not lead to put the own king in check.
    - When the auction's deadline expires, the bidder with the highest bid must be declared winner of the auction.
    - When the customer closes an order, the warehouse must either refuse it or inform the customer about the expected delivery date.
- Static constraints may implicitly impose dynamic constraints.
    - "Each order refers to at least one item" implies that an item is picked before creating the order.

Conceptual Model

Conceptual Model = Conceptual Schema + Information Base

- Conceptual schema: blueprint of the domain inside the IS.
  - Orders, employees, deliveries, cancelation, customer, gold customer, gift, payment, payment transaction . . .
- Information base (or conceptual database): blueprint of a specific state of the domain inside the IS.
  - Order o-123-bzFGH, employee Mario Rossi, delivery of o-123-bzFGH via airmail,. . .

Conceptual Model

Conceptual Model = Conceptual Schema + Information Base

- Conceptual schema: blueprint of the domain inside the IS.
  - Orders, employees, deliveries, cancelation, customer, gold customer, gift, payment, payment transaction . . .
- Information base (or conceptual database): blueprint of a specific state of the domain inside the IS.
  - Order o-123-bzFGH, employee Mario Rossi, delivery of o-123-bzFGH via airmail,. . .
- We focus on the development of ISs → conceptual schema only.

Principle of Necessity

To develop an IS it is necessary to define its conceptual schema.

Note: with some ambiguity, sometimes

Conceptual model = Conceptual schema.

Structural and Behavioral Schemas

- Structural schema: a specification of the key properties of the domain under study in terms of concepts and their relationships (ontological commitment).
  - Data-oriented perspective: what kinds of data are stored in the information base, what constraints apply to these data, and what kinds of data are derivable.
- Behavioral schema: a specification of the valid changes in the domain state, typically represented by domain events resulting from the execution of actions/tasks.
  - Process-oriented perspective: processes or activities performed to understand the way a particular business operates.
  - Behavior-oriented perspective: how domain events trigger actions.
    - ⋆ Can be understood in terms of the process-oriented perspective.

Components of a Structural Conceptual Schema

Conceptual schema = fact types + constraints + derivation rules

- Fact types: kinds of facts used by the IS to describe a state of the domain.
  - Object types (student, employee, country, . . .).
  - References to objects by value in the information base (matriculation id, SSN, country-code, . . .).
  - Relationship types (studies in, works at, includes, . . .).
- (Integrity) Constraints: restrict the allowed facts used by the IS to describe a state of the domain.
  - Static constraints apply to every state of the domain (every country has exactly one number representing its current population).
- Derivation rules: how to obtain derived facts from primitive facts.
  - Age of a person from her date of birth, ancestor relationship from parent of,. . .

Information Base

Abstract representation of the entities and relationships of a state of the domain, and their classification into entity and relationship types (*facts*). In FOL, entities are constants and facts are ground atomic formulae.

Facts stored in the information base should always be primitive.

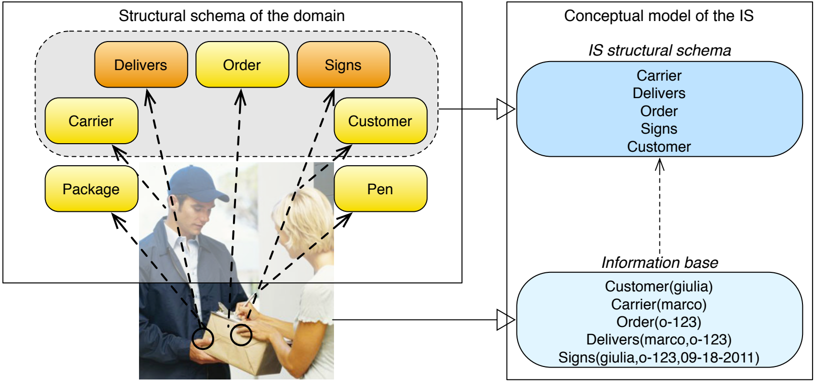
- This simplifies updates, reduces redundancy, helps for consistency.

Information Base

Abstract representation of the entities and relationships of a state of the domain, and their classification into entity and relationship types (*facts*). In FOL, entities are constants and facts are ground atomic formulae.

Facts stored in the information base should always be primitive.

- This simplifies updates, reduces redundancy, helps for consistency.



Structural schema of the domain

Delivers
Order
Signs
Carrier
Customer
Package
Pen

Conceptual model of the IS

IS structural schema

Carrier
Delivers
Order
Signs
Customer

Information base

Customer(giulia)
Carrier(marco)
Order(o-123)
Delivers(marco,o-123)
Signs(giulia,o-123,09-18-2011)

Events and Updates

- In general, the state of the domain maintained by an IS changes over time, and so does its information base.
- Change from $t-1$ to t: the state at t contains at least one different elementary fact from the state at $t-1$.
- Requested by means of a domain event: a set of structural events, each attesting an elementary change.
  - Domain event is an entity, instance of a domain event type!

Transactions

- The domain event becomes a compound transaction: it performs the macro-change if all the elementary updates (or single transactions) can be performed (see later).
- An elementary change affects a single elementary fact. Only two possibilities:
  - addition of a new elementary fact;
  - deletion of an existing elementary fact.
- A successful addition communicates to the IS that some fact is *true* in the current state.

Integrity

- Typically, not all possible configurations of the data are acceptable in a given domain
  - Constraints: physical laws, organizational boundaries, norms, internal policies, . . .
- Perfect world: total integrity of the information base, i.e., information base = state of the domain.
- Total integrity = validity + completeness.
  - An information base is valid if all the facts it contains are true.
  - An information base is complete if it contains all the relevant facts.
- Integrity broken
  - When the IS accepts updates that cannot appear, or are not acceptable, in the domain.
  - When acceptable but *false* data are added to the information base.
- Total integrity can be achieved only by *manual intervention*.
- Partial integrity can be achieved by modeling and enforcing (integrity) constraints.

Factual Consistency

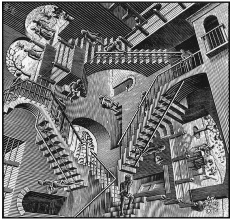
Factual consistency: the information base must *always* satisfy the integrity constraints.

- Enforced by the IS: when a structural event violates some constraint. . .
  1. the corresponding elementary update is rejected;
  2. the corresponding compound transaction is aborted.
- Not all factual errors can be detected by the IS.

Logical Consistency

Logical (conceptual) consistency: the integrity constraints must be (strongly) satisfiable.

- Satisfiable: there must exists one information base $\mathcal{I}$ that satisfies the constraints.
- Strongly satisfiable: $\mathcal{I}$ is nonempty and finite.
- Consider the following conceptual model:
  1. Everybody is supervised by somebody.
  2. Nobody supervises himself.
  3. If x is supervised by y and y is supervised by z, then x is supervised by z.



Why Logical Consistency is Important

Ex falso quodlibet (principle of explosion)

Any statement can be proven from a contradiction.

Any answer can be obtained by querying a logically inconsistent conceptual model.

$$\{\varphi, \neg\varphi\} \models \psi$$

Why Logical Consistency is Important

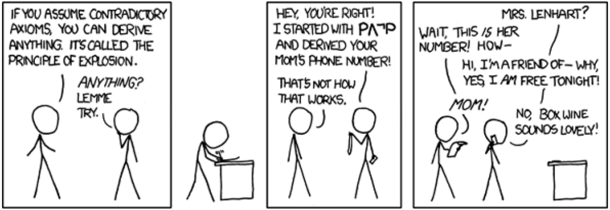
Ex falso quodlibet (principle of explosion)

Any statement can be proven from a contradiction.

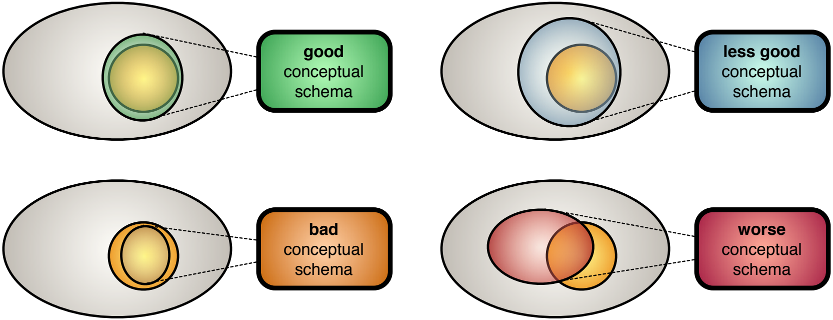
Any answer can be obtained by querying a logically inconsistent conceptual model.

$$\{\varphi, \neg\varphi\} \models \psi$$

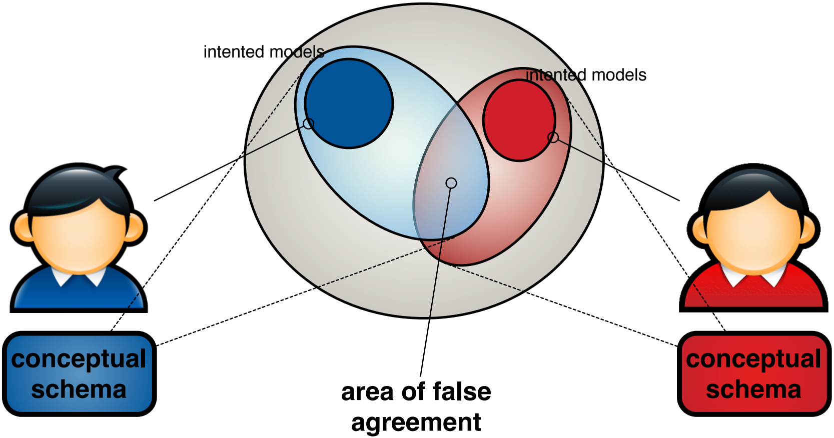
1. $\varphi \land \neg\varphi$ (hypothesis)
2. φ
3. $\neg\varphi$
4. $\varphi \lor \psi$
5. $\neg(\neg\varphi \land \neg\psi)$
6. $\neg\neg\psi$
7. ψ


IF YOU ASSUME CONTRADICTORY AXIOMS, YOU CAN DERIVE ANYTHING. IT'S CALLED THE PRINCIPLE OF EXPLOSION.
ANYTHING? LEMME TRY.
HEY, YOU'RE RIGHT! I STARTED WITH P∧¬P AND DERIVED YOUR MOM'S PHONE NUMBER!
THAT'S NOT HOW THAT WORKS.
MRS. LENHART?
WAIT, THIS IS HER NUMBER! HOW–
HI, I'M A FRIEND OF– WHY, YES, I AM FREE TONIGHT!
MOM!
NO, BOX WINE SOUNDS LOVELY!

The Need for Validation



The Need for Validation, and the Importance of Precision



N.B.: Precision can only be defined w.r.t. a language. Whether the language is good or not is another story!

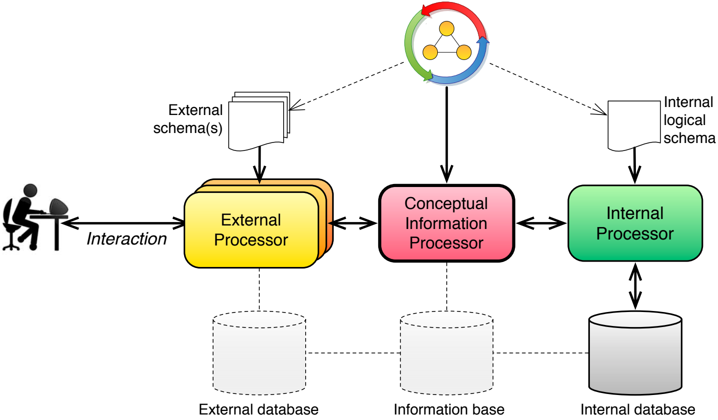
Business Processes and PAISs

Business Process (BP)

A BP consists of a set of activities that are performed in coordination in an organizational and technical environments. These activities jointly realize a business goal.

- IS with explicit BP support:
  - Process-aware IS: stores and executes BPs that manipulate the information base according to the dynamic constraints.
- IS without explicit BP support:
  - Processes hidden into software components that manipulate the information base.
  - Dynamic constraints must be reflected in the program.
  - IS understands the manipulation in terms of domain events and compound transactions.
  - The structural schema must find a counterpart in the program.
    - ⋆ Transient information model.

ISO Abstract Architecture of an IS



- Conceptual information processor: enforces that the evolution of the information base conforms to the conceptual schema.
- Dedicated architectural layers to manage user's interaction with the IS and the actual manipulation and storage of data.

Presentation layer

Manages the interaction with external users.

- External schema: a view of the conceptual schema in terms of concepts and operations accessible to a particular group of users.
  - What information can be accessed: read, access, deletion.
  - How this information must be presented to the users.
- External database: virtual database representing the state of the domain in terms of the external schema.
- External processor: exchange messages with users and enforces the prescriptions of the external schema.
  - Defines a language for communicating with the users.
  - Acts as a bridge/translator between users and the conceptual information processor.
- In general, multiple external databases/schemas/processors to deal with different groups of users.

Logical and Physical Layers

Manage the internal manipulation of data and their effective physical storage.

- Internal (logical) schema: expresses the conceptual schema in terms of the abstract data structures and operations supported by a concrete logical model.
    - For structural information: relational, object-oriented, ...
    - For behavioral information: executable process/program.
- Internal database: physical, internal storage for the actual data.
    - For relational data model: managed by a DBMS.
    - Conforms to the logical schema, realized as a physical schema (e.g., written in the specific DBMS language).
    - Focus on efficiency and conciseness.
- Internal processor: receives the commands from the information processor and executes them over the internal database.

Conceptual Layer

Governs the IS at the conceptual level.

- It is completely independent from user interfaces, storage and data access techniques: stability.
- Conceptual information processor: mediates the communication between the external users and the internal database.
    - Ensures factual consistency.
    - Transforms domain events into compound transactions over the actual data.
- Remember: the information base is virtual!

Conceptual Information Processor

- Metalanguage: language used to study a language.
- Metaconceptual schema: schema that specifies the design rules to be satisfied by conceptual schemas. Fixes the language used to develop conceptual schemas.
  - E.g.: a relationship type is a fact type that relates at least two entity types.

Conceptual Information Processor

- Metalanguage: language used to study a language.
- Metaconceptual schema: schema that specifies the design rules to be satisfied by conceptual schemas. Fixes the language used to develop conceptual schemas.
  - E.g.: a relationship type is a fact type that relates at least two entity types.

Stages of the conceptual information processor:

1. (**Modeling**) The modeler enters the conceptual schema into the IS. The information processor checks whether it is consistent with the metaconceptual schema. If so → goto 3, else → rejected.
2. (**Update**) A user send a domain event to the IS. The IS tries to execute the corresponding compound transaction.
   - If after the application of *all* elementary updates, the resulting information base is consistent → transaction committed, else → transaction aborted (roll-back).
3. (**Query**) A user queries the IS about the domain. The information processor supplies information about the conceptual schema or information base, if it has the info or can derive it.