Data modeling among non-programmers

Mads Hjorth
Danish Commerce and Companies Agency

Background

I live and work in Copenhagen, Denmark

Educational from Roskilde University with degrees in Educational Science and in Computer Science

10 years of building computer systems, from one-man jobs to +50 programmers projects

Teaching ‘about computers’ to diverse groups, from immigrants on welfare to graduate students

Extensive experience in practical group activities from studies, scouting and work

Strong interest for “learning by computers”

madsh@me.com

The views expressed in this presentation are the views of the speaker and do not reflect the opinion of previous or current employers

Data models I have worked with

Functional genomics
a worldwide database of patients carrying balanced chromosomal rearrangements to identify functions of affected genes.

What genes could be involved in male infertility?

What is the biological function of the gene DYX?

Chromosome bands i.e. 1p21
Karyotypes i.e. 46,XY,t(1;8)(p21;q24.21)
Traits i.e. broad nasal bridge (ICD10)

Danish government data
various national registers and digital self service solutions e.g. vehicle and business registration.

Is the car with license plate YZ 56 981 covered by a required insurance?

What new bakeries opened up in Copenhagen last month?

Vehicle Identification Numbers, Car models
Business types, European Business Register
Trans-organizational business processes

Danmarks Grundforskningsfond Danish National Research Foundation

UNIVERSITY OF COPENHAGEN

SKAT

virk.dk

CVR

... to the point

Data modeling is – in practice – a multidisciplinary and group-based activity, that leads to a symbolic (computable) representation of selected aspects of a domain of knowledge.

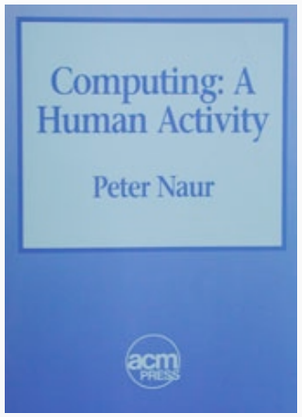
Some basic and very useful symbolic constructs are difficult to understand, especially for participants that do not share experiences from years of programming.

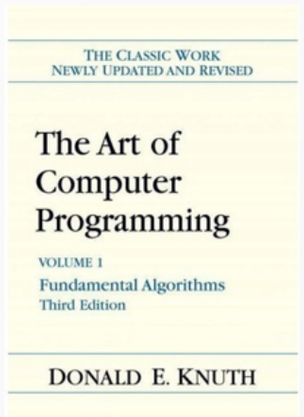
One way to reduce the difficulties may be to reframe the constructs into more common experiences closer to our natural intuition.

Multidisciplinary, but two recurrent disciplines...

Datalogy

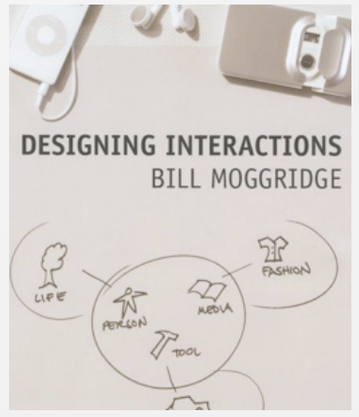
Study of the nature and uses of data.





Design

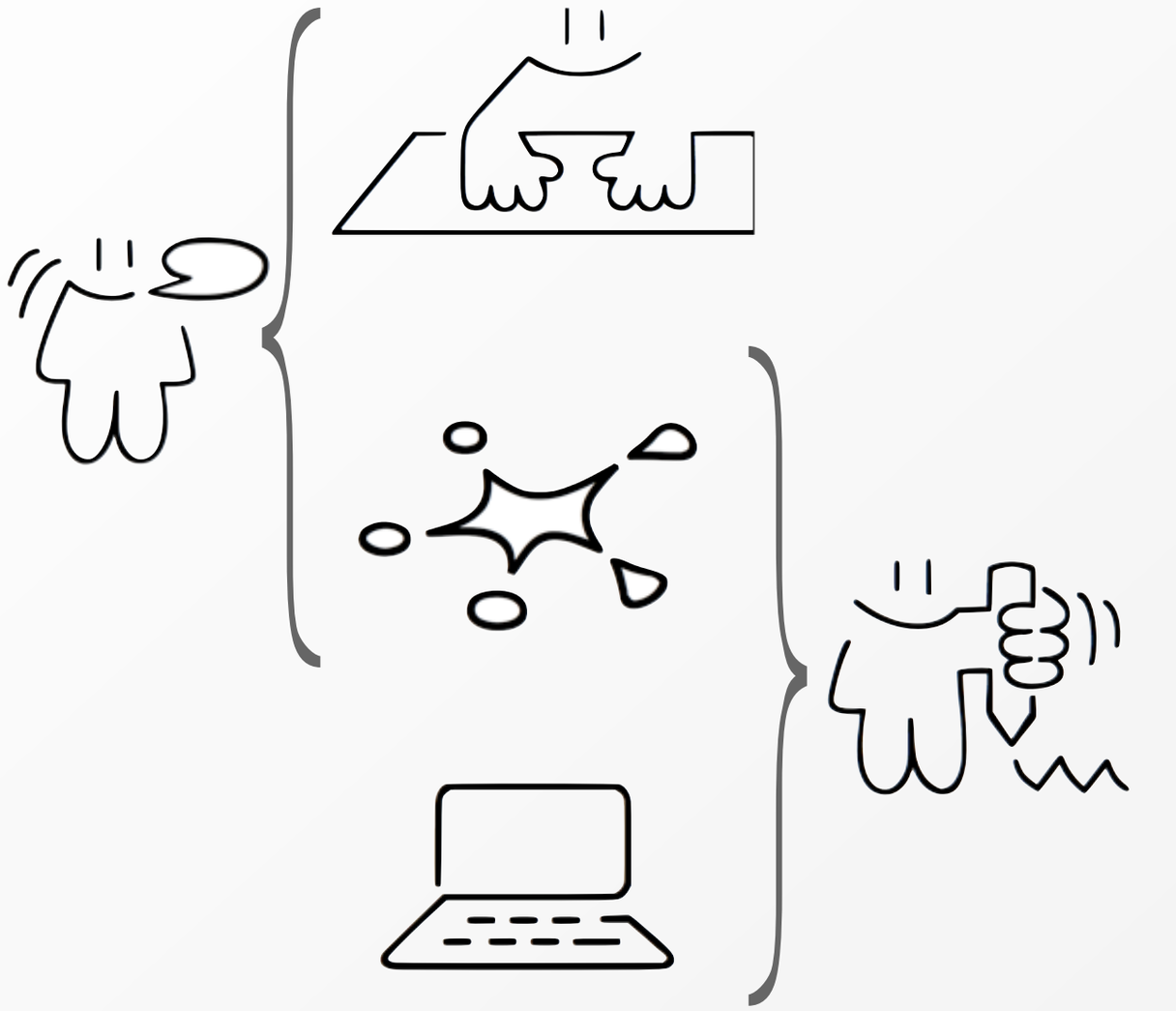
Finding practical solutions to meet a need.



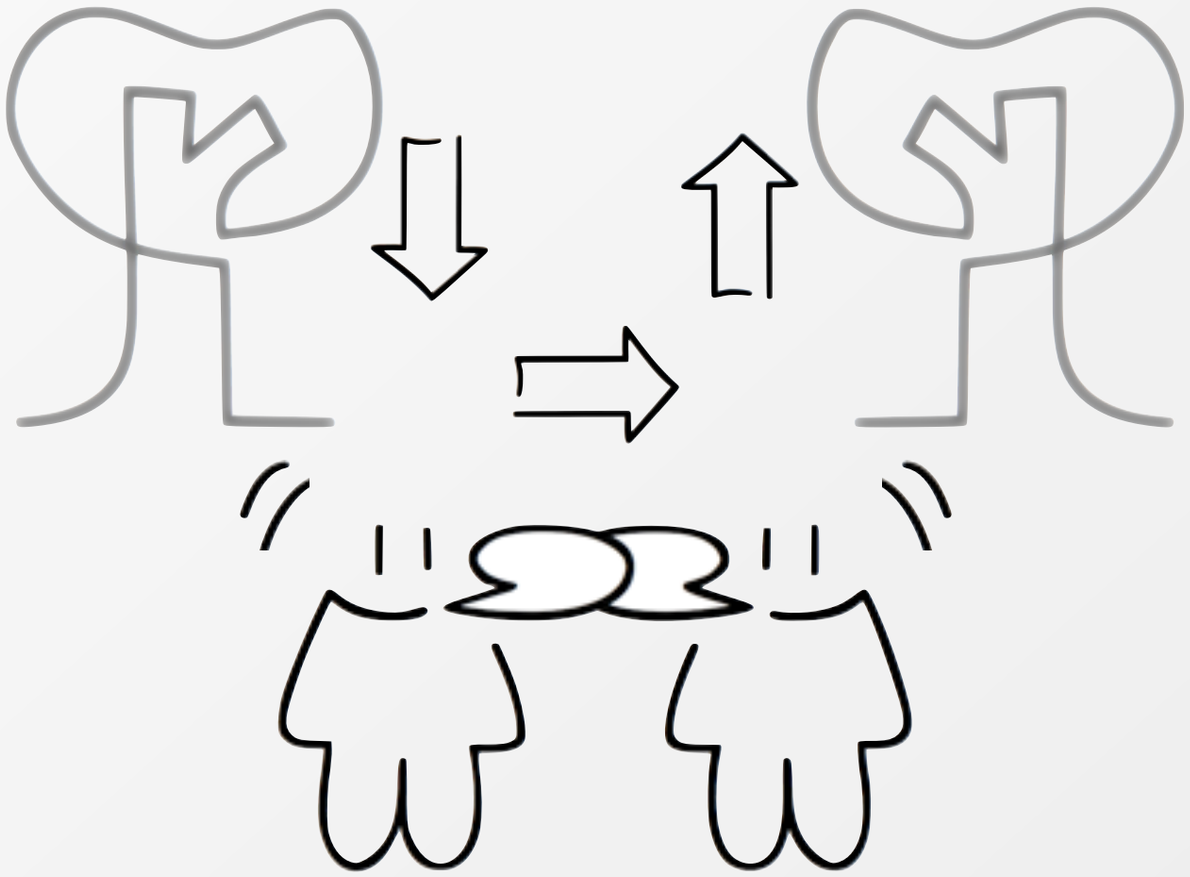
How little does every participant in the data modeling activity need to know about datalogy and design?

One discipline is being spread out to primary schools, colleges, universities and even day care... the other not so much.

Data models as symbolic representations



Data model is a language shared between domain experts, programmers and others...



...and is communicated through limited bandwidth with the intend to align ideas.

"**Human characterizations of reality are built out of a recognizable inventory of thoughts.**"
Steven Pinker: The Stuff of Thought, 2007.

Symbolic constructs as expressed in code

Functions

```
boolean validate(Company data) {
    ...
    return true;
}
```

Relational data

```
SELECT firstname, lastname FROM owner
LEFT JOIN car ON car.cid = ownership.cid
LEFT JOIN person ON person.pid = ownership.pid
WHERE car.licenseplate
```

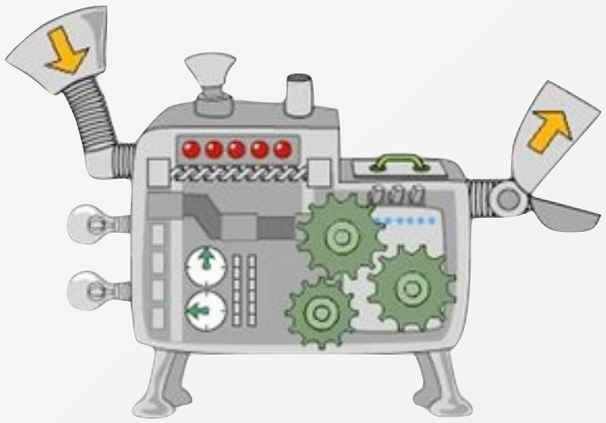
Types and inheritance

```
class FederalSavingBank extends BusinessEntity
                        implements FederalIncorporated
{
    ...
}
```

...but this is all based on the practice of programming.

Functions – examples

- *Find the patient most similar to patient X.*
- *Validate the request for registration of business X.*



Our elementary school understanding is inadequate, mainly because it is,

- dealing with numbers, not `x` or `x.y[].z`
- consumes the input to create a result
- hides some “global” inputs and “state-ness”
- hides away computational complexity

...and our practical problems are often,

- is the input structured enough?
  e.g. wellformed addresses
- is the background data available?
  e.g. existing real address
- how is it done?
  e.g. similar as most shared terms for traits

Functions – in a different perspective

Functions are not part of our natural mental inventory and we have very diverse experiences with them.

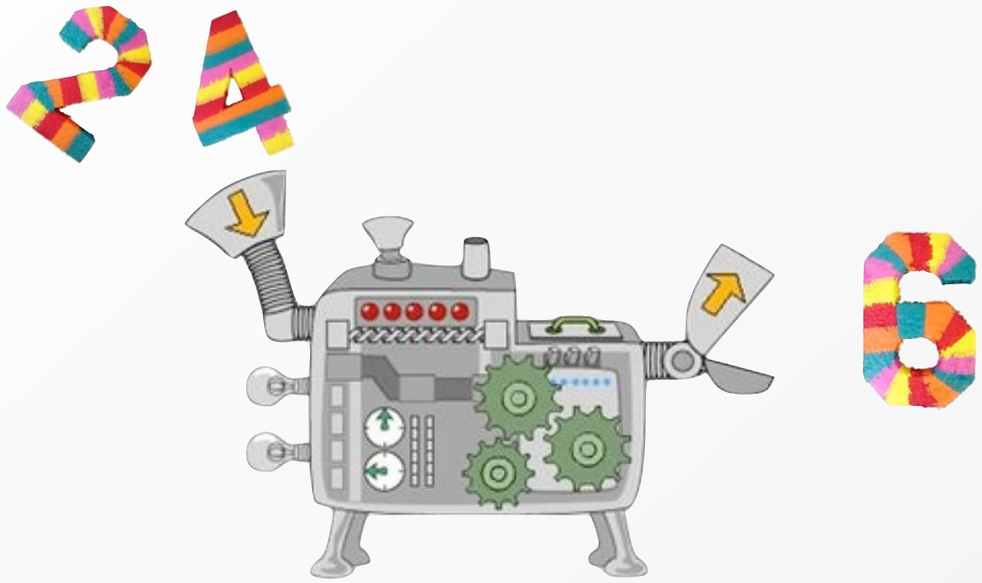
They might be understood by combining **events, states, things and goals** where events can be seen as **causing, enabling or preventing** others and things can be articulated into **parts**, but changes states as a **whole**.

When objects are visualized they are placed in a continuous space along with events.

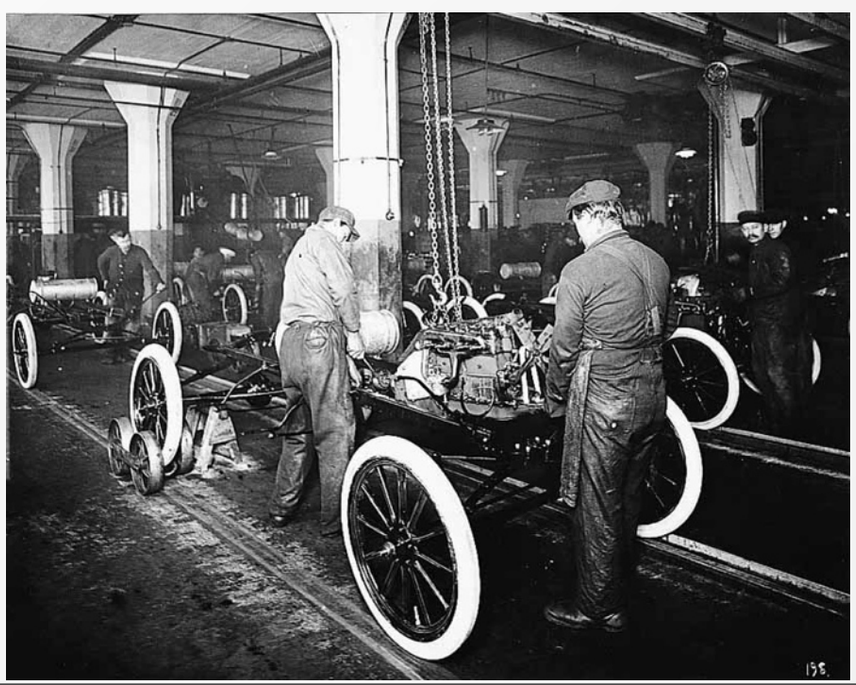
Functions might be understood as... **prototypes of actions performed with the intend to change the state of an object based one the state of other objects.**

Functions - revisited

not the magical plus machine...



...but a station at the assembly line



- ~~*Find the patient is most similar to patient X*~~
- *For a patient record in the register, go through all existing records and add a line to the record about how many terms they share with each other.*
- ~~*Validate the request for registration of business X*~~
- *A registration form can be rejected if the approval stamp is missing. The stamp is used only when all required fields are filled, and the fields match those of the address register.*

Relational data – examples

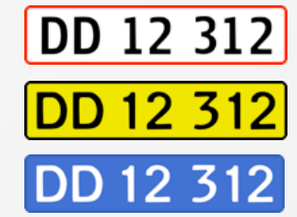
- *Where to store the color of a license plate?*
- *How many license plates on a car?*
- *Relation between phenotype and genotype*

The Entity-Relationship Model—Toward a Unified View of Data

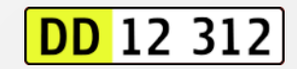
PETER PIN-SHAN CHEN
Massachusetts Institute of Technology

A data model, called the entity-relationship model, is proposed. This model incorp
the important semantic information about the real world. A special diagrammati
introduced as a tool for database design. An example of database design and des

ACM Transactions on Database Systems, Vol. 1, No. 1. 1976.



plus ~40 others and ~200 historical and new law...



an some exceptions...

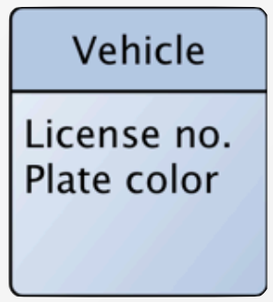
ER diagrams are based on mathematical sets

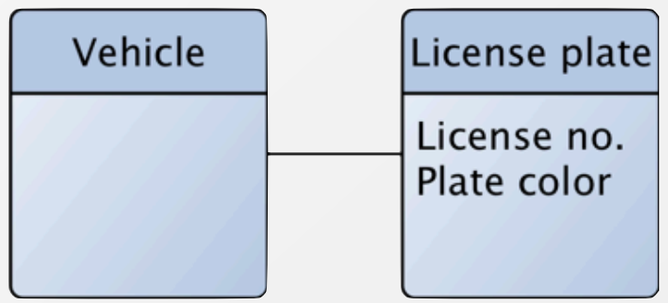
– not everyday language or mental inventory.

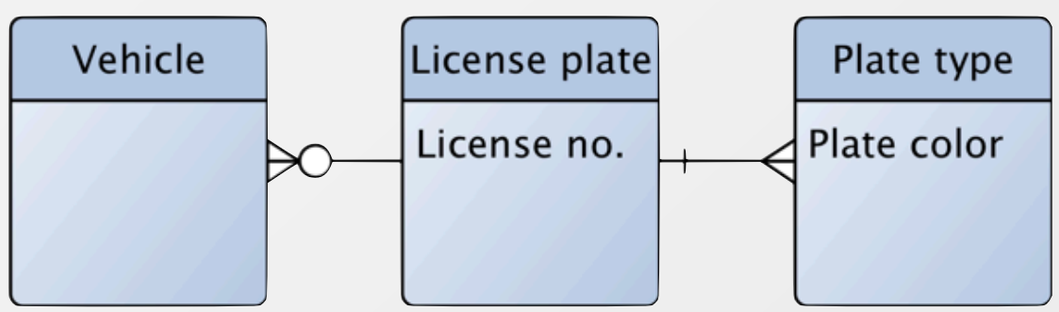
Humans have a mental zoom lens we can switch from parts to the sum of parts with little effort.

Relations also have parts.

Databases often has a fixed focal length and treats some relations as objects and some as properties.







Guy Hamilton. Goldfinger, 1964.

Relational data — revisited

Mathematical set theory is not in our mental inventory and few have solid experience in mathematics.

Relations between categories of objects might be understood by focusing on **prototypic objects** and allow many concurrent '**zoom** levels'. It can be captured in everyday language as **genitives**.

Let the programmers decide on the zoom, and focus on relations as genitives.

A loose dot notation and a controlled vocabulary can be useful.

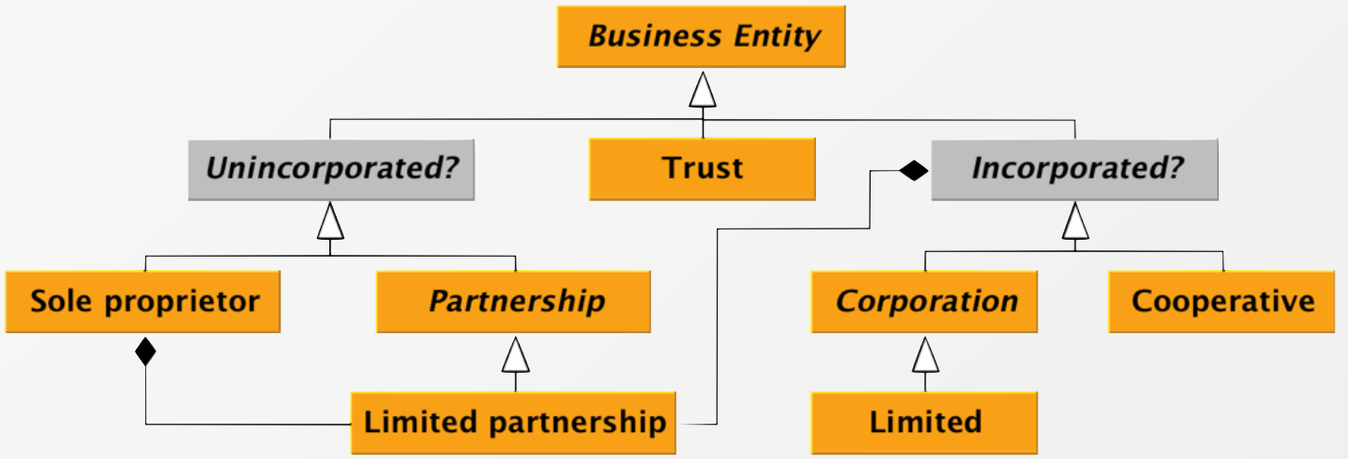
- ~~*Where to store the color of a license plate?*~~
- *Are we talking about the car's color, license plate's color or license plate type's color?*

  "car.plate.type.color"
- ~~*How many license plates on a car?*~~
- *How many fields to write down the license number on the registration form? And how many valid forms for each car?*
- ~~*Relation between phenotype and genotype*~~

  "patient.syndrome.trait.anatomi.expression.gene"

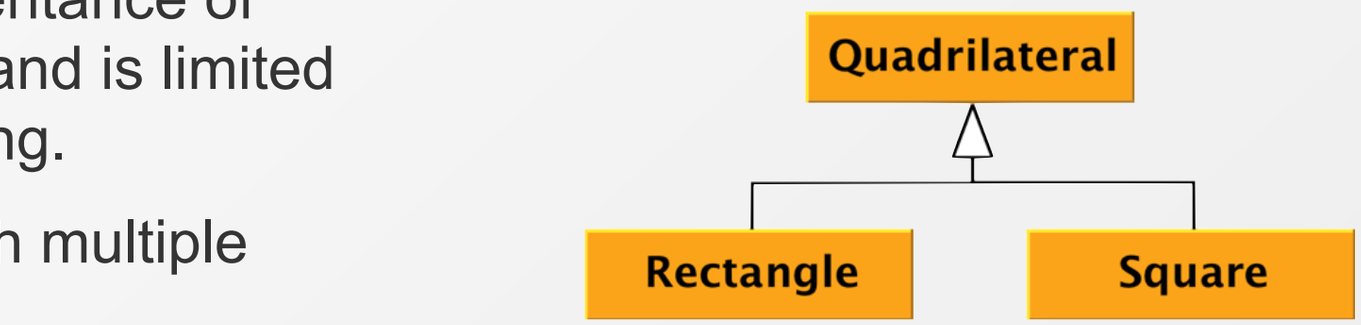
Types and inheritance – examples

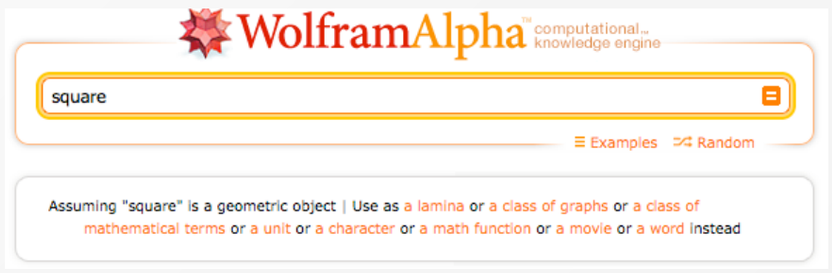
- *How many kinds of business entities?*
- *Is a disease a syndrome or a trait?*
- *A common term for squares and rectangles.*

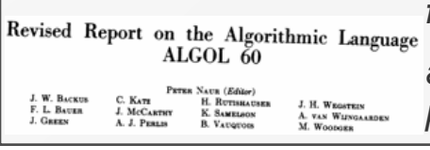


Class diagrams are based on inheritance of types in programming languages and is limited by not allowing ambiguous meaning.

Everyday language copes fine with multiple inheritance by inferring context.

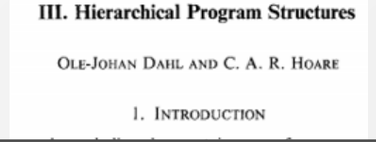





Revised Report on the Algorithmic Language ALGOL 60

formal syntax

activities → classes
processes → objects


III. Hierarchical Program Structures

OLE-JOHAN DAHL AND C. A. R. HOARE

1. INTRODUCTION

abstract vs concrete

a lot of languages

less features
standard API

domain specific languages?

Types and inheritance – revisited

Categorizations *is* in our mental inventory but is not based on classes or schemas.

Humans tend to categorize new objects by focusing on how many **key properties** they share with the **prototype** of an category.

Mental categories are related and one object can belong to **many categories**.

Categories can be **constructed** and **named**, but not all names stick and some categories remains unnamed.

Do not use the general generalization *is-a*, but allow for different hierarchies for every aspect. Prefer ontologies over taxonomies...

Construct new terms only when needed and name them carefully.

- ~~*How many kinds of business entities?*~~
- *What is the common term for businesses that share one important aspect e.g. taxation?*
- ~~*Is a disease a syndrome or a trait?*~~

  (many more aspects needed)
- *A common term for squares and rectangles.*
- *(No need for it in plain english??)*

Giant leaps forward and a small step back...

London 1854

500 deaths

September 8

fewer deaths

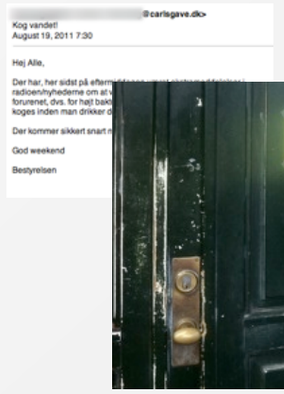
Copenhagen 2011

1 sample

Friday

340 e.coli / L

Saturday



Sunday

10-30 e.coli / L

Monday

0 known cases

What is the adequate information about sources and models to make everyday databased decisions comfortable?