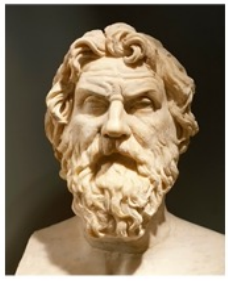
Lecture 0:
Distributed Information Systems
An Overview

During the first couple of lectures, my friends and I all thought the lectures were a bit abstract

- DIS student 2021

The investigation of the meaning of words is the beginning of education

- Antisthenes, 445 – 365 B.C.



Overview

1. What is an Information System?
2. Data Modelling
   - 2.1 Models
   - 2.2 Data and Models
   - 2.3 Representation
3. Managing data and information
   - 3.1 Data Management
   - 3.2 Information Management
   - 3.3 Distributed Information Management
4. About the lecture

1. WHAT IS AN INFORMATION SYSTEM?

Q: What is an information system?

- How would you define what an information systems is?
- Think about what you believe are information systems
- What is not an information system? Are there any computer systems that are not information systems?
- What is the difference between computer science and information systems?

- *Discuss with your neighbor to try to agree on the concept*
- *Write your answers into the Zoom chat*

In this part of the lecture we would like to understand the basic concept of information system. We are surrounded today by information systems and everybody has an intuitive understanding what an information systems is and does (a system that is treating information).

Information Systems – are everywhere

A day in a student's life

- IS academia: course registration, grades, …
- Moodle: course information, slides, …
- Bank account: payments, savings, ….
- Library system: literature
- Search engine: where to find food, …
- Facebook, TikTok: connecting to friends, news, …
- Google maps: finding your way
- Campus map: finding lecture hall
- Email: exchanging messages

A large organization such as EPFL is running dozens if not hundreds of information systems. With the advent of the Web and mobile computing and the resulting democratization of information technology and integration of information technology in every day's life a plethora of new information systems have been emerging. Increasingly, information systems are not only interacting with humans, but also are among each other, with sensors gathering data from the environment, algorithms making decisions and different systems exchanging their data. The recent trend of generating and analysing increasing amounts of data, the so-called Big Data, is even more demonstrating the growing importance of information systems. The following are some types of information systems that are common:

The classical information systems: Organizational databases, Business process management systems, Geographic information systems, Text retrieval systems, Business intelligence (data warehousing and mining systems)

More recent types of information systems: Social Networks, Question-Answering System, Recommender Engines, Bioinformatics systems (e.g., genome or protein sequence retrieval), Environmental monitoring systems (disaster warning, meteo website)

However, not every computing system is considered as an information system. Systems that are used for communication through different media (e.g., IP telephony), games or simulations are not widely considered as information systems. What is the distinctive feature of an information system? In the following we will attempt to provide a more precise characterization of the concept of information system as we understand the concept in the context of this course.

The Business Perspective

Jobs related to information systems

- Project Managers
- Chief Information Officers (CIO)
- Technical Writers
- System Analysts
- Requirements Analyst

Jobs related to computer science

- Computer Programmer
- Java Developer
- Database Administrator
- Software Engineer
- Network Engineer

The notion of information systems is overloaded with many different meanings, depending on the context. From a business perspective, it relates typically to jobs that are concerned with the design of computing systems in an enterprise context. The notion can be distinguished in this context from activities more directly related to software and systems implementation and management.

The Data Perspective

Information systems:

Computer systems handling and interpreting **large amounts of data**

- Transaction data
- Documents
- Maps
- Social networks
- Sensor data

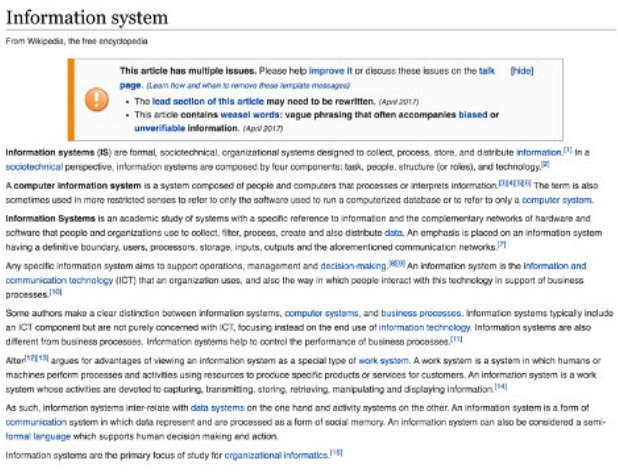
Not information systems:

Computer systems performing **lots of computation**

- Computational science and simulation
- Computer games
- Computer algebra (Matlab etc)

From a more technical perspective we can interpret the notion of information systems as computing systems that process large amounts of data, i.e., the focus is on processing data, instead of performing large numbers of computations.

Ask Wikipedia



Information system

From Wikipedia, the free encyclopedia

This article has multiple issues. Please help improve it or discuss these issues on the talk page. *(Learn how and when to remove these template messages)* [hide]

- The **lead section of this article may need to be rewritten.** *(April 2017)*
- This article **contains weasel words: vague phrasing that often accompanies biased or unverifiable information.** *(April 2017)*

Information systems (IS) are formal, sociotechnical, organizational systems designed to collect, process, store, and distribute information.[1] In a sociotechnical perspective, information systems are composed by four components: task, people, structure (or roles), and technology.[2]

A **computer information system** is a system composed of people and computers that processes or interprets information.[3][4][5][6] The term is also sometimes used in more restricted senses to refer to only the software used to run a computerized database or to refer to only a computer system.

Information Systems is an academic study of systems with a specific reference to information and the complementary networks of hardware and software that people and organizations use to collect, filter, process, create and also distribute data. An emphasis is placed on an information system having a definitive boundary, users, processors, storage, inputs, outputs and the aforementioned communication networks.[7]

Any specific information system aims to support operations, management and decision-making.[8][9] An information system is the information and communication technology (ICT) that an organization uses, and also the way in which people interact with this technology in support of business processes.[10]

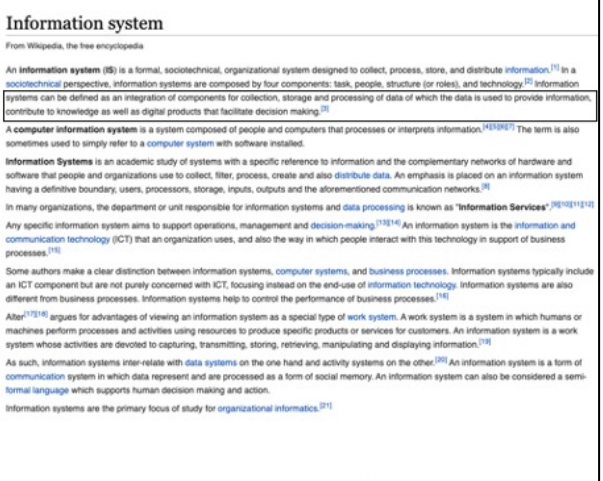
Some authors make a clear distinction between information systems, computer systems, and business processes. Information systems typically include an ICT component but are not purely concerned with ICT, focusing instead on the end use of information technology. Information systems are also different from business processes. Information systems help to control the performance of business processes.[11]

Alter[12][13] argues for advantages of viewing an information system as a special type of work system. A work system is a system in which humans or machines perform processes and activities using resources to produce specific products or services for customers. An information system is a work system whose activities are devoted to capturing, transmitting, storing, retrieving, manipulating and displaying information.[14]

As such, information systems inter-relate with data systems on the one hand and activity systems on the other. An information system is a form of communication system in which data represent and are processed as a form of social memory. An information system can also be considered a semi-formal language which supports human decision making and action.

Information systems are the primary focus of study for organizational informatics.[15]

2018



Information system

From Wikipedia, the free encyclopedia

An **information system (IS)** is a formal, sociotechnical, organizational system designed to collect, process, store, and distribute information.[1] In a sociotechnical perspective, information systems are composed by four components: task, people, structure (or roles), and technology.[2] Information systems can be defined as an integration of components for collection, storage and processing of data of which the data is used to provide information, contribute to knowledge as well as digital products that facilitate decision making.[3]

A **computer information system** is a system composed of people and computers that processes or interprets information.[4][5][6][7] The term is also sometimes used to simply refer to a computer system with software installed.

Information Systems is an academic study of systems with a specific reference to information and the complementary networks of hardware and software that people and organizations use to collect, filter, process, create and also distribute data. An emphasis is placed on an information system having a definitive boundary, users, processors, storage, inputs, outputs and the aforementioned communication networks.[8]

In many organizations, the department or unit responsible for information systems and data processing is known as "**Information Services**".[9][10][11][12]

Any specific information system aims to support operations, management and decision-making.[13][14] An information system is the information and communication technology (ICT) that an organization uses, and also the way in which people interact with this technology in support of business processes.[15]

Some authors make a clear distinction between information systems, computer systems, and business processes. Information systems typically include an ICT component but are not purely concerned with ICT, focusing instead on the end-use of information technology. Information systems are also different from business processes. Information systems help to control the performance of business processes.[16]

Alter[17][18] argues for advantages of viewing an information system as a special type of work system. A work system is a system in which humans or machines perform processes and activities using resources to produce specific products or services for customers. An information system is a work system whose activities are devoted to capturing, transmitting, storing, retrieving, manipulating and displaying information.[19]

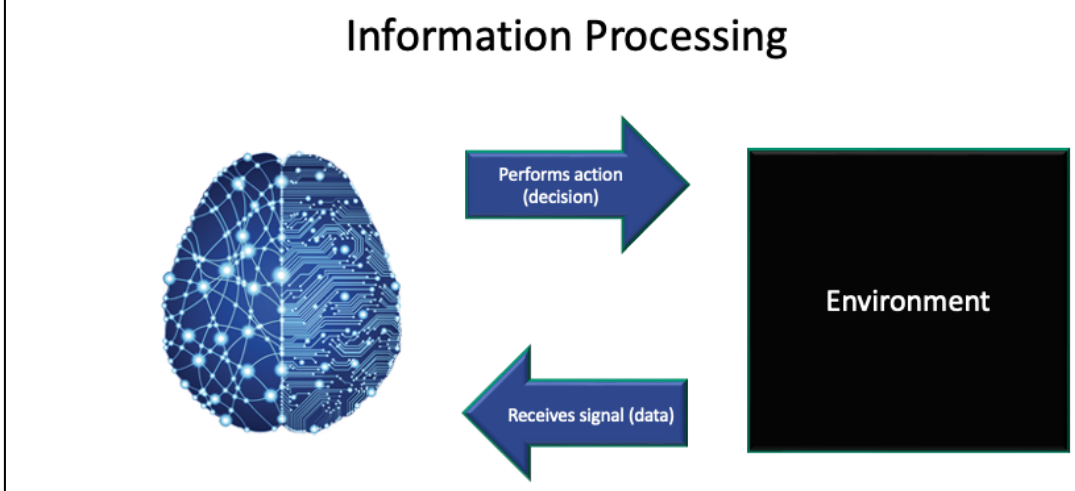
As such, information systems inter-relate with data systems on the one hand and activity systems on the other.[20] An information system is a form of communication system in which data represent and are processed as a form of social memory. An information system can also be considered a semi-formal language which supports human decision making and action.

Information systems are the primary focus of study for organizational informatics.[21]

2021

When inspecting the article in Wikipedia we can notice that the “business perspective” seems to be predominant.. However, it is interesting to note that recently the processing of data has been explicitly added in the characterization of the concept.

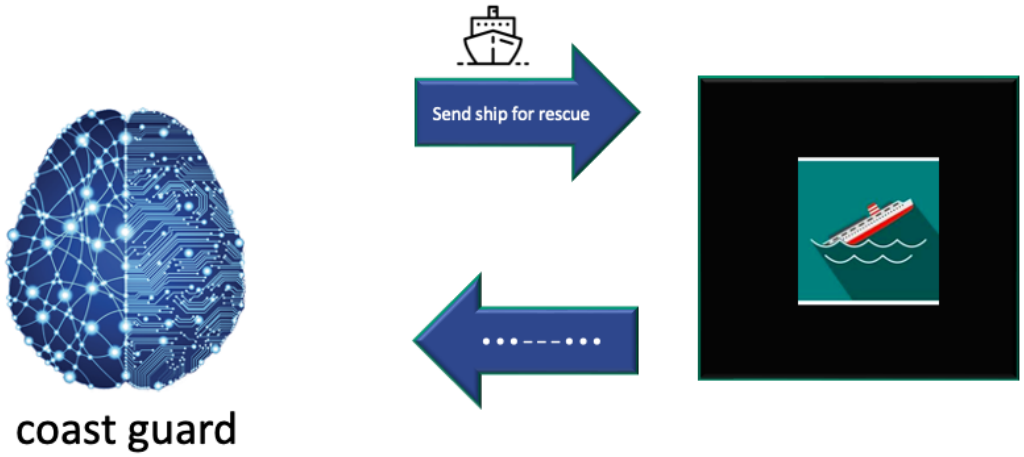
A Systems Perspective



This model applies to all types of systems: organisms, species, humans, organizations, societies, etc.

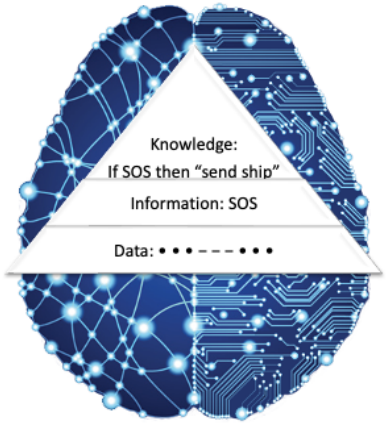
A systems perspective would rather focus on the question of how an information processing system is interacting with its environment. This is the perspective we will adopt in the following.

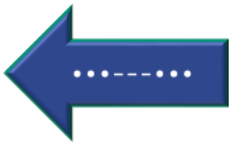
Information Processing: example



We can describe this process by a simple example. Let us assume that a ship is in distress and sends the (famous) SOS signal to the coast guard.

Data-Information-Knowledge





Data = signal received
Information = understanding of signal
Knowledge = decision taking

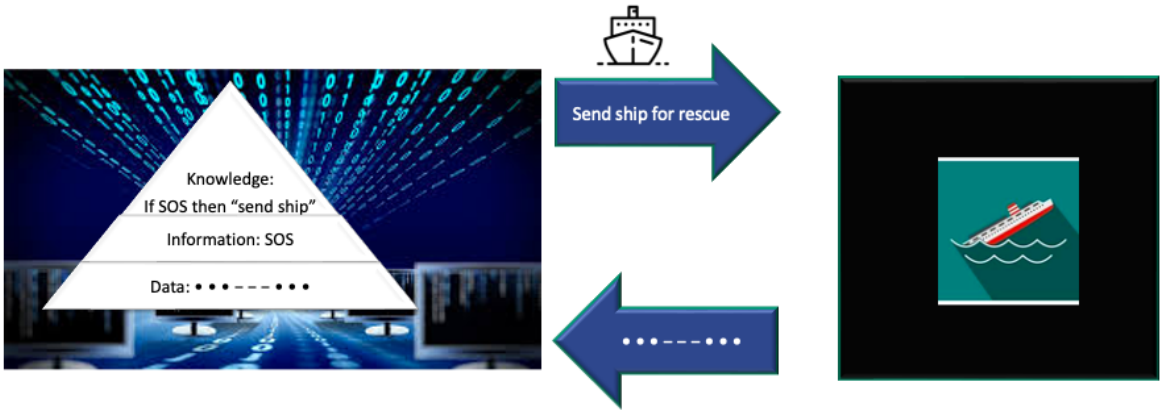
Looking in more detail on what is happening, we can identify different aspects of this process.

First, we observe that the ship is sending a signal. This corresponds to the transmission of data, irrespective of how the data is transmitted over communication channel, e.g., by radio, Internet, Morse, fume signals etc.

The receiver of the data (the Morse code) has then the task of interpreting the data (or signal). Only this establishes what we would consider information. So, information is related to the understanding of data or giving meaning to the data. Without being able to perform this interpretation of the data at the receiver, sending the signal would be useless.

Finally, understanding the message does not mean that the receiver also knows what to do. The receiver also has some knowledge, i.e., that capability of reacting in a meaningful way with the environment based on the information received.

Information Systems



An information system automates the process of receiving, interpreting and acting upon signals
A lot of signals = Big Data

The process described in the example, applies to any entity that is performing information processing in the context of an environment, e.g., an organism, an organization, or a computing system. Performing information processing in the context of computational systems is what we will be studying in this lecture.

What is an Information System?

An information system is a **computing system** interacting with the environment with the capability of

1. managing **data,**
2. inferring **information** by understanding this data
3. for performing a given **task**

Based on the previous considerations, we can come up with a first attempt to define what an information system is. Note that the notion of environment in the definition can be interpreted widely. It can consist of humans interacting with the information systems, as well as the physical environment or other information systems.

Compare to the definition provided by WikiPedia, the definition looks quite similar:

In WikiPedia it is written that tnformation systems can be defined as an

- integration of components for collection, storage and processing of data (-> managing data)
- of which the data is used to provide information (-> inferring information),
- contribute to knowledge that facilitate decision making (-> performing a task).

Types and Sources of Data

IS academia: course registration, grades, …
Moodle: course information, slides, …
Bank account: payments, savings, ….
Library system: literature
Search engine: where to find food, …
Facebook, TikTok: connecting to friends, news, …
Google maps: finding your way
Campus map: finding lecture hall

Text
Structured database
Knowledge bases
Social networks
Sensors
Images and videos

In our definition we have three main constituents: data, information, tasks

Data is relatively straightforward to understand. We have today a wide array of different data that is generated and can be handled in information systems. Looking at the different types of information systems we use daily, we can easily derive the different types of data that occur.

Tasks

IS academia: course registration, grades, …
Moodle: course information, slides, …
Bank account: payments, savings, ….
Library system: literature
Search engine: where to find food, …
Facebook, TikTok: connecting to friends, news, …
Google maps: finding your way
Campus map: finding lecture hall

Searching information
Event notification
Prediction
Classification
Recommendation
Summarization
Question answering
…

Similarly, considering the different information systems we use daily, we can also identify a large number of different tasks that we perform using them, or tasks in which information systems support us. When studying and developing information systems it is crucial to have a clear understanding of what is the task at hand.

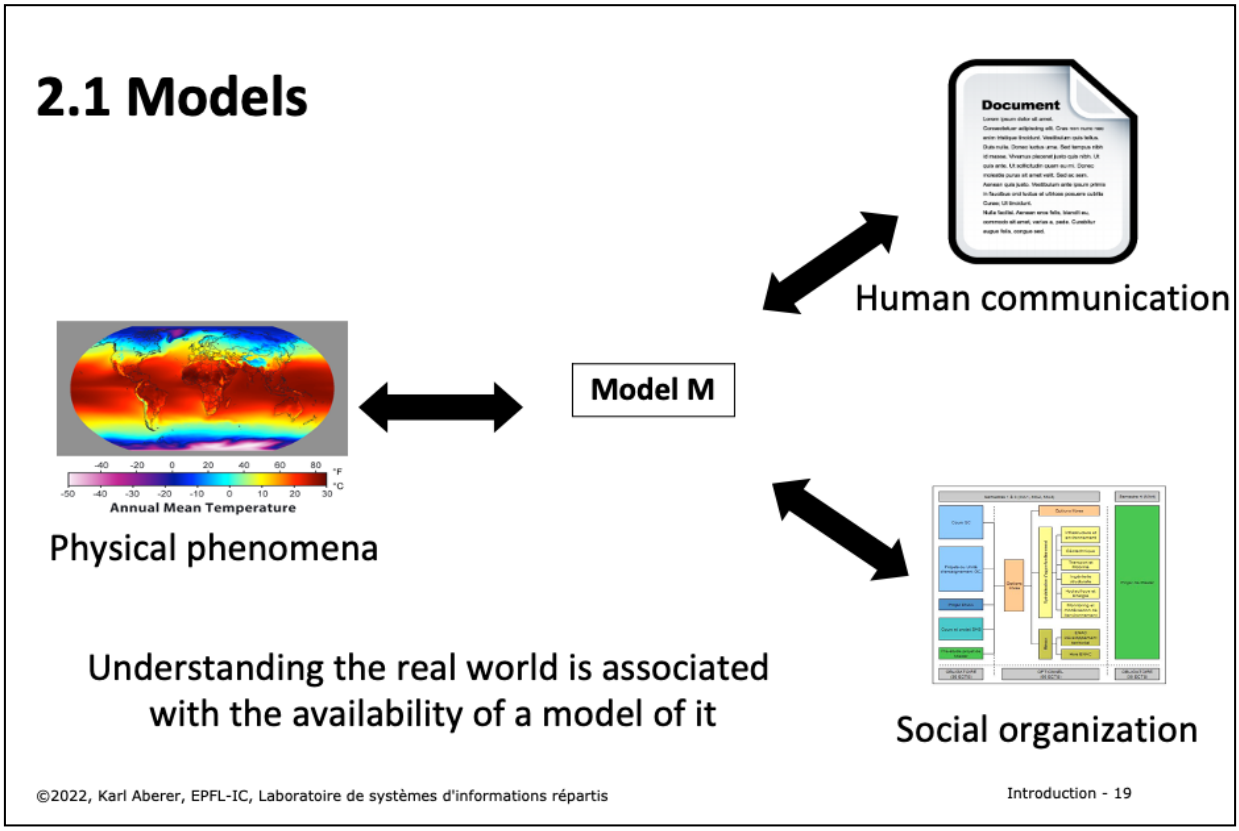
Q: What does “understanding” mean?

- What does it mean to understand/interpret data?
- What exactly is information?
- How is information different from data?
- How can we techically characterize the difference between information and data?

- *Discuss with your neighbor to try to agree on some answer*
- *Write your answers into the Zoom chat*

We have an intuitive understanding of what ”understanding” means. But what does it mean from a technical or scientific perspective?

2. DATA MODELING



We can identify broadly three types of environments information systems would typically interact with. For interacting with its environment the information systems needs a model of it, i.e., a model of some aspect of the real world.

1. Physical phenomena: these information systems manage environmental data and create models of physical phenomena. Typical examples are meteorological information systems, or geo-information systems.
2. Social organization: these information systems capture the roles, relationships, activities etc. in social organizations, such as businesses and institutions. This type of information systems is among the earliest one that had wide adoption, for enterprise applications such as finance, logistics etc.
3. Human communication: these information systems capture of how humans communicate and reason. They enable to capture the meaning of text and other media, but also model human traits such as sentiments or opinions. Information retrieval systems, of which web search engines are a specific example, are a typical representative of this class of systems.

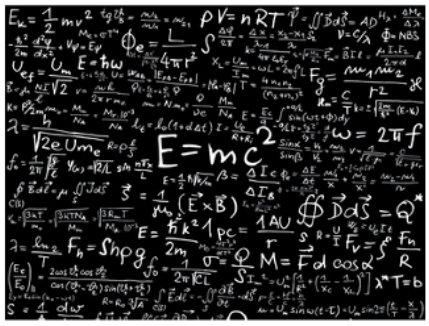
Models = Mathematical Models

A **mathematical model** is a representation of a system using mathematical concepts and language

A **mathematical model** consists of a set of

- **Elements** (or constants/identifiers)
- **Functions** (or relations)
- **Axioms** (or constraints)

The set of elements and functions must be consistent with the axioms



Information systems use models to represent some aspect of their real-world environment. But what is a model? The answer is simple: a model is a (mathematical) structure.

A mathematical structure consists of elements from a domain (also called constants in a mathematical language), functions and axioms. The elements provide identifiers for things (indeed everyTHING can receive a name – and a central task of information systems is to handle names, respectively identifiers, of real-world entities), the functions provide properties of those entities, and the axioms provide rules or constraints that state which properties are possible and not.

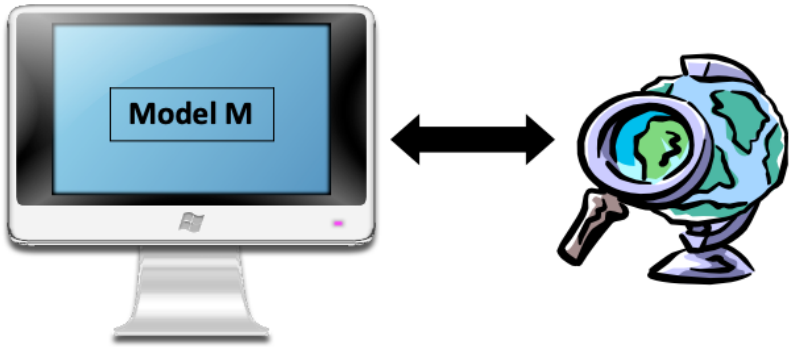
Some information systems support only very limited or specific kinds of models, others very generic models. Very often first-order logic is used as generic modelling approach for information systems. Models based on first order logic allow to represent the important entities, their relationships and properties in a generic approach. However, with the increasing need to process information for specific needs (e.g., processing text, images, sensor data), also more specific mathematical models are increasingly used, such as graphs, vector spaces, probabilistic models, differential equations, process models etc.

Here are some examples of formal models used in information systems:

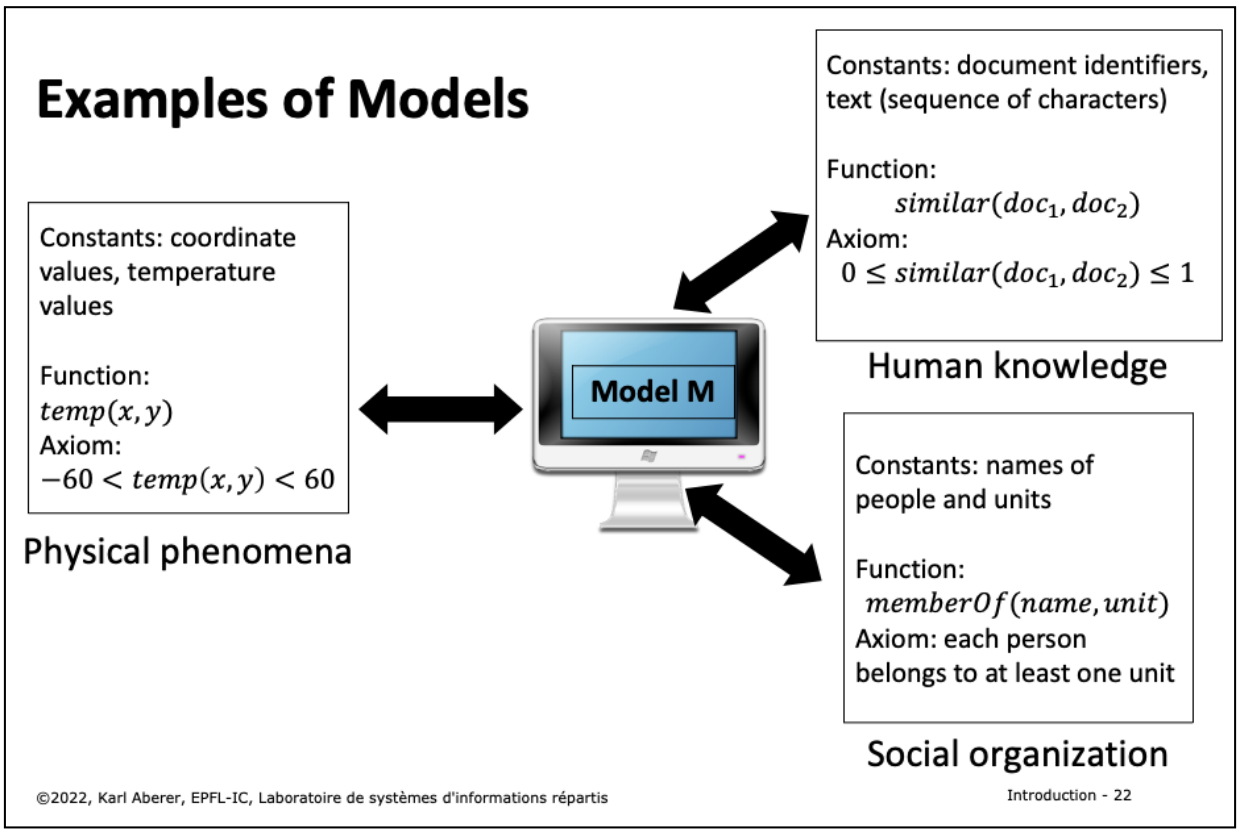
- Entity-Relationship models have been among the earliest conceptual models used for information systems. They have been derived from knowledge representation mechanisms developed in AI.
- OWL: is a generalization of the entity relationship model enabling logical inference (for concept classes). It has become the basic model for the Semantic Web.
- Graph models: used for social network data, biological network data, communications network data
- Vector space models: used to represent feature spaces of text and media content
- Probabilistic models: used to represent uncertainty in content and sensor data
- Differential equations and simulation programs: used to represent behaviors of complex systems
- Process models: capture the structure and dynamics of business processes, also called workflows.

What is an information system?

An information system is a **computing system** that manages a representation of a **model** of its **real-world environment** within a computer system for a given **purpose**



Based on the previous consideration we can come up with a first definition for information systems. We consider information systems that are built on information technology, i.e., that are computing and communication systems. In addition, one can quite safely state that information systems need to have available a model of the real-world environment, otherwise that would not know what they do to achieve their purpose.

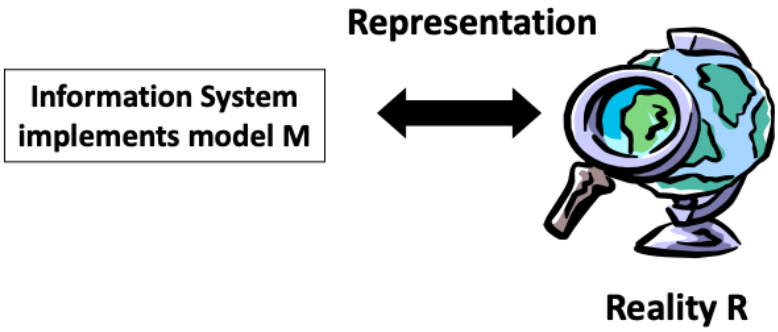


We have stated that models are mathematical models. For illustration, we can show possible models for the three examples that we have provided for real-world environments in which information systems operate:

A temperature map we can model, for example, as a two-dimensional matrix. An organizational structure is best captured using relationships among entities, whereas the meaning of documents can be captured by their degree of similarity.

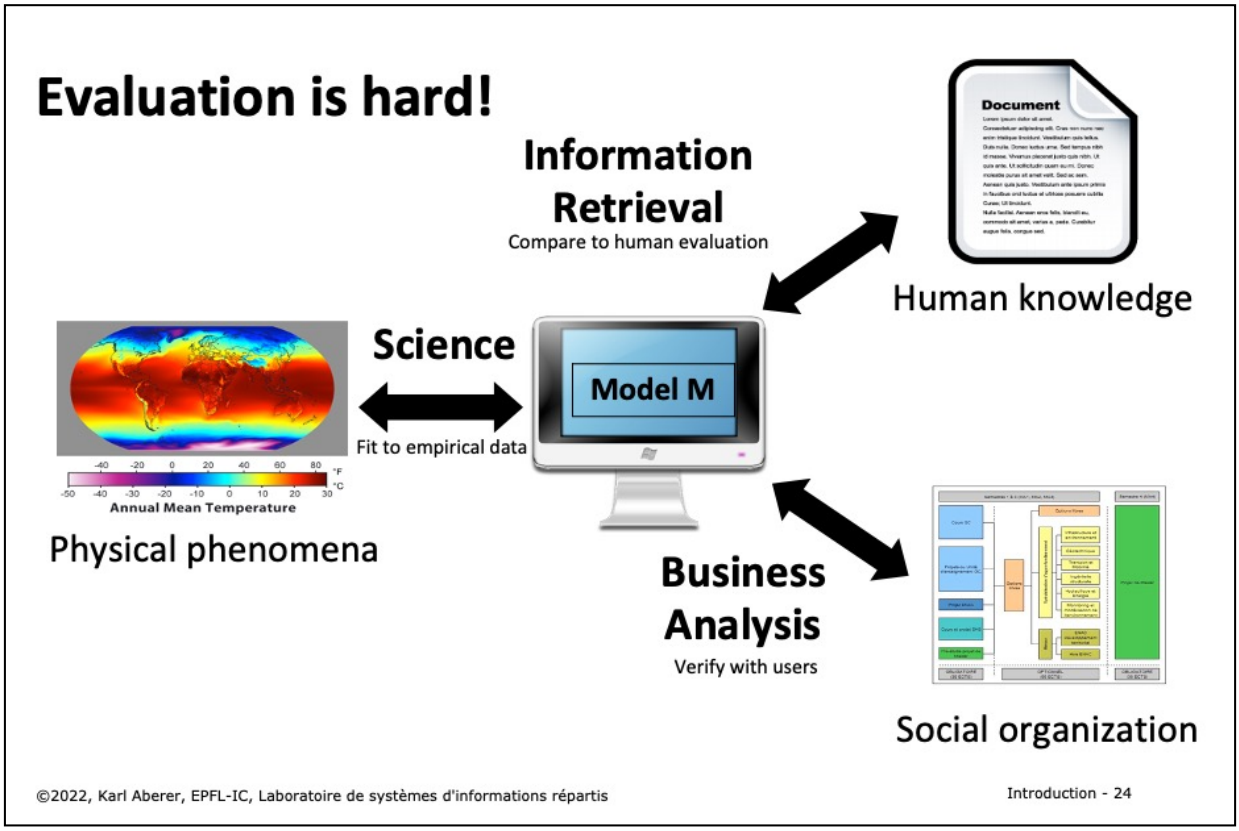
How do we know that a model is good?

The model should represent some aspect of the real world



We have to **evaluate** whether the model represents reality properly

When building a model of the real world environment of an information system, a natural question to ask is of how we can know that the model is indeed a good model of the real world.



Since there is no way to formally define or validate the relationship between a mathematical structure and the real world, methods are required that help to make plausible, that a model represents the real world correctly in some sense. With respect to physical phenomena it is indeed Science that endeavours to create models of reality (e.g. physical laws) using the scientific method. With respect to capturing human thought, expressed for example as written text, the field of information retrieval has established methods to verify whether computer models appropriately represent or emulate, the human reasoning processes. For traditional business information systems, it is the profession of business analysts that translate real world requirements and organizational structures into models for information systems.

These approaches always implement some form of feedback mechanism, in which the models that are developed are verified with respect to reality. E.g., business analysts present the models to potential users and refine them based on the feedback. Scientists verify their models with respect to experimental data and assume they are valid as long they are not falsified (as we know since Popper we can never proof that a model is correct!).

Q: What is data?

- What is the connection between data and models?
- What actually is data?
- How information systems deal with data to support models?

- *Discuss with your neighbor to try to agree on some answer*
- *Write your answers into the Zoom chat*

2.2 Data and Models

Example: a Mathematical Model for trajectories:

- Domains: time T, space $S = Long \times Lat$
- trajectories Tr is a set of functions
- one trajectory $tr \in Tr$ is a partial function $tr: T \rightarrow S$

a sequence of GPS point (x,y,t)

In order to illustrate the use of models in an information system, we will use a simple, yet practical, example, the management of trajectories. Here we describe a basic trajectory model in a 2-dimensional space, as it is frequently used to describe, for example, vehicle trajectories recorded using a GPS.

(this example will also be used in the exercises)

Representation of Functions

Functions can be represented

1. by a specification or algorithm or
2. by enumerating values it can take
   - The enumerated values are called **data**

Representing a single trajectory tr

- as algorithm: $tr(t) = s_0 + \frac{(s_1 - s_0)}{(t_1 - t_0)}(t - t_0)$
- as data, a set of samples: $tr(t_0) = s_0, tr(t_1) = s_1, \ldots$

Functions are a key constituent of information systems. They relate objects with their properties and different objects among each other. An interesting and central question is of how functions are represented.

There exists two fundamentally different ways to do this: either by explicitly enumerating function values for given function arguments, or by providing a specification or algorithm to compute the function value from a given function argument. Both ways are in fact used in information systems.

In our trajectory example, we can use both methods. We could describe trajectory by (measured) samples, or by a function generating a trajectory.

In the context of science, functions are often called quantitative or qualitative variables.

Functions in Information Systems

Functions can be implemented as algorithms

- functions, queries, views etc.

Information systems often rely on representation of functions as data

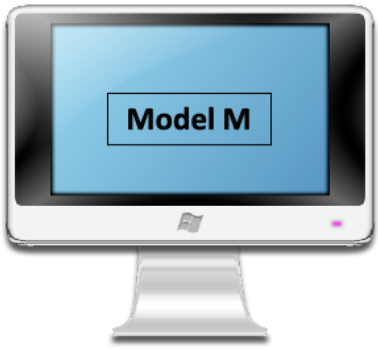
- many aspects of the world are not algorithmically defined, e.g., birthdate of a person or outcome of an experiment, but are rather observations

However, since many facts about the real-world cannot by specified by an algorithm (e.g. computing the birthday from the name of a person), the explicit representation of functions plays a particularly central role in information systems. Explicitly enumerated functions are commonly called **data**.

Nevertheless also implicitly represented functions play an important role in information systems (computing the age of a person from its birthday). Such functions appear under many different names, such as queries, views, user-defined functions, stored procedures etc.

Data Structures

How is a model represented in a computer system?



A mathematical model M is represented using a **data structure D**.

A data structure D is a discrete mathematical structure and their operations that can be **processed by a computer**

So far we have considered models as abstract mathematical structures, consisting of domains, functions and axioms. For being able to handle a model in a computer system, we need to represent the model within the computer system. For that purpose we need a representation mechanism that can be "understood" and processed by a computer. In other words we need a mechanism that can be used to represent a model. Such a mechanism is called a **data model**. A data model consists of discrete mathematical structures and operations that a computer can represent and execute.

Abstract Data Types (ADT)

Are mathematical models of a data structures

Example: associative array `A`

- $A(K,V) = \{(k,v)|k \in K, v \in V\}$
- K, V are other ADTs
- Operations:
  `A.put(key, value), A.get(key), A.delete(key)`
- Constraint: every key occurs only once

Can represent a function $f: K \rightarrow V$

Abstract data types represent abstractions of data structures and the signatures of the functions for manipulating them.

For example, the **associative array** is an abstract data structure (abstract data type) that can be used to represent functions. Associative array is a data structure that manages a set of key-value pairs and that supports a set of operations to manipulate the associative array, such as adding, deleting and modifying the elements of the associative array. The associative array can represent a function by encoding a finite set of function values for given arguments.

It is called abstract data type, since different (physical) implementations of the same data structure are possible.

Example

Mapping the domains to the data structures

$R: T \rightarrow Float$

time in seconds from some reference time

$R: S \rightarrow Tuple(Float, Float)$

longitude, latitude in degrees

Mapping the functions to the data structures

$R: Tr \rightarrow A(T, S)$

$R(tr) = \{(R(t_0), R(s_0)), (R(t_1), R(s_1)), \ldots, (R(t_n), R(s_n))\}$

Representing the trajectory model in Python

```
set(dict(float, tuple(float, float)))
```


The example shows one possible way of representing our mathematical model of trajectories using concrete data structures. Note that we specify not only the mapping R, relating the mathematical structure to the data structures, but as well the units used, for a good reason.

Is this mapping trivial?

Not exactly. If the meaning is not precisely understood bad things may happen.

When NASA Lost a Spacecraft Due to a Metric Math Mistake

Not considering properly units, has led already to some major catastrophies.

Object-Oriented Models

Allow to encapsulate ADTs and mimic the (signature of the) mathematical structure

```
class trajectory:
    def __init__(self):
        self.p = {}
    def add_point(self, t, c):
        self.p[t] = c
        return
    def location(self, t):
        if t in self.p:
            return self.p[t]
        else:
            return None
```

```
tr = trajectory()
tr.add_point(1, [0,0])
tr.location(1)
```

```
[0, 0]
```


Object-oriented models are a more elaborate way to represent mathematical structures in a programming context. They allow to group all the signatures of functions in a common context.

Data Models

A **data model** is a language (or framework) used to specify data structures. It consists of

1. Data Definition Language (DDL)
2. Data Manipulation Language (DML)

The specification of a data model using a DDL is called a (database) **schema S**.

A data model is a framework and language to define data structures.

Data modelling languages consist of two parts: A data definition language DDL enables the specification of data models, consisting of possible data structures and integrity constraints. The data manipulation language allow to specify the functions in the data model.

A specification of a data model using a DDL is called a database scheme, respectively simply schema.

Attention

The term data model is used in different ways!

- Sometimes it refers to frameworks for specifying conceptual, logical and physical models
- Sometimes a specific schema is called data model

The use of the term data model in computer science is not always consistent. Sometimes a specific schema expressed in a data modelling language is called a data model. We will prefer the use of data model for frameworks for specifying data structures.

Physical Implementation of Data Structures

Requires a binary representation of the data structure

- Different implementations have different performance characteristics

Example: Map associative array to an array structure

k1	v1	k2	v2 ….
000	010	110	001 …

implement the functions of the ADT

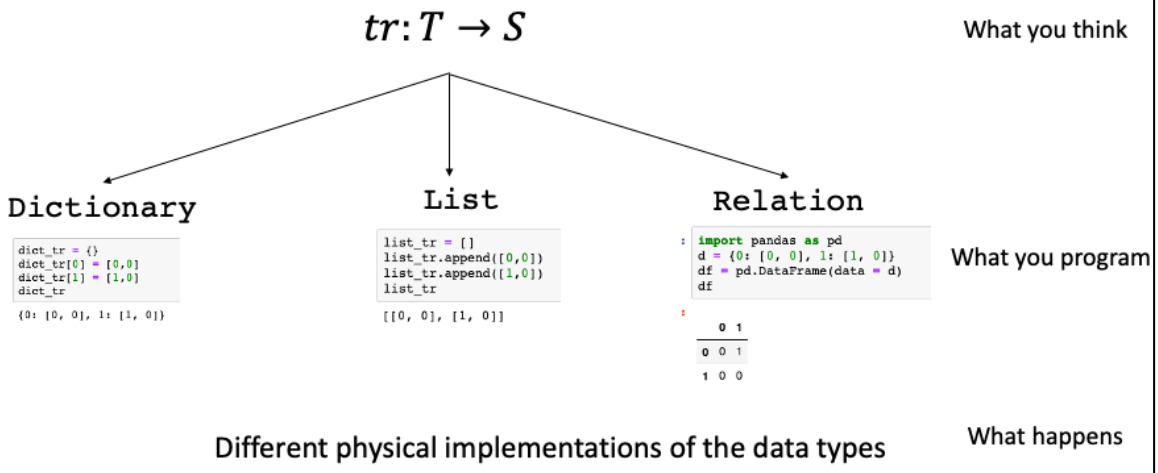
Alternative implementations of associative arrays

- hash tables, binary search trees, tries, …

The data structures, as mathematically defined by abstract data types or object-oriented models, need to be represented in a computer system usign the primitives that are available in the computing system. For data, this primitive is usually a binary representation. Therefore, the data structure is mapped into a binary presentation, and the operations on the data structures are implemented using those representations. This is then the physical implementation of a data structure.

Relationship among Model and Data Model

The same function can be represented using different data structures



Data structures allow to represent the functions that are part of a model within the computer systems. The same function can be represented in different ways using different abstract data types and data structures. Some representations are more appropriate than others. E.g., using a hashtable enforces the constraint that a function can have only one function value, whereas using a list of tuples requires to enforce this constraint and typical operations in the code.

Three levels of modeling

Conceptual modelling

- mathematical model that user thinks in

Logical modelling

- mathematical model that computer can process

Physical modelling

- binary representation of logical model

To summarize, we can identify three levels of modeling, when designing an information system: the conceptual, the logical and the physical level.

These three levels have been identified as a formal standard in the ANSI-SPARC architecture for abstract design standard for database systems in 1975.

Three Notions of Data

Conceptual level (data as in **science**)

- Values a variable (function) can take

Logical level (structured data as in **programming**)

- Value (instance) of an abstract data type

Physical level (data as in **information theory**)

- Binary representation

We can also relate the concept of “data” to these three modeling levels. This reveals the interesting fact that we are continuously using the term data with varying meaning (without necessarily being aware of that).

The first meaning corresponds to the notion of data as used in a scientific experiment, where data are values a function can take. These are also often called samples. This is also the meaning that is associated with data in the corresponding Wikipedia article.

A second meaning corresponds to the notion of data as being an instance of an abstract datatype processed in a computing system. This corresponds to the notion of data as it is being used, for example, in the context of database management systems.

Finally, when considering the physical representation of data as binary strings, data corresponds to the concept of data as being used in the context of information and coding theory (e.g, when investigating data compression)

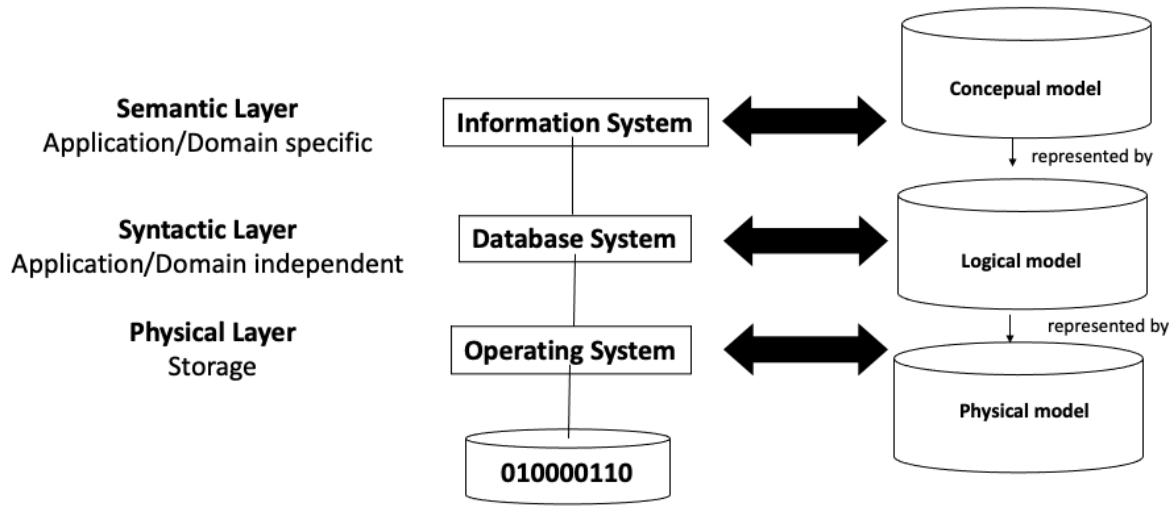
Representation of Conceptual Models

Conceptual models of the real world are **represented** in information systems as data structures

- The elements are represented as **structured data**
- The functions are represented as **programs** operating on this data or represented as **structured data**
- The structured data is represented in **binary format**

To summarize, we understand now in more detail of how mathematical models are represented in information systems, via structured data and its binary representation. Information systems from a systemic viewpoint, as described earlier, deal ultimately with all three aspects of modeling, conceptual, logical and physical.

Refined View of an Information System



Factoring in the design principles of data management systems we can provide now a refined view of a typical information system architecture. The semantic layer is concerned with the modeling of the real world, the conceptual modeloing. In order to implement the model the syntactic layer provides a data model in which the semantic model can be implement and that is supported by a generic software (typically a database management system) that supports a wide range of information systems. The physical layer deals with the problem of representing the constructs of the data model efficiently in the available computing infrastructure, using the typical mechanisms as provided by its operating system, such as access to storage and network resources.

Q: What does “representation” mean?

- We have been using repeatedly the notion of representation, but what (mathematical) notion does representation refer to?
- How do we know that a representation is a representation?
- Are there different kinds of representation?

- *Discuss with your neighbor to try to agree on some answer*
- *Write your answers into the Zoom chat*

2.3 Representation

A **representation** is a very general relationship that expresses similarities or equivalences between mathematical structures.

- Maps the domain of one structure into the other
- Maps functions and relations of one structure into the other
- Preserves some properties

As we have seen, the term representation is widely used in the context of information systems, and more generally in mathematics. Though we have an intuitive understanding of the concept, it is worthwhile to investigate in some more detail, what exactly is meant by this concept. We give here a generally accepted definition of the concept representation in mathematics. It is a structure-preserving mapping, which means that it maps the domains as well as the structural elements, i.e. functions and properties. What the concept of preservation of properties means, requires some further explanation.

Example: Homomorphism

Let X and Y be two mathematical structures with the same functions

A **homomorphism** $H: X \rightarrow Y$ has to satisfy for every function f:

$$H\big(f(e_1, e_2, \dots)\big) = f(H(e_1), H(e_2), \dots)$$

This is one possible preserved property!

If H is in addition bijective, then H is an **isomorphism**

One type of representation that preserves properties in a mapping between two mathematical structures, are homomorphisms, a widely used concept in mathematics. Informally, homomorphisms preserve structural properties fully. If the homomorphisms are in addition bijective, we have isomorphism. We can think of isomorphism as a mapping between two mathematical structures that are equivalent.

Example: Representing a Discrete Model

Mathematical model:

graph $G \subseteq D \times D$

neighbor function: $n: D \rightarrow \{D\}$

Representation as abstract data type

$R: D \rightarrow Integer$

$R(G)$ of type `list(tuple(int,int))`

```
def neighbor(i, G):
    … code to retrieve all neighbors …
```

R is bijective, isomorphic!

In many cases where we represent discrete structures in computer and information systems the representations are realized by isomorphisms. That means that the structures are exactly represented, without loss of information.

Binary Representation

Data structure: `list(int)`

Example: `1,8,3, …`

Binary representation:

$R(n) = b_2 b_1 b_0$ if $n = b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \cdot 2^0$

$R(n_1, n_2, \ldots) = R(n_1) R(n_2) \ldots$

Example: `001 100 010 …`

R is bijective, isomorphic!

Also, representations of data structures in binary form, are examples of isomorphic mappings.

Data Exchange Format

Data structure: `list(int)`

Example: `1,8,3, …`

JSON representation:

‘[1, 8, 3,…]’

R is bijective, isomorphic!!

Similarly, also data exchange formats are examples of exact representations (fortunately so, otherwise we would lose data when exchanging it among computer systems).

Example: Representing a Continuous Model

Mathematical model: domains $T, S = Long \times Lat$

Trajectory: $tr: T \rightarrow S$

Representation of time domain in Python:

$R: T \rightarrow Float$

Almost a homomorphism!

```
def timesteps(s, u, n):
    print(s)
    if n == 0:
        return s
    else:
        return timesteps(s + u, u, n-1)
```

```
timesteps(0, 0.2, 10)
```

```
0
0.2
0.4
0.6000000000000001
0.8
1.0
1.2
1.4
1.5999999999999999
1.7999999999999998
1.9999999999999998
```


However, for non-discrete mathematical structures the situation is different, since we cannot map a continuous domain into a discrete domain without loss of information.

One typical illustration for the consequences of this, are anomalies that we encounter in numerical computation.

Representations are not always accurate

The representations need not always be homomorphisms

- They preserve some relations approximately
- Measures for the quality of approximation are needed

Therefore, representations do not necessarily always perfectly preserve the original structure. This might be due to the impossibility of doing so (as in the case of mapping continuous domains into discrete domains), or on purpose. Often representations are used to simplify or approximate the original structures, to make them more amenable for processing.

One important case of such mappings used to approximate the original structure is found in so-called “representation learning”. There, complex discrete structures, e.g., text, are mapped into vector spaces in order to capture essential properties of the original data. We will study this approach in detail in this lecture.

Example

Representing an empirical mathematical model, by another mathematical model

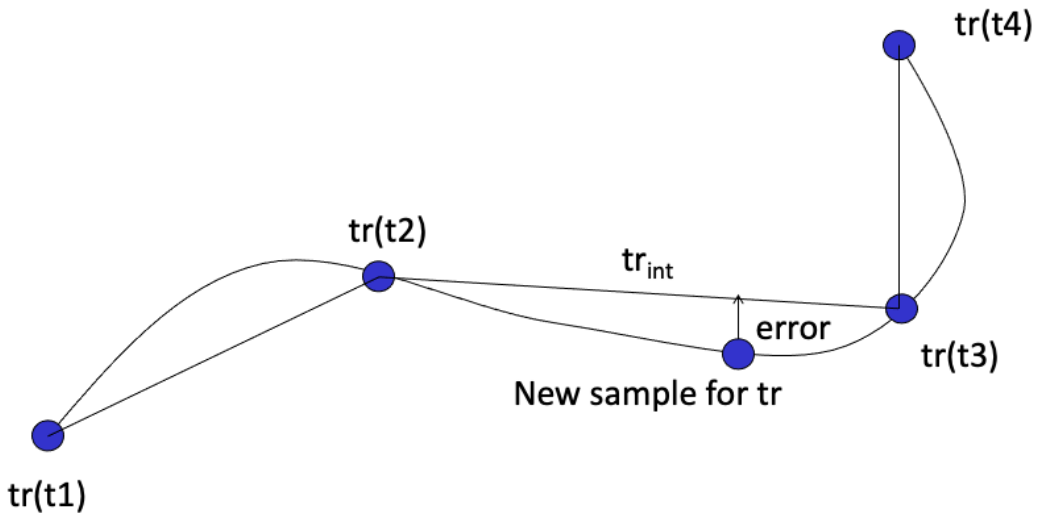
- Assume we have only partial data for a trajectory tr (samples)
- We interpolate the missing values, e.g., linear interpolation

$$tr_{int}: T \rightarrow S$$

- If new data arrives, we can estimate the error
- The error measure characterizes the quality of the new representation tr_{int}

One possible reason for an approximative representation is that from the original structure data is missing. In this case the representation can use, for example, interpolation to replace the missing data values.

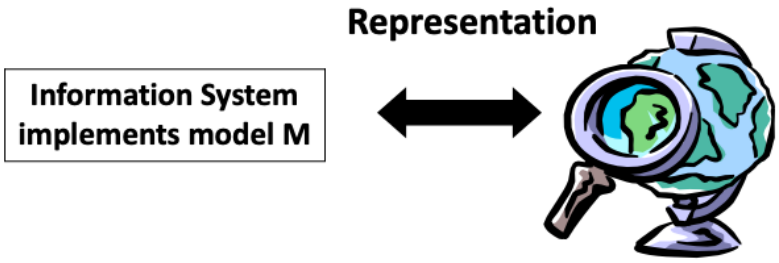
Illustration



This examples shows a simple interpolation mechanism for trajectory data.

Representing the Real World

A model represents a reality, thus there exists a (hypothetical) **representation relationship**



Evaluating the model to characterize the quality of representation

As we stated earlier, mathematical are used to represent some aspect of the real world. Therefore, we can also consider this as a form of representation. Obviously, this relationship can in general not be formally described.

3. MANAGING DATA AND INFORMATION

Q: What is needed to manage data?

- Identify problems and techniques that are used to manage data
- What kind of systems are being used to perform those tasks?

- *Discuss with your neighbor to try to agree on some answer*
- *Write your answers into the Zoom chat*

3.1 Data Management

The collection of data represented in a data model D is called a **database DB.**

A computer system that is designed to (generically) manage databases is called a **database management system DBMS**.

We already mentioned that the explicit representation of functions by enumeration, resulting in data, plays an important role in information systems. In the context of data management such a set of data is called a **database**. A computer system that is designed to manage databases is called a **database management system DBMS**. Thus, database management systems can be used to manage databases, and thus to implement information systems. Many information systems are implemented without relying on the use of a (generic) database management systems, but us a proprietary implementation of the the corresponding data management capabilities.

Data Management

Efficient management of large amounts of data

- efficient storage and indexing
- efficient search and aggregation

Ensuring **persistence** and **consistency** of data under updates and failures

- Persistence = data stored independent of lifetime of programs
- Consistency = data correct independent of type of failures

When handling large amounts of data, we face two key challenges: first, the management should be performed efficiently. This concerns questions of how the data is mapped to the different available storage media, and what auxiliary data structures, so-called indices, are created to support the efficient access and questions of how operations on the data, e.g., for search and aggregation, are performed algorithmically in an efficient way. Second, the data needs to be kept safely. Different to data that is managed within the lifetime of a program, so-called transient data, data in information systems is maintained independently of the lifetime of any program. This property is called **persistence** of the data. Ensuring persistence and consistency of data, under all possible failures and when the data is stored in a variety of storage media us a non-trivial problem. The main purpose of data management systems is to provide performant implementations to support these tasks.

Examples of DBMS

Programming environment

- e.g. Python

Relational data management

- e.g. SQL, Pandas

Distributed data processing

- e.g. Map-Reduce, Spark

Text processing

- e.g. Elasticsearch, Lucene

Whereas traditionally DBMS meant to designate a quite narrow class of systems, like relational database management systems, with the multiplication of information systems and associated infrastructures we may observe a multiplication of platforms that can be considered as database management systems.

Example: Relational Schema

Data Definition Language

```
CREATE TABLE Trajectory
    (tid integer, date datetime, x float, y float,
    primary key(tid, date))
```

Data Manipulation Language

```
SELECT date,x,y FROM Trajectory WHERE tid = 123
```


The probably best known data modelling formalism is the relational data model (note the use of the term data model!). In this small example we see of how to use the DDL to define a table with a specific structure (e.g. three fields) and how to define an integrity constraint for this data structure, i.e. that the Studentid field needs to be unique. Using the DML a query is formulated. A query is nothing else than a function that is computed on the data structure.

Example: DataFrames

	tid	date	x	y
0	1	2008-02-02 15:36:08	116.51172	39.92123
1	1	2008-02-02 15:46:08	116.51135	39.93883
2	1	2008-02-02 15:46:08	116.51135	39.93883
3	1	2008-02-02 15:56:08	116.51627	39.91034
4	1	2008-02-02 16:06:08	116.47186	39.91248
...	...	...	...	...
583	1	2008-02-08 15:11:31	116.48347	39.91954
584	1	2008-02-08 15:21:31	116.50789	39.93128
585	1	2008-02-08 15:31:31	116.53174	39.91536
586	1	2008-02-08 15:41:31	116.57156	39.90263
587	1	2008-02-08 15:51:31	116.54723	39.90841

```
dt[dt['x'] < 116.5]
```

	tid	date	x	y
4	1	2008-02-02 16:06:08	116.47186	39.91248
5	1	2008-02-02 16:16:08	116.47217	39.92498
6	1	2008-02-02 16:26:08	116.47179	39.90718
7	1	2008-02-02 16:36:08	116.45617	39.90531
8	1	2008-02-02 17:00:24	116.47191	39.90577
...	...	...	...	...
579	1	2008-02-08 14:31:31	116.48503	39.91422
580	1	2008-02-08 14:41:32	116.44460	39.92156
581	1	2008-02-08 14:51:32	116.40047	39.92594
582	1	2008-02-08 15:01:31	116.44152	39.93236
583	1	2008-02-08 15:11:31	116.48347	39.91954

In the context of Python with the Pandas framework a programming environment is available, that provides some of the features found in relational DBMS, for example the capability to perform declarative queries on data tables.

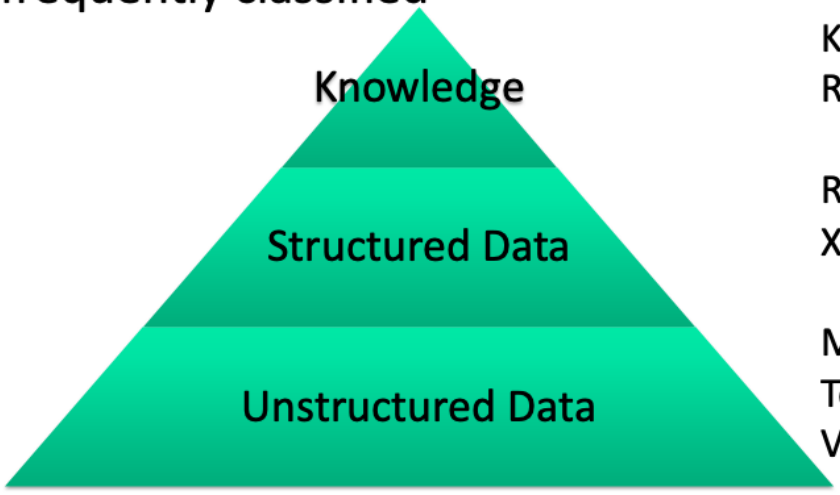
Q: What is information management?

- We have seen what are data management tasks. What are then information management tasks?
- Which tasks an information management system has to support?

- *Discuss with your neighbor to try to agree on some answer*
- *Write your answers into the Zoom chat*

3.2 Information Management

Depending on the “degree of abstraction” data is frequently classified



Knowledge Graphs
Rules

Relational data
XML and RDF data

Measurement data
Text, Images and Video

There exist conventions to distinguish different levels of abstractions for data. One can think of these levels of abstraction as levels of providing “higher level meaning“ to “raw data” that has been recorded or captured. These interpretations are generated either by automated or human analysis of the data, or a combination of the two.

This convention for classifying different types of data refers a priori to data in the sense of conceptual data. For each of the three levels, however, specific logical data models (e.g. graphs, relations) are preferably used.

Levels of Abstraction - Characteristics

Unstructured data

- Data captured from measurements and human input
- Fixed data structure (e.g., time series)

Structured data

- Data structure defined using a data model
- Captures relationships in the data

Knowledge

- Schema can evolve dynamically as knowledge expands
- Captures decision rules, objectives and intentions

The classification is not accurate!

Unstructured data is usually referred to as data that originates directly from some interaction with the real-world environment, be it through measurements (sensors, images, videos), human inputs (natural language, text, query logs), or machine input (system logs). The data is in fact structured (e.g., an image is an array of pixels, or a text is a sequence of characters), but the structure is considered to carry little “semantic” meaning.

Structured data is data that is modelled based on a schema the represents different categories and relationships in the data. A standard example being the relational data model, where applications or users define tables with specific meaning. Structured data could be extracted from unstructured data, e.g., a table could contain the list of all objects recognized in an image.

Knowledge is also structured data and considered as a basis for decision making. Other properties associated with the notion of knowledge are the ability to reason with it, its ability to evolve dynamically including changing its schema and the derivation of new knowledge from existing knowledge through inference. Since the representation of knowledge is deemed to be more flexible, usually graph-based data models are used, that allow to represent in an unconstrained way new relationships.

Note: the notion of “knowledge management” is used in business typically with a different, though related, meaning. It refers to information systems that manage the knowledge of an organization, consisting, e.g., of its documents and information on the skills of its collaborators.

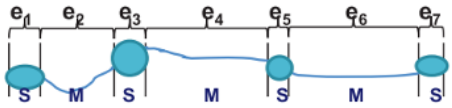
Example

Unstructured data: a GPS trace

a sequence of GPS point (x,y,t)

Bob's and Alice's GPS trace

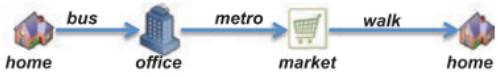
Structured data: Road segments, Places



a sequence of episodes

Places that Bob and Alice visit

Knowledge: Concepts and Inferences



episodes with semantic annotations

Bob and Alice are frequently together in Ouchy, thus: *Bob loves Alice*

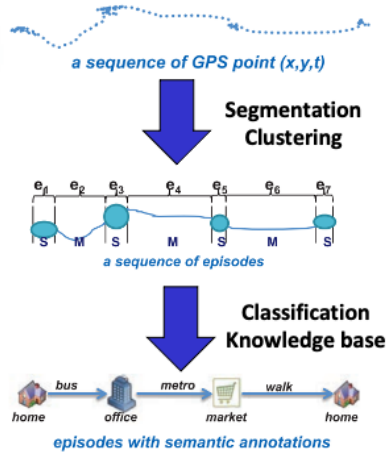
This examples illustrates how such abstractions can look like in practice. We can consider GPS traces as raw, unstructured data. Unstructured expresses here the fact that apart from the physical location we have no further understanding of the meaning of the data. By using automated analysis (e.g. segmentation methods) and background knowledge (e.g. maps) one can extract information about places (e.g. where GPS coordinates did not change for a period) and roads / paths where movement is detecting. Aligning such structured data with maps can give an interpretation of where the person was and by what means it moved, which then constitutes knowledge. Inferences on such knowledge might then reveal relationships among persons and produce high level knowledge such as "Bob loves Alice".

Model Building

Creating "higher level abstractions" from "lower level data"

- Using Statistical, Machine Learning methods, rule-based approaches, …
- Typically on large datasets
- Also called: data mining, data science, data analytics etc.

Create new models that represent the real world in a different way

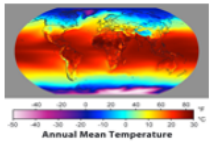


Deriving higher level abstractions or new interpretations of existing data is a central problem of information management. We will call this activity Model Building. Model building relies on numerous methods, both automated and with human intervention. We will introduce in this lecture many examples of such methods from the areas of data mining, information extraction and knowledge inference.

Model building is a central task in data science.

Information Management

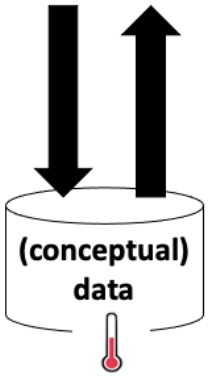
Tasks
Search
Filtering
Classification
Prediction
Recommendation
Summarization
Integration
Question answering



model M

Model Usage: given a model, perform the task

Example:
Count the number of stops of a trip.



Model Building: given data, derive a model that supports the task

Example:
Given GPS traces, find places and road segments

Since the central role of an information system is to create a model of the real-world environment based on data, the key information management tasks are related to the generation of models that derive from available data, new data that is suitable to support a task at hand.

Using then the model to perform the task is what we call Model Usage. Using a model consist of providing input to the model and to infer data that is then used for decision making.

Example: Trajectories

Original model: trajectory is a partial function $tr: T \rightarrow S$

New Models

- Interpolation: $tr_{int}: T \rightarrow S$ (same domains)
- Extracting trips:
  $trip = (t_{start}, t_{end}, s_{start}, s_{start}) \in T \times T \times S \times S$
  - The trips can be obtained by analyzing the velocity of position data

- Task: Counting the number of stops

Methods: Data mining, Machine Learning, Rules, Statistics

This examples illustrates in more detail the step of model building for the trajectory example. Starting from a raw trajectory (which we may consider as unstructured data), we extract a more structured model of a trajectory, distinguishing places and movements among those places.

New Models generate New Data

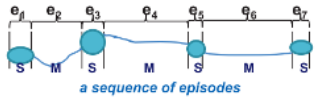
Model (Information System)

Data (Database System)

Obtain data from measurement

x	y	t
12	13	5:00
12	14	5:01
13	15	5:02

Derive new model from data



Data corresponding to the model

place	x	y	start	end
p111	12	13	2:00	2:10
p112	13	15	2:25	2:30

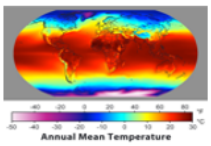
Derive new model from data

Data corresponding to the model

Subject	Relation	Object
home	ISA	Place
bus	ISA	Vehicle
Bob	ISAT	home

Every new model, generates new data, by transforming the original data into data corresponding to the new model. Then for performing specific tasks functions can be evaluated on this news data, generating additional new data.

Interpretation of Models



model M

data

Conceptual modeling: analyze the real world and specify a model

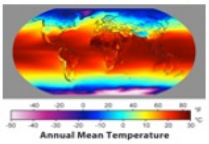
Example: define concepts temperature, location, measurement

Evaluation: given a model, evaluate it against reality

Example: compare predicted stops to real stops

A second important task in information management consists of establishing the connection between the model used in an information system and the real world. In other words it is about establishing and validating that the models correctly represents the real world aspect. This is possible only in indirect ways and can go two ways. First, if we do not yet have a model, we need to observe and analyse the real world and (intellectually) derive models. For example, in the case of GPS trajectories we would identify the concepts road segments as key concepts and represent them in a model. On the other hand, given a model we would like to verify whether the model is correct, for example, whether it produces prediction that correspond to reality. In our running example on trajectories this could be achieved by comparing the predicted stops with ground truth data on stops that has been obtained manually. Evaluating models is both an important task for information retrieval (verifying that the models used are correct) and data mining (verified that newly generated models are correct). We will discuss detailed methods for performing these tasks in this lecture.

Interacting with the Real World



Building systems where the data interacts with the real world

model M

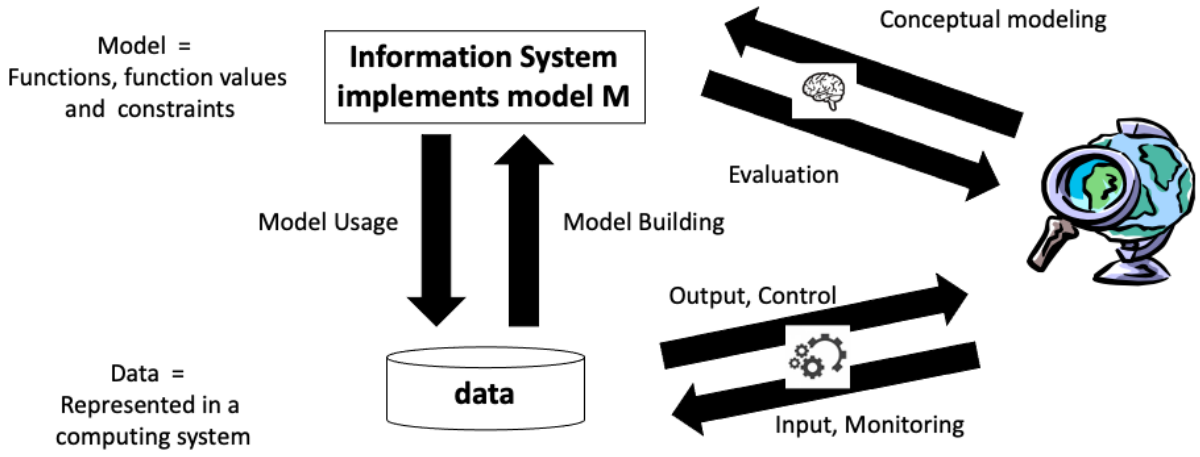
Output: data visualization, control
Example: control an autonomous vehicle

data

Input: users, sensors

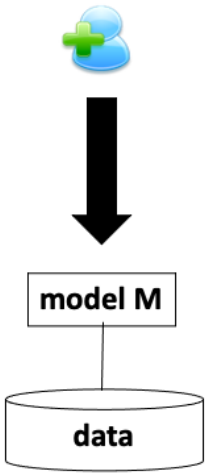
Regarding the interaction between an information system and their real world environment, we can distinguish the interaction with human users and direct interaction with the real world environment via sensors and control devices. Interaction with humans concern both the presentation of information to human users, i.e. through data visualizations and the collection of input from humans through user interfaces. Direct interactions with the real-world environment is what is called the Internet of Things, where computers receive data through sensors and control directly devices.

Information Management



Here we summarize the main information management tasks, and of how they interconnect the models, the data and the real world.

Utility



Users need information system to take **decisions**

Utility of information linked to the value achieved

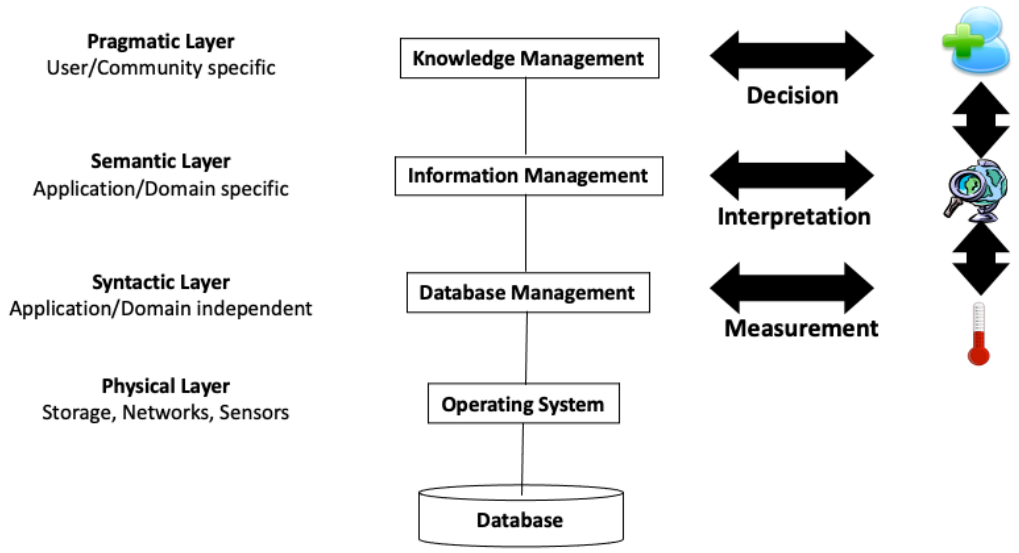
Value depends on
- Importance of decision
- Quality of decision

Quality of decision depends on quality and understandability of information!

Using information systems for decision making is associated with the notion of **knowledge management**.

Finally we are coming back to the issue of the purpose of an information system. Information systems are needed to make informed decisions. Thus their reason to exist is to provide information helpful for decision making. The **utility of information** depends on the nature of the decision on the one hand, and on the quality of the decision that is possible based on the information on the other hand. Information systems can make such utility measures explicitly available in different forms. Systems for rating and recommendation, e.g. in social networks, are an example where quality of information is handled explicitly.

Refined View of an Information System



Considering the information management tasks mentioned before we can refine our view on information systems by several aspects. We can add an additional layer that corresponds to the users of the system, and that we call the pragmatic layer as users introduce the dimension of information utility. Information utility is addressed in areas such as data quality management, agent systems, social networks.

Q: What does distribution mean?

- What does the notion of distribution in information systems designate?
- What are the implications of distribution?
- What makes distributed information sytems different from non-distributed information systems?

- *Discuss with your neighbor to try to agree on some answer*
- *Write your answers into the Zoom chat*

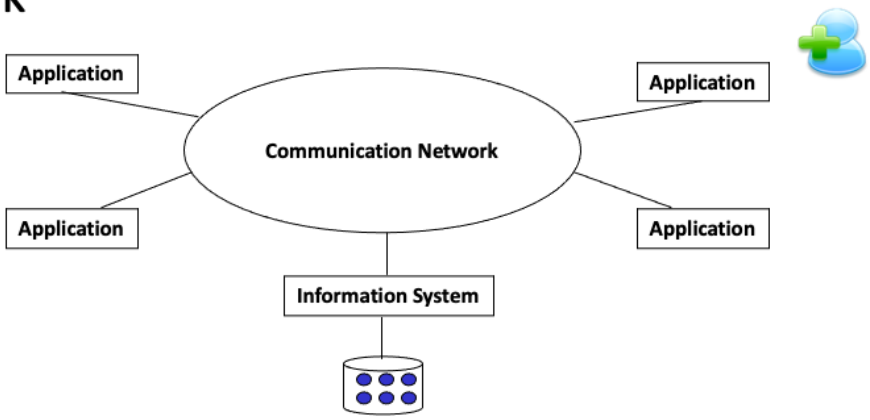
3.3 Distributed Information Management

Distribution can occur at different levels

- Network
- Data
- Models
- Control

Centralized Information System

Centralized Information System on Computer Network

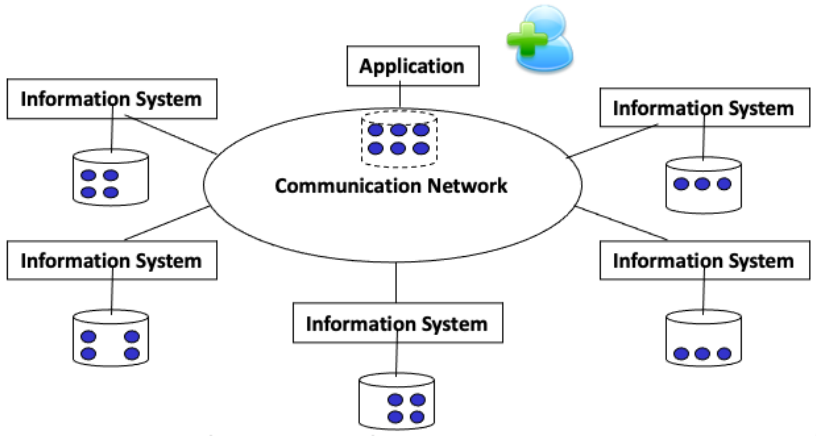


Except in the very early days, information systems had always been used in computer networks. From the perspective of information management this has no significant implications, as long as the information system itself is centralized, i.e., running on one physical node under a single authority. The network just enables the access of a user to the information system from a remote location.

Physical Distribution

Use of distributed physical resources: locality of access, scalability, parallelism in the execution

Distributed Data Management



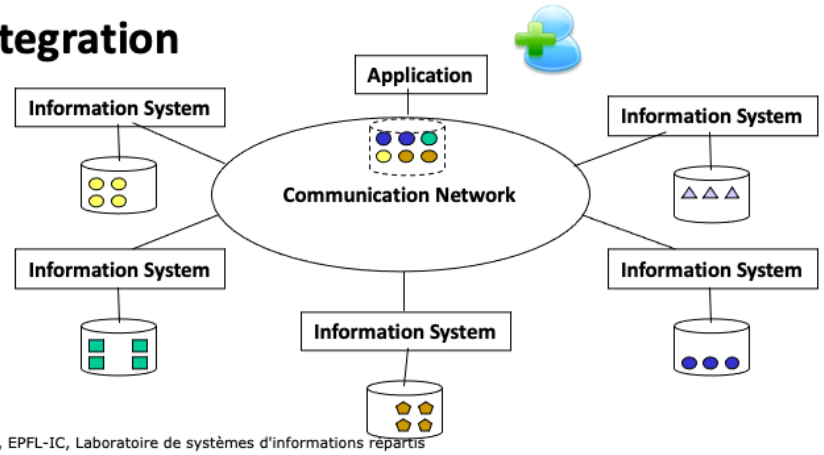
There exist however many reasons not to store all data of an information system in a single node of the network. Some of these reasons are related to the optimized use of available resources. We might want to move data in the network close to the node where it is accessed, we might want to take advantage of parallel processing of the data, and we might want to avoid bottlenecks in order to improve scalability of the system. All these are good reasons to distribute the data physically. However, physical distribution should be ideally fully transparent to the user. The user has still the impression of accessing a single information system that is running under a single authority. This model of distributed processing of data is the subject of **distributed data management**.

Logical Distribution

Use of different data models: **semantic heterogeneity**

- Independently developed information systems
- Different models for related concepts

Data integration

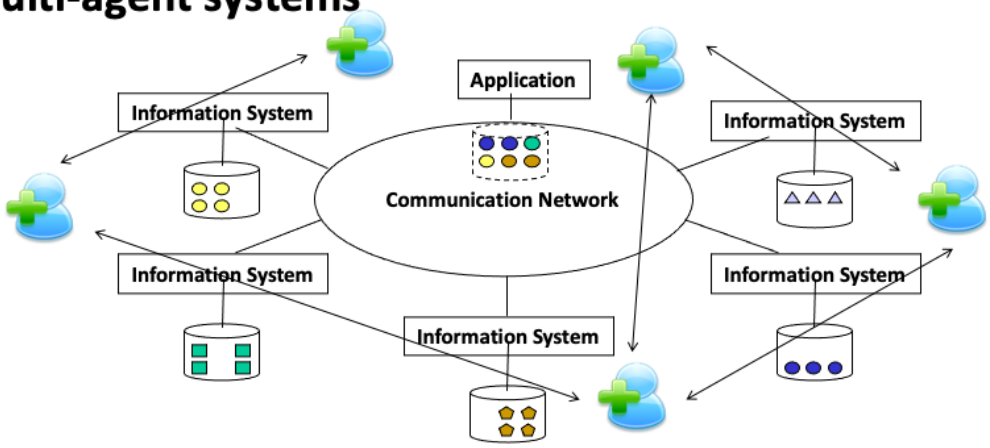


Having a homogeneous view of a distributed information systems might not always be possible. If we want to access information in systems that have been developed by different entities, or if we want to integrate information from different information systems that have been independently developed, we can no longer assure that the information can be homogeneously accessed. The same information might be represented using different models and data structures, and the access methods might be different. In that case we are talking about **heterogeneous information systems**. The heterogeneity results from a distribution of the decision authority when designing the system. In order to overcome heterogeneity methods for integrating data and making information systems interoperable are needed.

Autonomy – Distribution of Control

Independent users have to collaborate, coordinate, negotiate, to perform information management tasks

Multi-agent systems



Finally in heterogeneous information systems we must deal with the problem that the different information systems are under the control of different **autonomous** authorities. This poses problems of coordination, mutual trust, and privacy protection. The information systems can be considered as independent agents, that may pursue common goals, while protecting their own interests.

Key Issues in Distributed Data Management

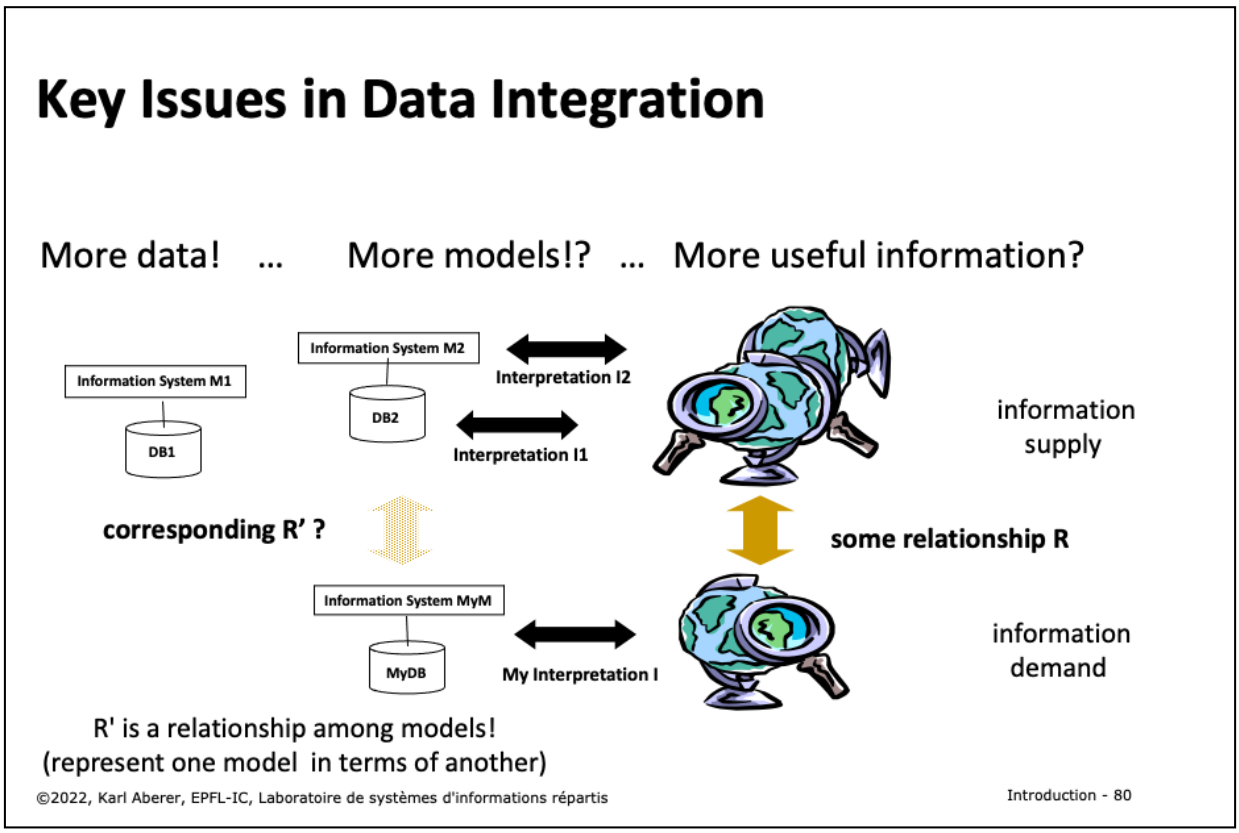
Where to store data in the network?

- **Partitioning** of data
- **Replication** and **caching**
- Considering typical access patterns and data distributions

How to access data in the network?

- **Push** vs. **pull** access (query vs. filtering)
- Indexing of data in the network
- Distribution of queries and filters
- Considering the communication model

Distributed data management deals with similar questions as centralized data management, namely optimizing the storage of the data and the processing of accesses of the data. The new key element is that data can now be stored at different nodes in a network, and that the cost of data transmission over networks becomes an essential performance consideration. Since cost of data transmission is generally considered as expensive, in distributed data management often multiple copies of the same data are kept at different locations in order to speed up access or data is partitioned to bring the data closer to the application that use different parts of the data. This in turn implies new problems of keeping distributed data consistent while executing transactions that involve different nodes in the network. The area of distributed transaction management is dealing with this problem.



We are living in the age of Big Data. More and more data is being produced, data is becoming more easily and directly accessible, data intermediaries are disappearing and users obtain direct access to data sources (disintermediation). The question is whether we have automatically more information. This is not clear

Since the growing number of data comes also from a growing number of different sources, the data corresponds to a growing number of models and the models in which the data is provided do not necessarily match the models required to address a task or to understand the meaning of the data. There exist many information systems supplying data, but each having their own representation on the world, which does not necessarily match the demands of a specific consumer.

Since the different sources of data use all models that relate to some aspect of the real world that are related to each other, there exists also a relationship between the different models for translating between the different ways the same real-world aspect is represented in different information systems using their specific models.

The Problem

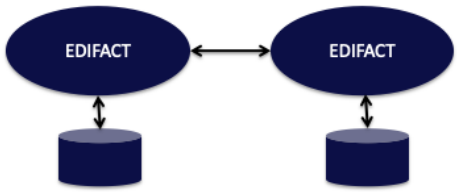
Semantic heterogeneity

- The same real world aspect can be represented differently
- Requires agreement on the meaning of shared data
- Relating different models (and thus different representations and their interpretations) requires often human intervention
- human attention is a scarce resource !

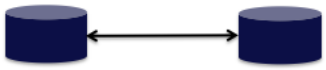
Relating different models used in information systems to each other is solving the problem of **semantic heterogeneity**. This problem is as hard as creating new models for information systems and requires typically human intervention, thus the scarcest resource we have available.

Mapping: Three Approaches

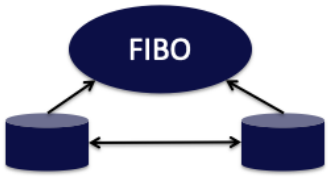
1. Standardization
   - Mapping through standards



2. Mapping
   - Direct mapping



3. Ontologies
   - Mediated mapping

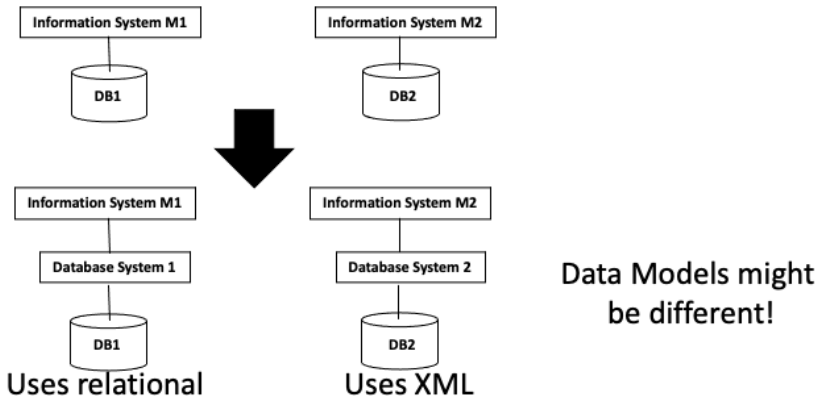


Conceptually there exists three main possibilities of how to address semantic heterogeneity. A first approach is to map all the models everything to one common global model. This approach is taken with standardization. For example, EDIFACT (https://www.unece.org/cefact/edifact/) is an international standard that models all concepts that are commonly used in business and trade. For exchanging information systems used in that domain map their data to EDIFACT and can thus exchange their information. A second approach consists of trying to construct directly a mapping among two information systems, without having any additional, shared knowledge in form of standards or ontologies among the two information systems. For example, in industry often specialized software is developed to perform this task. A third approach, is to relate the model of an information system to a common model, frequently called ontology, and use this mapping to construct a direct mapping among the different models used in the information systems. For example, in the financial domain there exists a standard reference ontology called FIBO (https://fib-dm.com/finance-ontology-transform-data-model/).

More Problems?

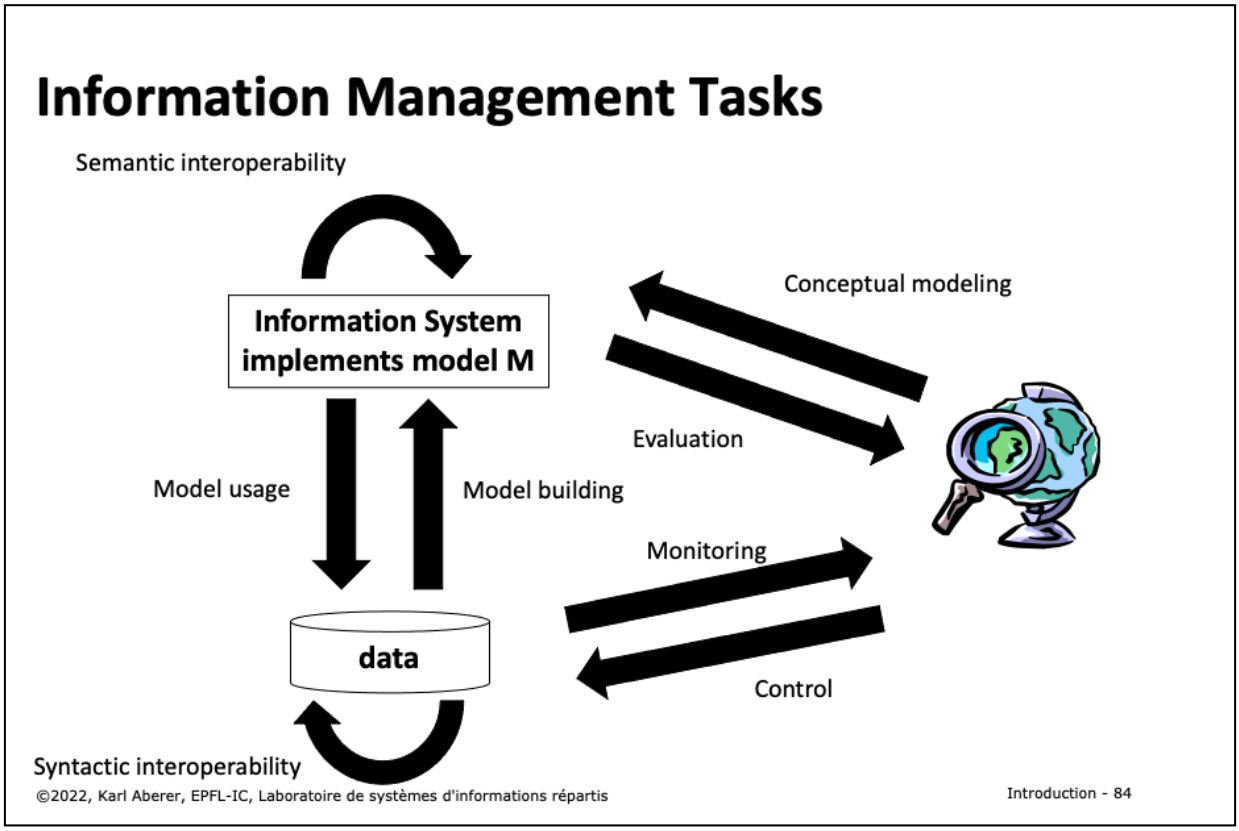
Syntactic heterogeneity

– The same conceptual model can be represented using different logical data models

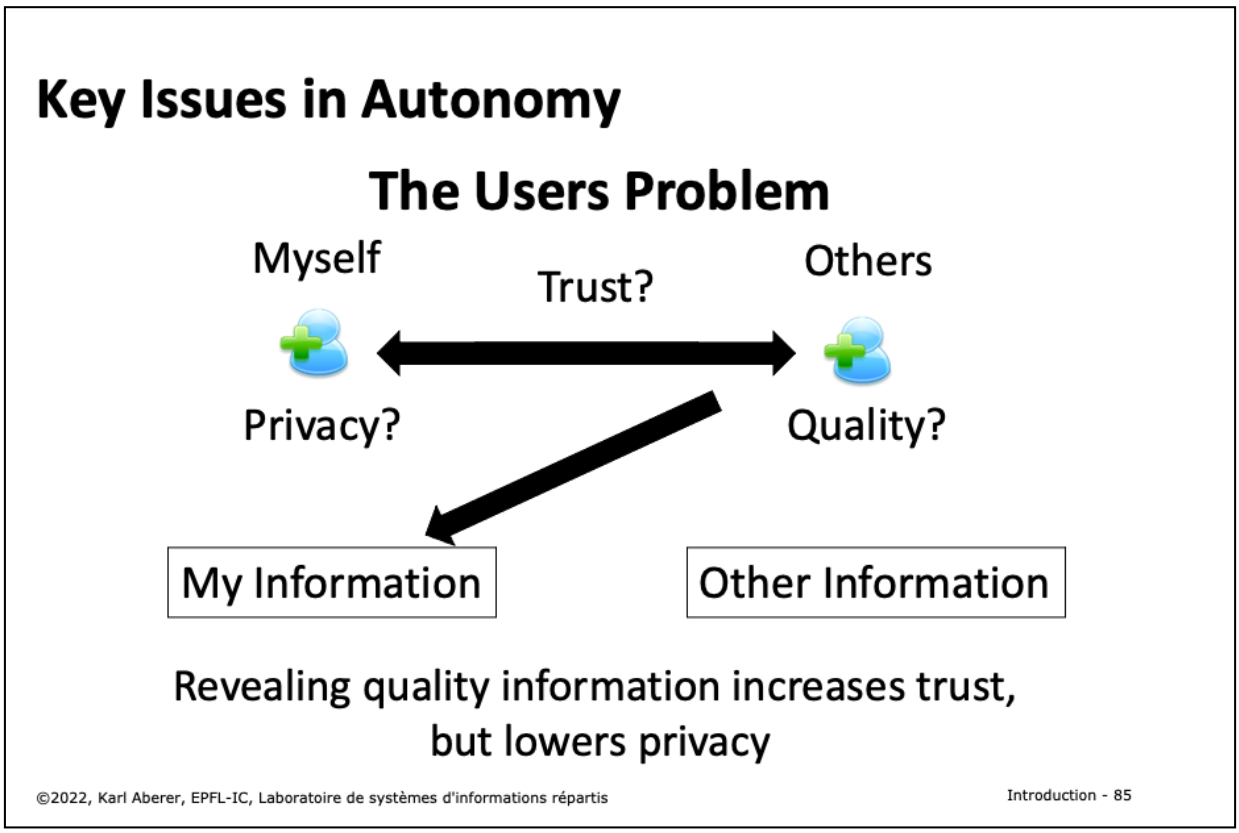


To complete the picture we have also to mention that heterogeneity among information systems cannot only occur due to the fact that the same real world aspects are modelled differently, but also due to the simple problem of using different underlying data models to represent the chosen model. In that case we are talking of **syntactic heterogeneity**. For overcoming syntactic heterogeneity mappings among different data models are needed. This problem is somewhat simpler than solving semantic heterogeneity, since the relationship among different data models can be treated completely within a formal context. But it is nevertheless not completely trivial, as different data models offer different data structures to store the same data and transformations might be algorithmically complex. The problem has been studied in different contexts including:

- Storage of programming language objects (e.g. Java) in database systems (e.g. relational)
- Integrating data from different types of data management systems, using different data modelling formalisms (e.g. relational, hierarchical, XML)
- Storing different types of data (e.g. XML, graphs, arrays) in generic databases management systems (e.g. relational)
- Exchanging data from database systems (e.g. relational) through document formats (e.g. XML)
- Representing graph-oriented models (e.g. RDF) as documents (e.g. XML)



Semantic and syntactic interoperability are two additional tasks in information management that we can add to our global picture.

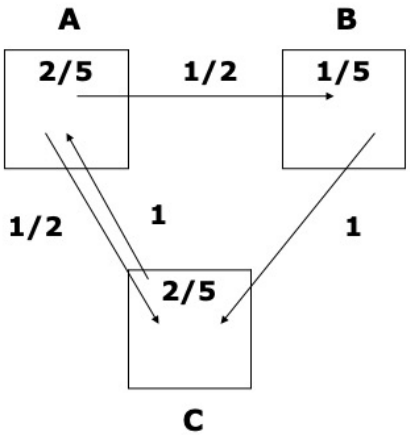


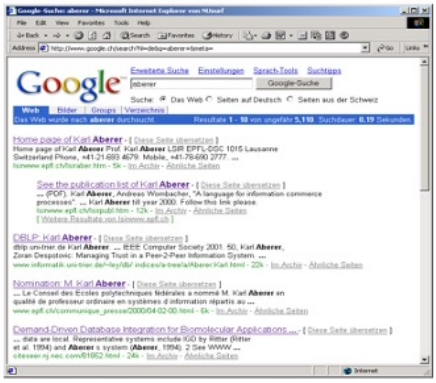
In a setting where different autonomous users exchange information, questions related to the utility of information come into play. As users have their own private interest, they have to consider them in interactions with other users. For example, when receiving information from another user, a fundamental question is whether the information can be trusted. It might be that the other user might have an interested to provide wrong information in order to obtain an advantage. When providing information to another user a different problem needs to be considered. Can we trust the other user to use the information properly, or will the information be used in unintended and potentially harmful ways. This is the privacy problem, and it is a fundamental challenge for an information society.

The two problems of information trust and privacy are related to each other. The more quality information we reveal the more trust we may expect to receive in return, but the more we also put our privacy at risk.

Evaluating Quality of Information

Recommendations (e.g. Google PageRank)





One way to evaluate the quality of information, and thus the level of trust we can have in a user providing information, is to share recommendations with other users on how they perceive the quality of this information. This is what, in principle, Google has been doing by introducing Pagerank, a method to assess the quality of a Web page by considering the Web links that point to that Web page. The underlying idea is that the more Web pages refer to one page, the more trustworthy it is and at the same time also to consider the trustworthiness of the recommender itself.

Evaluating Trust

Reputation-based trust: if users behaved honestly in previous interactions, they will do so in the future

Overall profile makeup

94 positives. **91** are from unique users and count toward the final rating.

4 neutrals. **0** are from users no longer registered.

1 negatives. **1** are from unique users and count toward the final rating.

ebay ID card — seller01 (90)

Member since Tuesday, Dec 15, 1998

Summary of Most Recent Comments

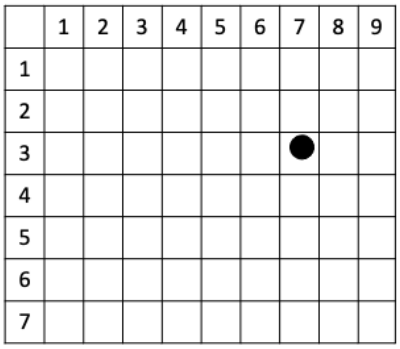
	Past 7 days	Past month	Past 6 mo.
Positive	2	3	15
Neutral	0	0	0
Negative	0	0	0
Total	**2**	**3**	**15**
Bid Retractions	0	0	0

Another way to evaluate trust is to analyse earlier behaviours of users. This is, for example, widely used in ecommerce sites, where users can provide ratings for vendors. The underlying assumption is, that if vendors have behaved well in the past they will also do so in the future. From a vendors perspective such ratings of course foster honest behaviour, as negative ratings would affect the future business. Evaluating trust on such histories of behaviours, is called reputation-based trust, where the reputation is based on or corresponds to the data gathered about a user.

Protecting Privacy

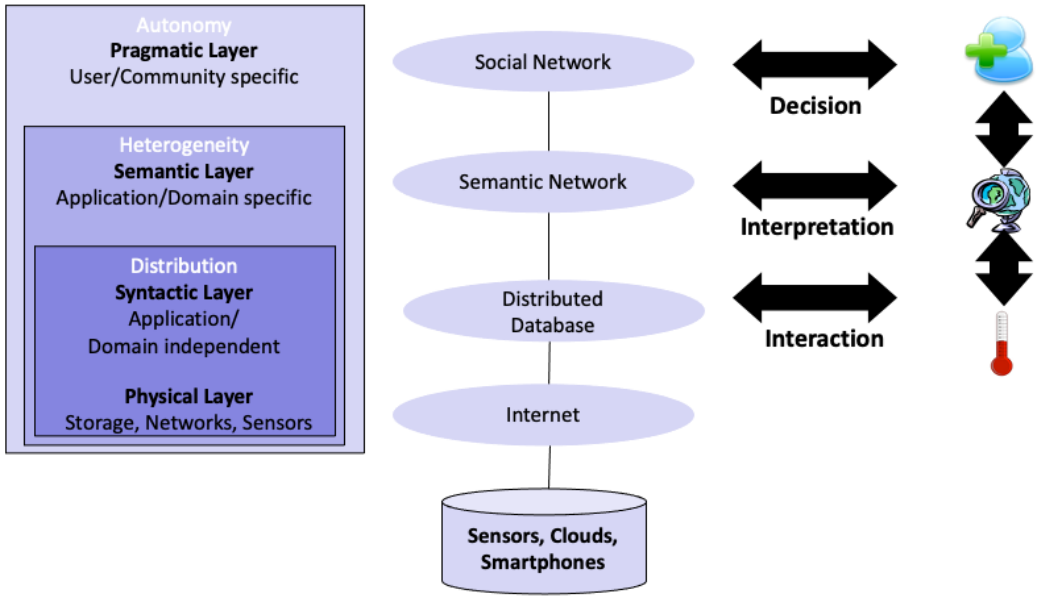
Example: location privacy – obfuscation methods

- Perturbation: (3,7)
- Adding dummy regions: (3,5), (1,4), (6,3)
- Reducing precision: (2,5), (3,4), (3,5), (3,6), (4,5)



In order to deal with privacy, methods such as obfuscation are used to provide sufficient information to obtain a useful service, but not so much that sharing the information may be harmful. Here we illustrate different methods that could be used to obfuscate a GPS location that is reported, e.g., when using a mobile service. Other methods to protect privacy include access control and data anonymization.

Refined View of a Distributed Information System



4. ABOUT THE LECTURE

Focus of the Lecture

Distribution: How to share information?

How to deal with semantics and models in a distributed information system?

1. Use a common model
2. Create relationships among models
3. Use a common reference model

Today's information systems are inherently distributed, since they serve the sharing of information among humans (and increasingly machines). Therefore, the focus of the lecture is in the question of sharing of information. This question implies in particular the problem of dealing with the representation of this information in different models. Following the characterization of the possible ways of sharing information exposed earlier, the lecture will comprise three parts, each corresponding to one of the three ways.

1. Natural Language as Common Model

NL is our common mechanism for communication

- Can express any fact, but also intentions, beliefs etc.
- Same fact can be represented in diverse ways

Information retrieval: interpret text for search (part 1)

- Relationships among texts

Information extraction: extract facts from text (part 3)

- Relationship among text and factual data

What concerns the use of a common model, the by far most important model for shared representation of information is natural language. Therefore, in the first part we will focus on using NL for sharing of information, which is a key problem that has been extensively studied in the field of information retrieval. We will later also explore methods for extracting structured data from natural language, which is the subject of the field of information extraction.

2. Create Relationships among Models

Represent one model in terms of another one

- Data mining: extract new models using data from existing ones (part 2)
- Data integration: determine relationships among models (part 3)

In the second part of the lecture, we will study methods to extract models of higher abstraction level from large amounts of data typically collected in a distributed setting and generated by large numbers of actors. This data can be text in social media, social graphs, transactions in ecommerce or product evaluations.

3. Shared formal representation

Knowledge Graphs (Ontologies) (part 3)

- Formally represent knowledge extracted from data or provided by humans
- Facilitates the creation of mappings among models and the derivation of new facts

The third part of the lecture will be devoted to common abstract reference models, which today are typically represented as knowledge graphs. These can be either human-created or machine-generated.

Overview of the Lecture

Part 1: Information retrieval

- Understanding natural language

Part 2: Mining unstructured data

- Inferring structure from data aggregated from distributed sources
- Social graph mining (clustering), Document classification, Recommender systems (prediction), Association rule mining

Part 3: Knowledge graphs

- Creating and using shared formal models

This summarizes the contents and the underlying conceptual framework of the lecture.