

Introduction:

Distributed Information Systems

An Overview

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 1

Overview

- 1. What is an Information System?**
- 2. Data Management**
- 3. Information Management**
- 4. Distributed Information Management**

Objectives:

Understand the difference among an IS and other IT systems.

Understand the role of IS for managing models

Understand different aspects of reality represented in information systems

Understand the constituents of a model

Understand that functions in information system are often explicitly represented

Understand the concept of interpretation of a model

Understand that having a correct model is a very difficult problem

1. WHAT IS AN INFORMATION SYSTEM?

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 3

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

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 4

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). A large organization such as EPFL is running dozens if not hundreds of information systems. With the advent of the Web and more recently 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

More recent types of information systems

- Social Networks
- Query-Answering System
- Recommender Engines
- Business intelligence (data warehousing and mining systems)
- Bioinformatics systems (e.g. genome or protein sequence retrieval)
- Environmental monitoring systems (disaster warning, meteo website)
- Publish-subscribe and data dissemination systems (e.g. RSS, mobile broadcast)
- Etc.

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

concept in the context of this course.

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?

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 relate to the design of computing systems in an enterprise context. The notion can be delineated from activities more directly related to their implementation.

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)

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 7

From a system perspective we can interpret 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



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]} 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.^[1]

Any specific information system aims to support operations, management and decision-making.^{[6][7]} 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.^[1]

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.^[1]

Alter^{[8][9]} 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.^[1]

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**.^[1]

2018

2021

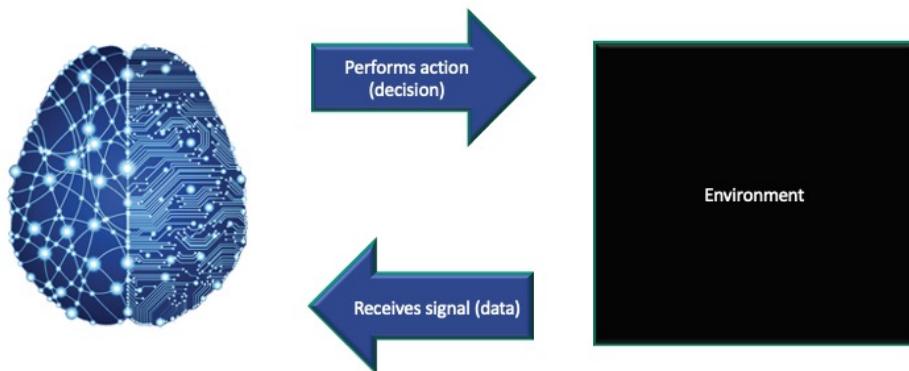
©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 8

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

Information Processing



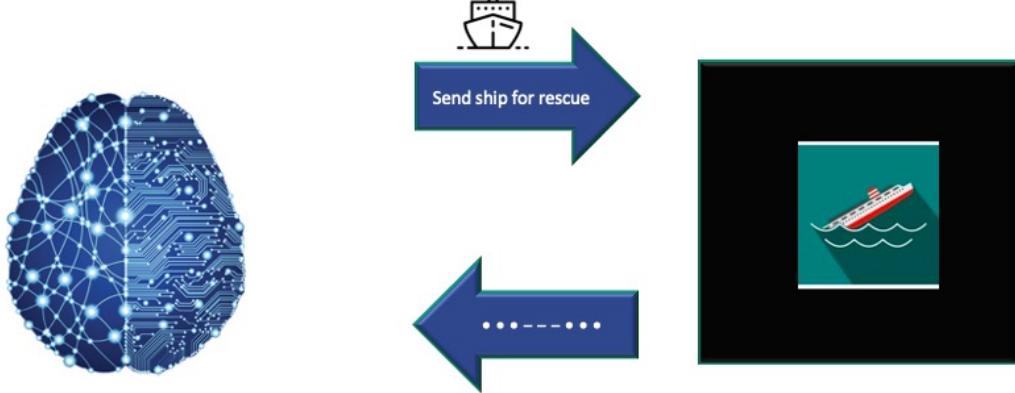
This model applies to all types of systems: organisms, species, humans, organizations, societies, etc.

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 9

A systemic perspective is rather focusing 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

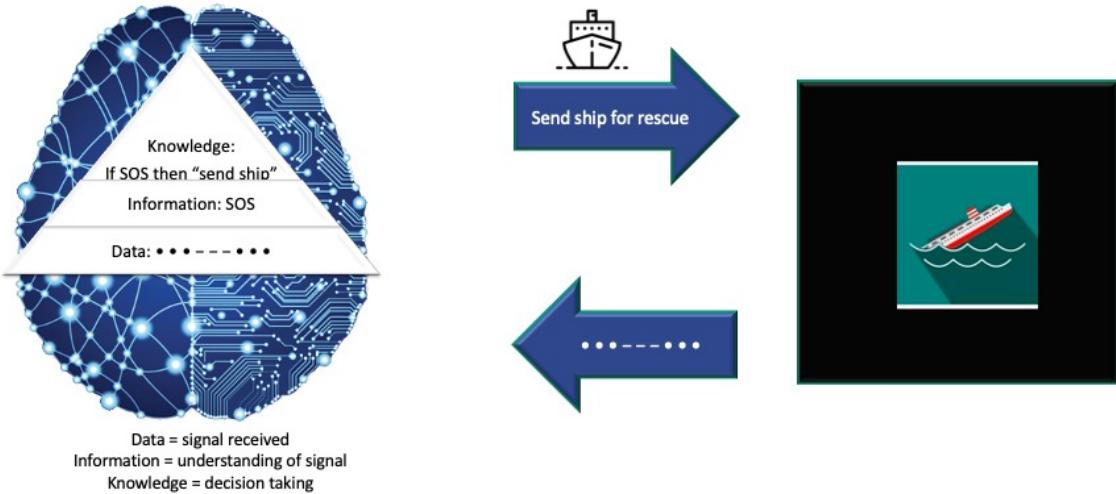


©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 10

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

Data-Information-Knowledge



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 11

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

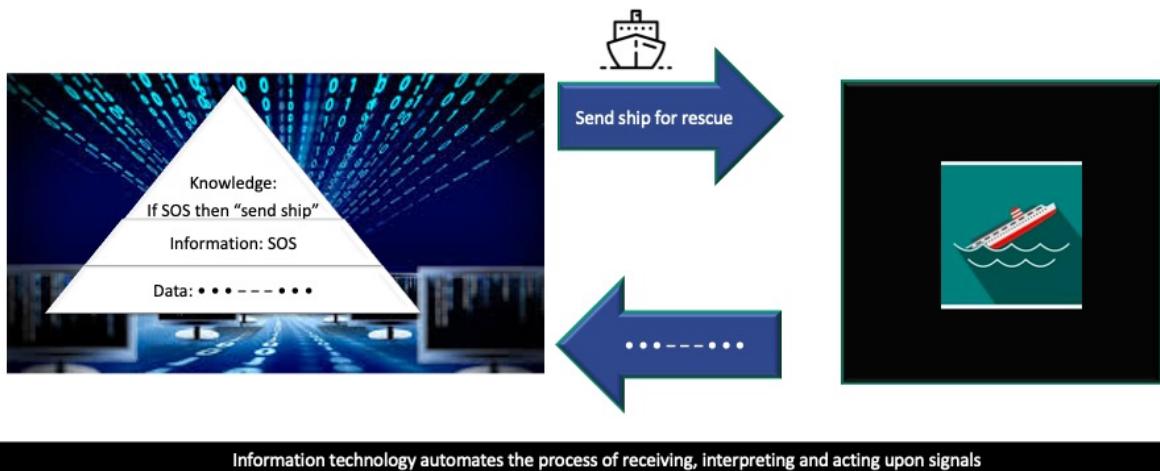
First, we see that the ship is sending the signal. This corresponds to the transmission of data, irrespective of how the data is transmitted, e.g. by radio, Internet, Morse, fume signals etc.

The receiver of the data (the Morse code) has then the task of interpreting the data (or signal). Only this creates what we will consider later as information. In this step the data receives a meaning. Without being able to perform this interpretation at the receiver, sending the signal would be useless.

Finally, understanding the message does not mean that the receiver also knows what to do. This is where we start to consider knowledge, i.e. that capability of reacting to information

received in a meaningful way with the environment.

Information Technology



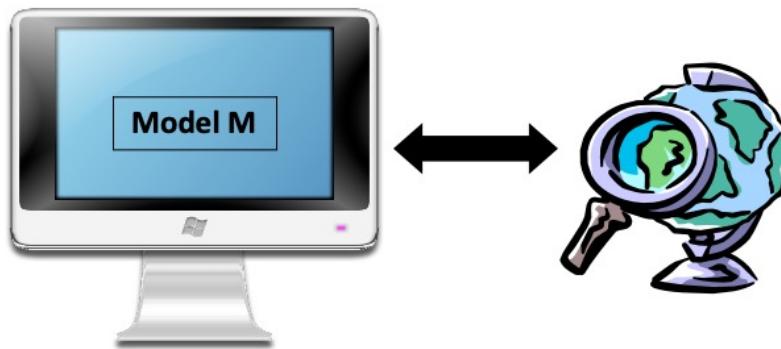
©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 12

The process described in this 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. The latter is what we will studying in the course of this lecture.

What is an Information System?

An information system is a **computing system** managing **data representing a model** of some aspect of the **real world** for a given **task**.



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 13

Based on the previous consideration we can come up with a working definition for information systems. For sure, information systems are software systems. In addition, one can quite safely state that all information systems represent within a computer system a model of a part of the world/environment they are interacting with. And this model is needed to fulfill some purpose. We base our definition on this : “An information system is a **computing system** that manages a representation of a **model** of (some aspect of) the **real world** within a (distributed) computer system (for a given **purpose**)”.

This definition involves a number of concepts that require further explanation:

real world: The notion of real world refers not necessarily to our physical environment only. It can be anything from abstract concepts (e.g. a legal information system) to technical systems including computer system or networks itself (e.g. information systems for network management).

purpose: every information system has an entity (human, computer) that makes use of it. It does so, in order to perform a certain task related to some aspect of the real world (e.g. making a decision, performing a computation etc.).

aspect: this implies that there exist many different ways to represent the real world and same aspects of the real world in information systems, depending on the purpose.

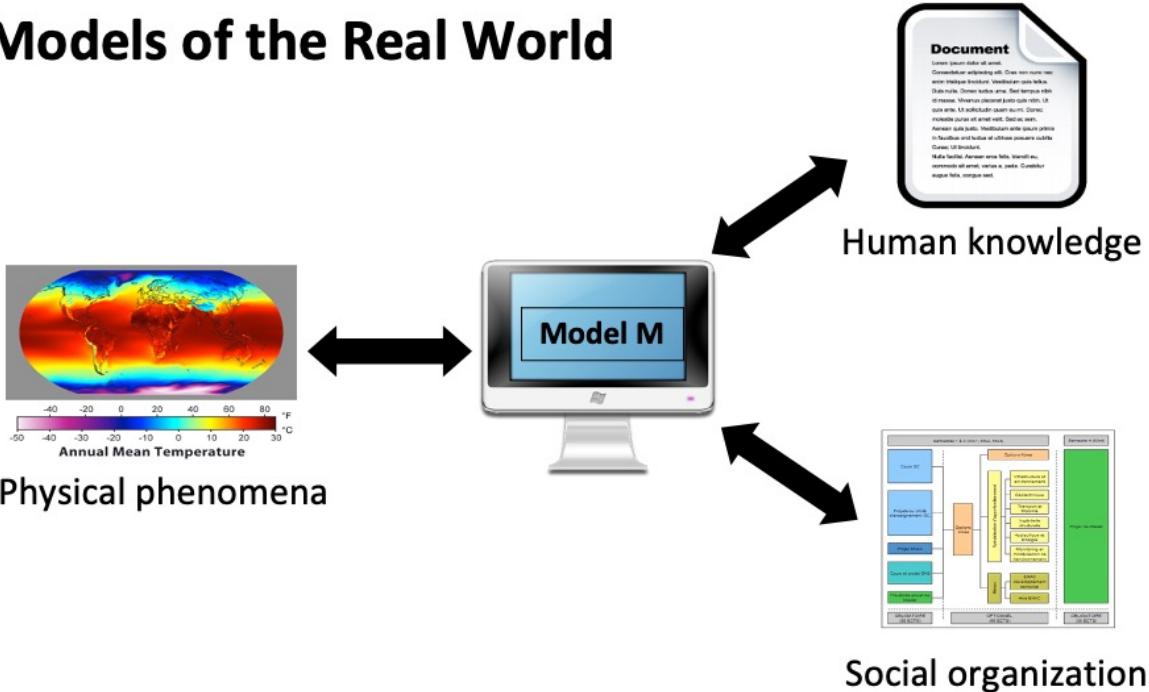
model: what a model is and what is its role we will explore in more detail in the following.

Compare to the definition provided by WikiPedia. It is in fact not so far from our definition.

Information systems can be defined as an

- integration of components for collection, storage and processing of data (-> software)
- of which the data is used to provide information (-> model),
- contribute to knowledge that facilitate decision making (-> purpose).

Models of the Real World



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 14

We can identify broadly three types of real-world aspects that are captured by information systems:

1. Physical phenomena: these information systems measure the environment 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 probably the earliest one that had wide-spread use, for applications such as finance, logistics etc.
3. Human thought: these information systems model human thought and reasoning processes. They enable to capture the meaning of text and other media, assess the importance and quality of information, but also model human traits such as sentiments or opinions. Information retrieval systems, of which web search engines are a specific example, are the typical representative of this class of systems.

Broadly information system represent physical phenomena, the social organization of humans and the result of human thought processes.

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

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 15

Information systems use models to represent some aspect of reality. But what is a model? The answer is simple: any (mathematical) structure can serve as a model.

A mathematical structure consists of elements (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 provide names, respectively identifiers, to real world objects), the functions to provide properties of these things, 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 for information systems as generic approach. Models based on first order logic allow the to represent the important entities, their relationships and properties of the real world 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 graph models, vector spaces, probabilistic models, differential equations, simulation programs 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 have been developed to capture the structure and dynamics of business processes, also called workflows.

Some you have already encountered in courses on data management or software engineering, the use of

some others we will demonstrate later in this course.

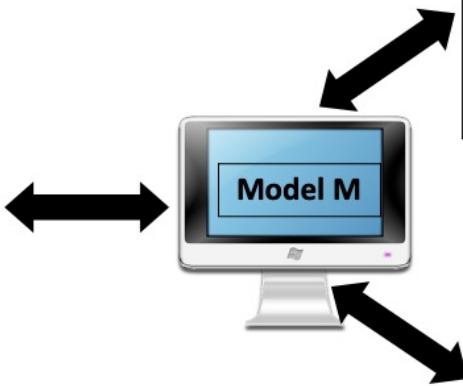
Examples of Models

Constants: coordinate values, temperature values

Function:
 $temp(x, y)$

Axiom:
 $-60 < temp(x, y) < 60$

Physical phenomena



Constants: document identifiers, text (sequence of characters)

Function:
 $similar(doc_1, doc_2)$

Axiom:
 $0 \leq similar(doc_1, doc_2) \leq 1$

Human knowledge

Constants: names of people and units

Function:
 $memberOf(name, unit)$

Axiom: each person belongs to at least one unit

Social organization

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 16

For our three running examples we can provide concrete instances of possible models. A temperature map we can model, for example, as a two-dimensional matrix. An organizational structure is best captured using relationships among entities, whereas documents can be modelled by their degree of similarity.

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

...

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 17

After providing some clarification of the notion of “model”, let’s consider what is a “task”.

Obviously, we can every concrete information system, like the ones we have listed as examples in the beginning, identify which is the concrete purpose or task. From that we can identify more general classes of tasks, like search, recommendation etc.

When developing methods for information systems, it is always crucial to understand which type of task we have in mind. We will in the course of the lecture identify different types of tasks, and approaches to address them.

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

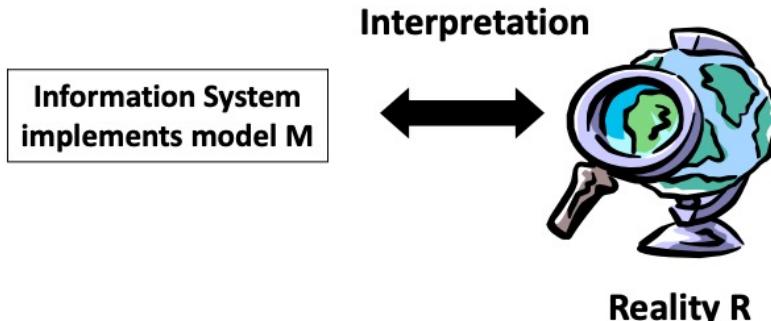
Sensors

Images and videos

Finally we have also to understand what we mean by data. One aspect is the type of data used in an information system. When looking again at the example, we will quickly identify the main classes of data. We will during the lecture also study methods that apply to some of this classes.

How do we know that a model is good?

The model should represent some aspect of the real world



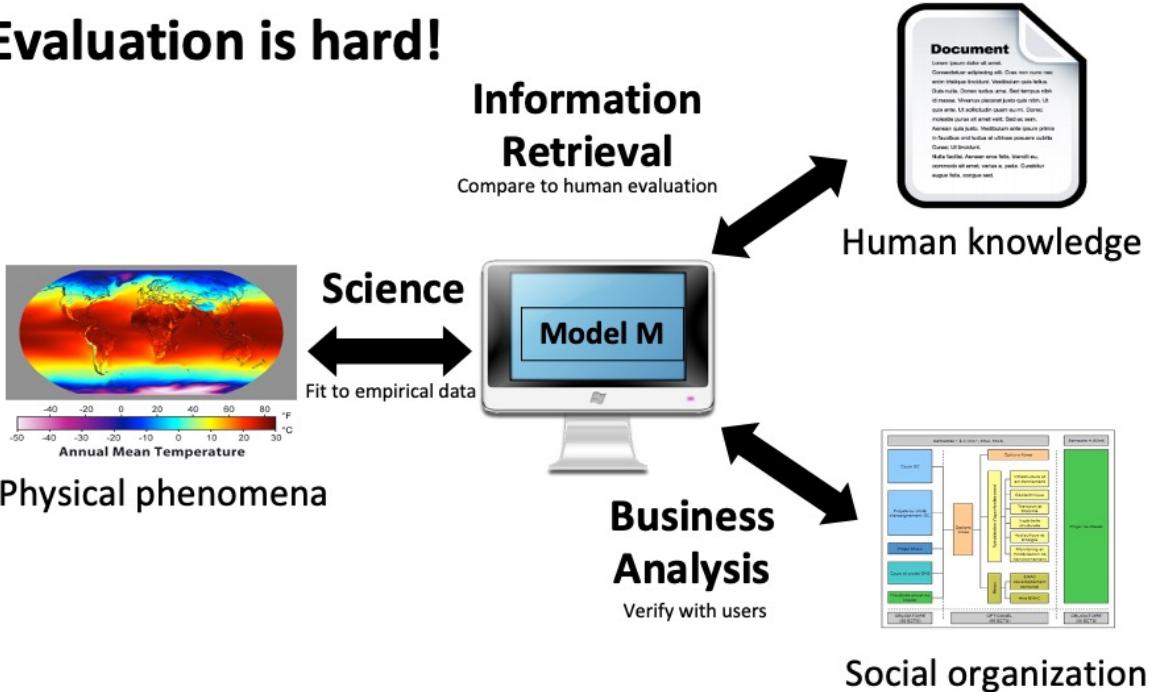
We have to **evaluate** whether the model represents reality properly

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 19

When building a model of the real world in an information system, a natural question to ask is of how we can know that the model is indeed a model of the real world. In an abstract sense the answer to this question is very simple: we have to provide an interpretation function, that maps every element of the model to some real world object, and preserves the properties of the model.

Evaluation is hard!



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 20

Since there is no way to formally define or verify functions from a mathematical structure into the real world indirect methods are required that help to verify, or at least make plausible, that a model represents correctly the real world. 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!).

Which is true? The relation between a model and the real world is

1. never computable
2. uniquely defined
3. not part of the model

Can you give an example of a computable interpretation relationship?

2. REPRESENTATION OF MODELS

Example: a Mathematical Model

Modelling a set of 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 $tr: T \rightarrow S$



In order to illustrate the representation of models in information system, we will use a simple, yet practical, example, the management of trajectories (this example will also be used in the exercises). Here we describe a basic trajectory model in a 2-dimensional space, as it is frequently used to describe, for example, vehicle trajectories.

Representation of Functions

Functions can be represented by a specification or algorithm or by enumerating the values it can take

- Functions are called quantitative or qualitative variables

The enumerated values are called **data**

Representing a single trajectory tr

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

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.

Actually, 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 actually 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.

Functions in Information Systems

Information systems strongly rely on representation of functions as data

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

Algorithmically defined functions play nevertheless an important role

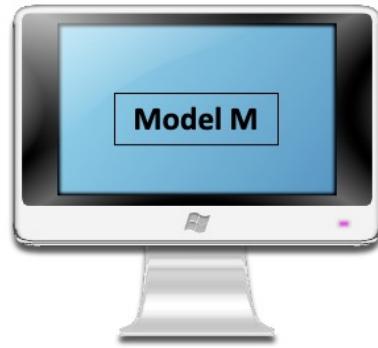
- models, queries, views etc.

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

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 26

So far we have considered models as abstract mathematical formalisms, consisting of constants, 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 formalism that can be used to represent a model. Such a formalism is called a **data model**. A data model consists of data structures and operations that a computer can represent and execute.

Representation of Mathematical Models

Representing the trajectory model as data structure

- Choose an appropriate (abstract) data type
- Map the model to the data type to obtain a data structure
- Data structure will be processed by computer

So, an essential step in understanding how information systems work, is to understand of how the initially conceived mathematical model is represented in terms of data structures.

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 $f: K \rightarrow V$

The study of data structures is at the heart of computer science. For example, the **associative array** is an abstract data structure (abstract data type) that is used to represent functions. Thus it is evidently of central interest for representing functions of models. Associative array is a data structure that manages a set of key-value pairs (representing the function) and that supports a set of operations to manipulate the associative array, such as adding, deleting and modifying the elements of the associative array. 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

$R: T \rightarrow \text{Float}$

time in seconds from some reference time

$R: S \rightarrow \text{Tuple}(\text{Float}, \text{Float})$

longitude, latitude in degrees

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

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

The example shows one possible way of representing our mathematical model of trajectories using concrete data structures. Note that we specify not only the representation mapping, 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 lead already to some major catastrophies.

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, ...

Representing the trajectory model as data structure in Python

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

Finally the data structures, as mathematically defined by abstract data types, cannot be directly represented in a computer system. Rather they have to be mapped to 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.

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]
```

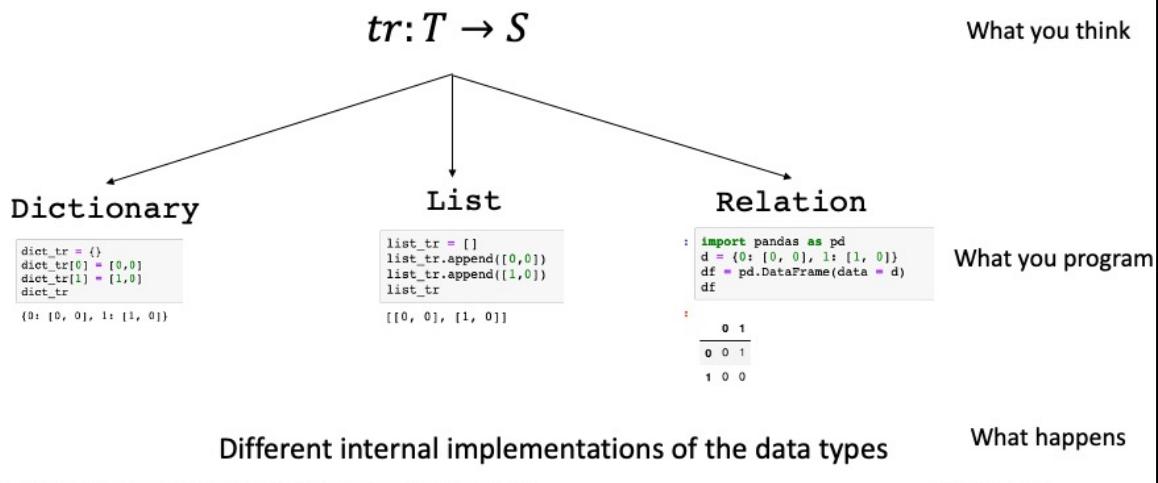
©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 32

In order to simplify the mapping from mathematical structures to an implementation object-oriented models can help. They allow to flexibly introduce new data structures corresponding to signatures of mathematical models, while hiding the details of the implementation (based on basic abstract data types or other objects). This is called encapsulation.

Relationship among Model and Data Model

The same function can be represented using different data structures



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 33

Data structures allow us 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 the software developer to enforce this constraint and typical operations in the software 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

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 34

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 first proposed 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

Interestingly, we can also relate the concept of “data” to these three modeling levels., which reveals the interesting fact that we are continuously using the notion of 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 process in a data processing 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 talking about data compression)

Representation of Mathematical Models

Mathematical models of information systems are represented 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, whereas typically initially the conceptual model is introduced to characterize the functional properties, it's (efficient) realization in terms of data structures is then developed, and finally further physical optimization can be performed at the binary level.

Functions in models ...

1. can always be computed
2. can always be represented as data
3. can sometimes be both computed and enumerated

Can you give a concrete example of a function in an information system, where the function can be alternatively computed or enumerated?

3. REPRESENTATION

What does “Representation” mean?

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. The concept of preservation of properties remains in this definition less clear.

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 :

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

This is one possible preserved property!

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

One very strong way to preserve properties are homomorphisms, a widely used concept from mathematics. Informally, homomorphisms preserve structural properties exactly. If the homomorphisms are in addition bijective, we have isomorphism. We can think of isomorphism as a mapping that shows that two structures are equivalent, i.e. essentially the same.

Example: Representing a Discrete Model

Mathematical model:

$\text{graph } G \subseteq D \times D$

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

Representation as abstract data type

$R: D \rightarrow \text{Integer}$

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

```
def neighbor(i, G):
```

```
... code to retrieve all neighbors ...
```

R is bijective, isomorphic!

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 41

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

Binary Representation

Data structure: `list(int)`

Example: `1, 8, 3, ...`

Binary representation:

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

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

Example: `001 100 010 ...`

R is bijective, isomorphic!

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 42

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

Data Exchange Format

Data structure: `list(int)`

Example: `1, 8, 3, ...`

JSON representation:

`'[1, 8, 3,...]'`

R is bijective!

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:

$R: T \rightarrow \text{Float}$

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
```

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

ntroduction - 44

However, for non-discrete mathematical structures the situation is less clear cut. One standard example are anomalies that we encounter in numerical computation, which result from the representation of non-discrete structure (real numbers) in a discrete domain.

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 real numbers), or on purpose. Often representations are used to simplify or approximate the original structures, to make them more amenable for processing.

This is exploited in so called “representation learning” where discrete structures are mapped into vector spaces, in order to benefit from vector space metrics and operations to capture properties that are hard to model or compute in the original structure. We will make use of this approach extensively for representing text data through vectors in information retrieval.

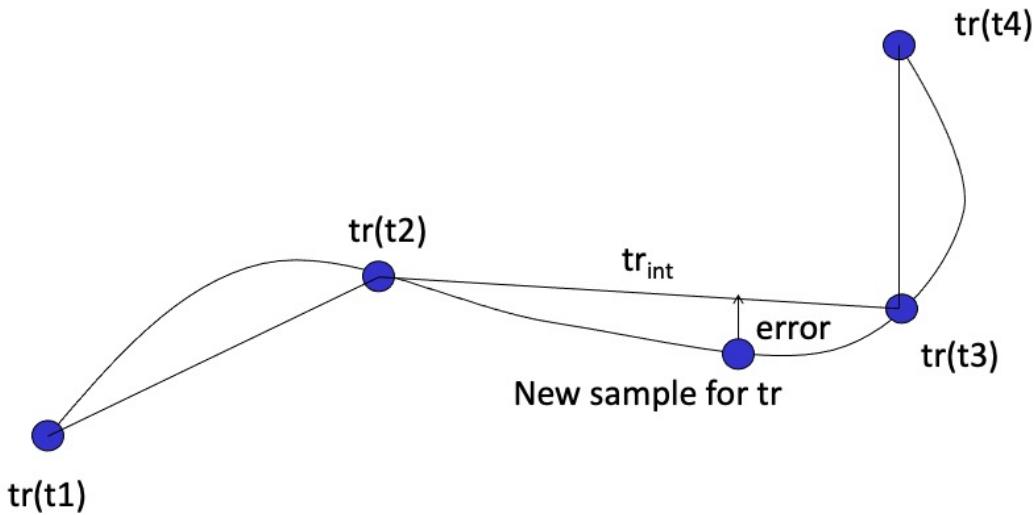
Example

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



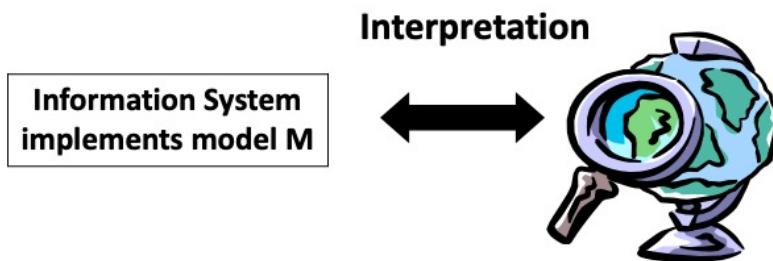
©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 47

This example shows a simple interpolation mechanism for trajectory data.

Representing Reality

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



Evaluating the model to characterize the quality of representation

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 48

Ultimately every model that we build is meant to represent some aspect of reality. Therefore also here we apply the concept of representation. Obviously, this relationship cannot be formally captured in mathematical formalism or through computation.

For which relationship the concept of representation is not applicable?

1. Between a model and the real world
2. Between two different models of the same real world aspect
3. Between a model and a data structure
4. Between a data structure and a physical model in a computing device
5. It is applicable to all of the above

4. DATA MANAGEMENT

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**.

In this context we have to be very careful about terminology, as data model is (sometimes) interchangeable used to designate two different things: (1) a data model used to represent a model within a computer system (this is the sense we will use in the following) and (2) a formalism respectively language to specify a whole class of data models. 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.

Database

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 realize information systems. The inverse is not necessarily true. Many information systems that use database do this without using a specific database system.

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

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 53

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 consider a wide range of platforms as potential database management systems today.

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

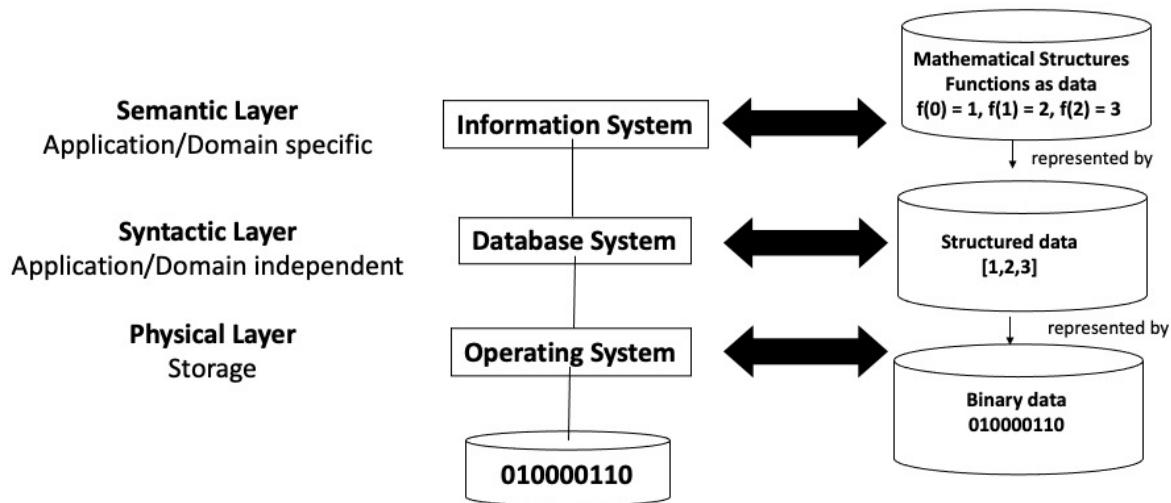
dt[dt['x'] < 116.5]				
tid		date	x	y
4	1	2008-02-02 16:06:08	116.47186	39.91248
5	1	2008-02-02 16:16:08	116.47217	39.92498
6	1	2008-02-02 16:26:08	116.47179	39.90718
7	1	2008-02-02 16:36:08	116.45617	39.90531
8	1	2008-02-02 17:00:24	116.47191	39.90577
...
579	1	2008-02-08 14:31:31	116.48503	39.91422
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

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 55

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

Refined View of an Information System



The term “data” has different meanings!

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 56

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 modeling. In order to implement the model the syntactic layer provides a data model in which the semantic model can be implemented 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.

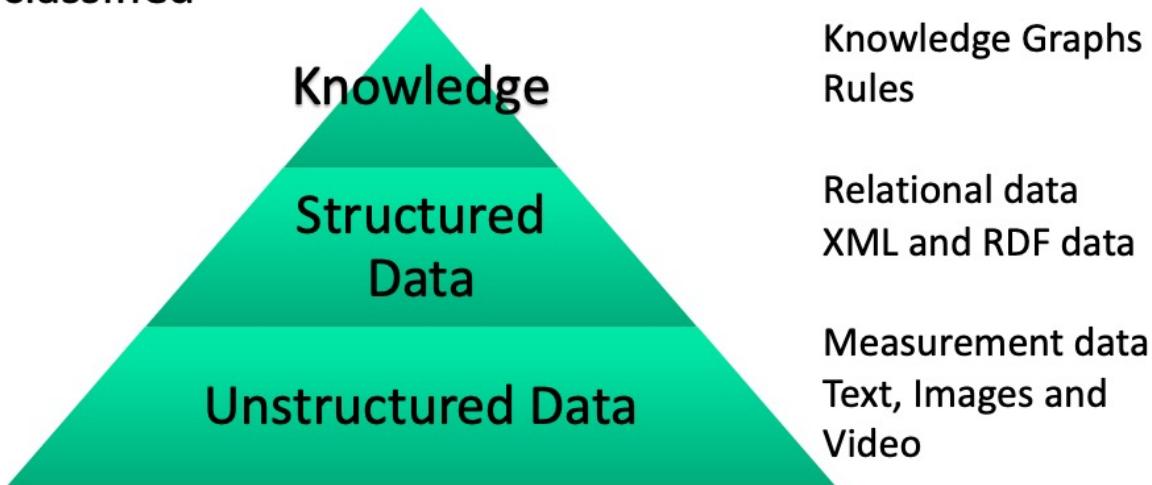
What is not specified in a data definition language?

1. The structure of a relational table
2. The query of a user
3. A constraint on a relational table
4. The definition of a view

5. INFORMATION MANAGEMENT

Levels of Abstraction

Depending on the “degree of abstraction” data is classified



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 59

There exist some conventions to distinguish data into different levels of abstractions. One can think of these levels of abstraction as levels of increasing interpretation of “raw data” that has been recorded or captured. These interpretations are usually obtained either by automated or human analysis of the data, or a combination of the two.

Note that though this classification appears similar to the one into conceptual, logical and physical layer, this is a different conceptualization. In each of the abstraction layers we will find the three modeling layers (as e.g. illustrate by our use of trajectory data as an example of modeling, where trajectory data would normally be considered as unstructured data).

Levels of Abstraction - Characteristics

Unstructured data

- Data captured from measurements and human input
- Structure given by static data types (e.g. time series)

Structured data

- Data is structured according to a schema
- Captures relationships in the data

Knowledge

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

The classification is not strict!

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 60

Unstructured data is usually referred to as data that originates more or less directly from some direct interactions with the real world environment, be it through measurements (sensors, images, videos), human inputs (natural language, text, query logs), machine input (system logs) or similar. 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 follows a static schema, that implements a interpretable model of some domain of interest. The most prominent example being the relational data model, where applications or users define tables with specific meaning. Such meaning 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, but even more abstract, and considered as being close to how humans represent their

knowledge. Knowledge is considered as a basis for decision making. Other interpretations of knowledge are the ability to reason with it. This implies deriving 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 practice in with a slightly different 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.

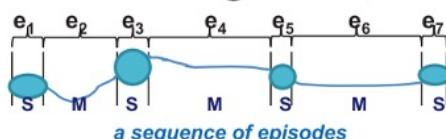
Example

Unstructured data: a GPS trace



Bob's and Alice's GPS trace

Structured data: Road segments, Places



Places that Bob and Alice visit

Knowledge: Concepts and Inferences



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

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 61

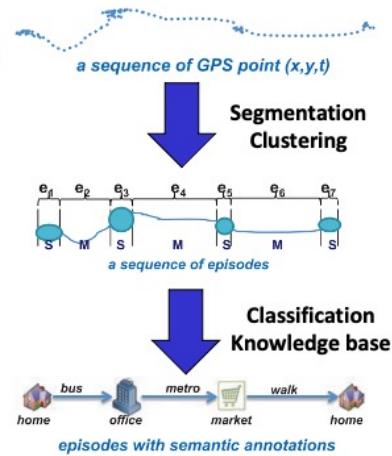
This example illustrates how such abstractions look like in practice. We can consider GPS traces as raw, unstructured data. Unstructured expresses here the fact that apart from the physical location we have no further understanding of the meaning of the data. By using automated analysis (e.g. segmentation 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 detected. 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 reality in a different way

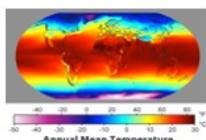


Thus, an important task in information systems is creating higher level abstractions from lower level data in order to support eventually decision making. This process is called Model Building. It uses numerous automated, semi-automated as well as manual methods, many of which we will introduce throughout the course. This activity carries also many different names, typically of the format data XXXXX.

Information Management

Tasks

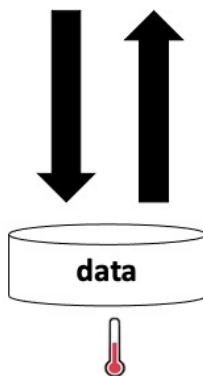
- Search
- Filtering
- Classification
- Prediction
- Recommendation
- Summarization
- Integration
- Question answering



model M

Model Usage: given a model, perform some task (downstream task)

Example:
Given road segments, find taxis that passed the road



Model Building: given data, find a model that matches the data

Example:
Given GPS traces, find places and road segments

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 63

Since the central role of an information system is to create a model of reality based on data, the key information management tasks are related to the interplay between data and models at the conceptual modelling level. We can identify two directions for this interplay: from models to data, and from data to models.

From models to data is commonly what we understand by **Model Usage**, including tasks such as retrieval or prediction. Given a model of reality we would like to obtain more data about specific aspects of reality. If we have a model of the temperature distribution in the world, we would like to retrieve the temperature at a specific location or a global average temperature. For Web search we would use a model of how search terms provided by user to a Web search engine relate to documents considered as being relevant by the user, to retrieve the results of a user query.

Model Building

Creating new models that “interpret” the data in novel ways.

Two flavors

- Model building = creating representations in new domains
- Model building = creating new functions/relations in the same domain

Going from data to models is what we commonly understand as **Model Building**, also called data mining, data science, machine learning, depending on the context etc. Often we find big data collections for which we do not have a proper or only incomplete interpretation by a model. For example, we might have temperature measurements at given points, but do not understand the correlations among those measurements or the values at locations without measurements. If we have large document collections, we do not understand which are the topics that are covered by those documents. Data mining deals with the problem of providing algorithms that reveal hidden structures in data in order to create new models.

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

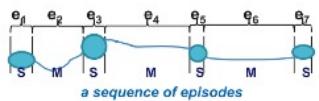


Methods: Data mining, Machine Learning, Rules, Statistics

This example 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 (as represented in DB)

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

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

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

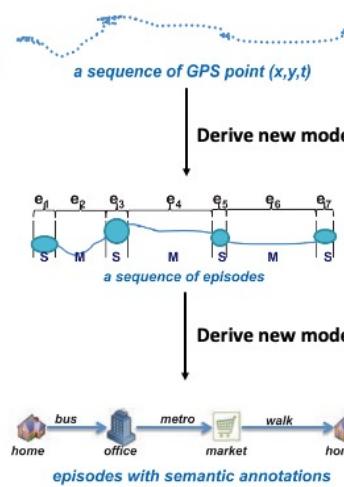
©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 66

Every new model, generates new data, either by transforming the original data into data corresponding to the new model, or generating new data through model functions.

New Models generate New Data

Model (Information System)



Obtain data from measurement

Data corresponding to the model

Derive new model from data

Data corresponding to the model



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Data (Database System)

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

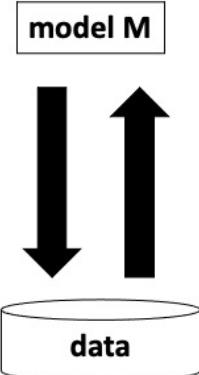
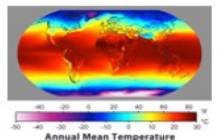
place	x	y	start	end
p111	12	13	2:00	2:10
p112	13	15	2:25	2:30

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

Introduction - 67

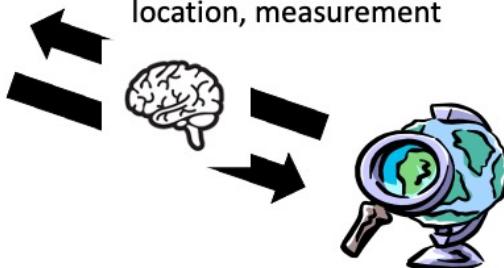
In more detail, the process of model building consists of deriving new models (new functions, elements, axioms) from data, which in turn by model usage can generate new data, from which more models can be built.

Interpretation of Models



Conceptual modeling: analyze the real world and specify a model

Example: define concepts temperature, location, measurement



Evaluation: given a model, evaluate it against reality

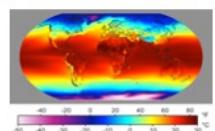
Example: compare predicted temperature to measurements

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

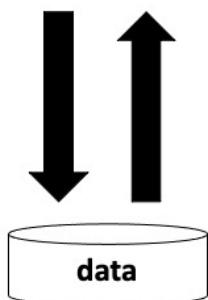
Introduction - 68

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 verifying the interpretation relationships. This is possible only in indirect ways. Again we have two directions involved in the relationship among models and the real world. 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 temperature modelling we would identify the concepts temperature, location and measurements as key concepts and represent them in a model. More generally conceptually modelling is widely used in the development of business information systems, where business analysts perform requirement analyses in order to determine what are the organizational structures and the processes within a business, and what are the processing needs. 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 temperature this could be achieved by comparing temperature values predicted by the model for certain locations with the true values. 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



model M



Building systems where the data interacts with the real world

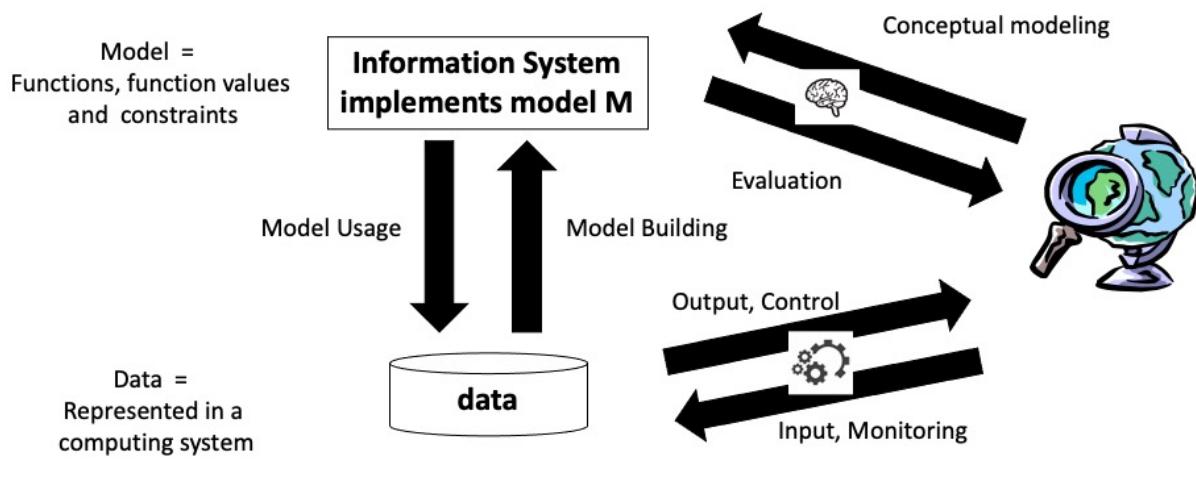
Output: data visualization, control
Example: control an autonomous vehicle



Input: users, sensors

Regarding the interaction between an information system and their real world context, we can consider the interaction with human users as well as direct interaction with the real world environment via sensors and control devices. With respect to data output for human users the visualization of large datasets is becoming increasingly important. With respect to human input a recent development is the use of input from large communities, so-called crowd-sourcing. Direct interactions with the real-world is what is called today the Internet of Things, where computers receive data through sensors and control directly devices.

Information Management



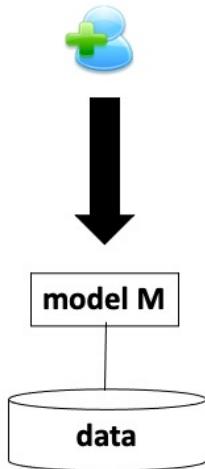
©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 70

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

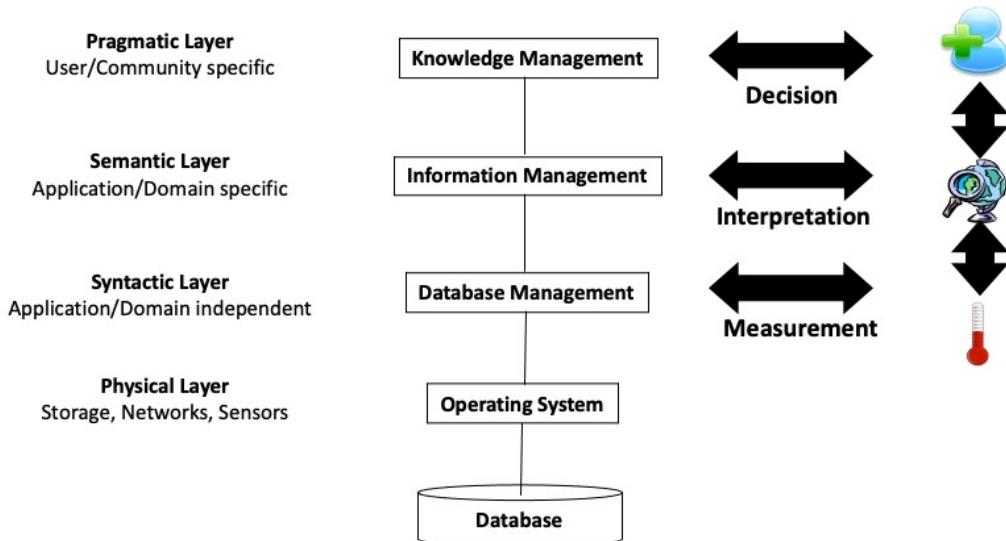
- Importance of decision
- Quality of decision

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 useful information. 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. Traditionally utility of information has been considered implicitly in the design and implementation. But as the awareness of the importance of information is growing, more and more we also see that utility of information, and related factors, such as quality are treated and managed explicitly. For example, data quality in databases is evaluated and monitored nowadays in many cases explicitly. Systems for rating and recommendation, e.g. in social networks, are another example where quality of information is handled explicitly.

Refined View of an Information System



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 72

Considering the information management tasks mentioned before we can further 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. There exists no well-established notion for this layer. It can be considered as part of knowledge management, but this concept is much wider, and it is implicitly found in many areas such as evaluation of information systems, data quality management, agent systems, social networks etc.

Grouping Facebook users according to their interest by analyzing the content of their posts is ...

1. a retrieval task
2. a data mining task
3. an evaluation task
4. a monitoring task

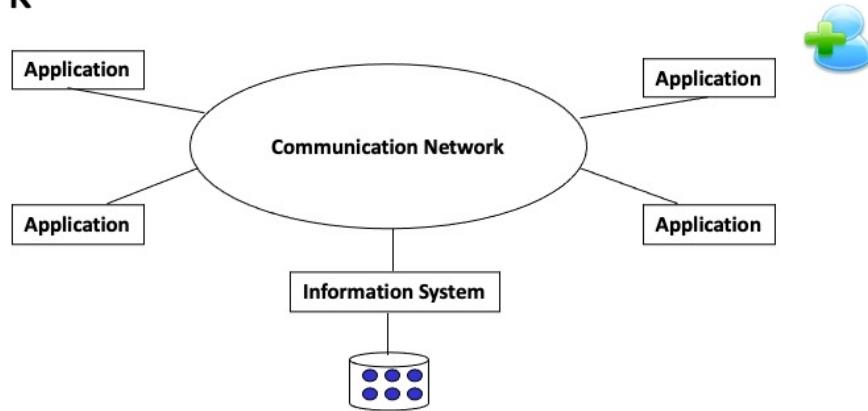
6. DISTRIBUTED INFORMATION SYSTEMS

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 74

Centralized Information System

Centralized Information System on Computer Network



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

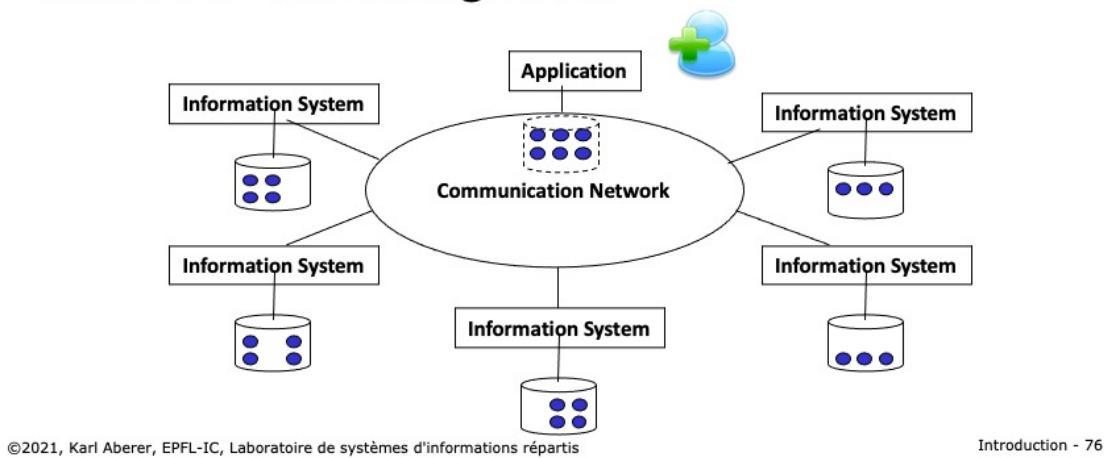
Introduction - 75

Except in the very early days, information systems had always been used in computer networks. This does not imply any significant additional problems beyond those we have discussed already, as long as the information system is centralized, i.e. running on one physical node under a single authority. The network just enables the interaction of a user with 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



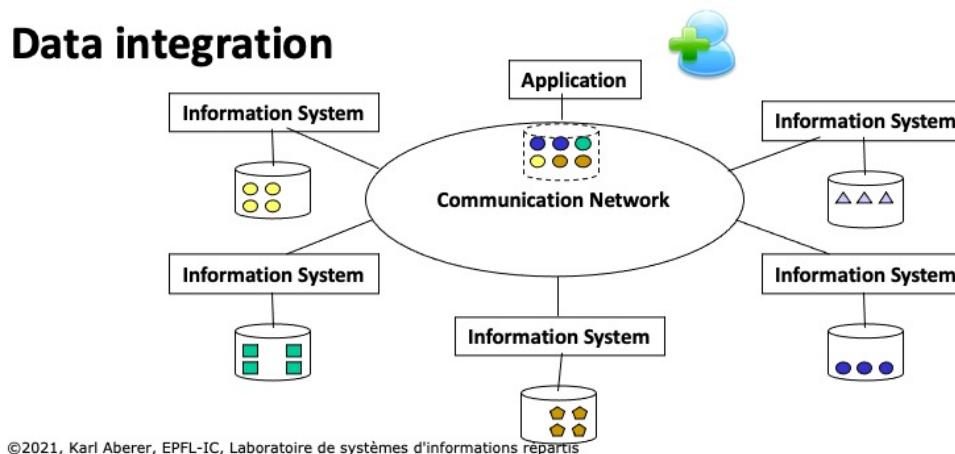
There exist however many reasons not to store all data in a single node of the network. Some of these reasons are related to the optimized use of available resources. We might want to move data in the network close to the node where it is accessed, we might want to take advantage of parallel processing 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**.

Semantic Heterogeneity – Logical Distribution

Use of different data models

- Independently developed information systems
- Different models for related concepts

Data integration

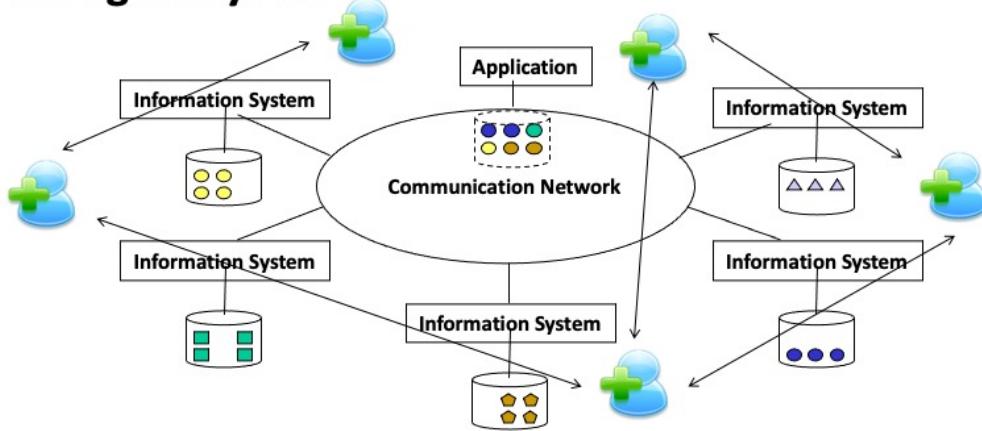


Having a homogeneous view of a distributed information systems might not always be possible. If we want to access information in systems that have been developed by different entities, or if we want to integrate information from different information systems that have been independently developed, we 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



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 78

Finally in heterogeneous information systems we have to 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.

Creating a web portal for comparing product prices is (primarily) a problem of ...

1. Distributed data management
2. Heterogeneous data integration
3. Collaboration among autonomous systems

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.

Key Issues in Heterogeneity

Data overload

- more data, more useful information ?

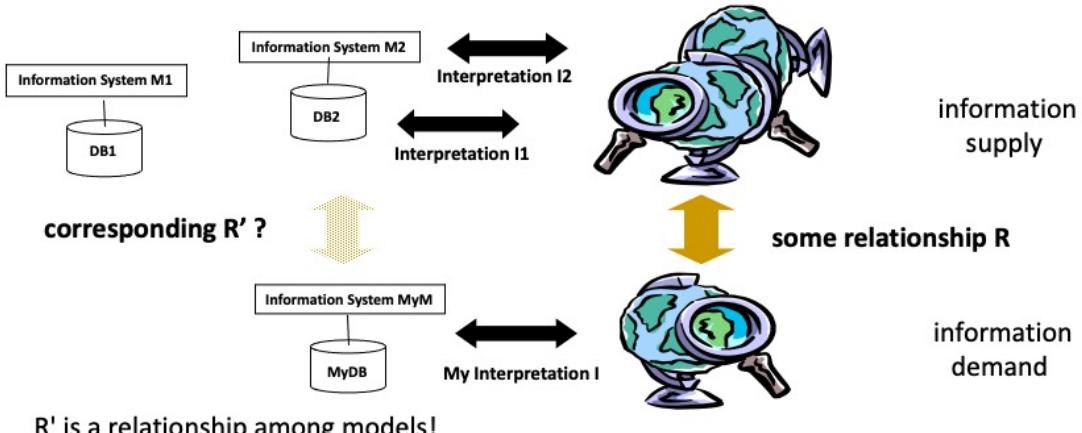
Information starvation

- problem: *data supply* does not match *data demand*
- models used by data provider are different from models used by data consumer

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. If the data cannot be properly interpreted and used by the data consumer, thus if the data supply does not match the data demand, the utility of the data remains limited. More data does not imply more information! The data needs to be made available according to a model that is useful for the data consumer. And this model might be largely different from the model used by the data producer. In fact, the data consumer might even not be able to understand that model. This is what is called information starvation.

Distributed Information Management

More data! ... More models!? ... More useful information?



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 82

This figure illustrates the problem of information starvation: there exist many information systems supplying data, but each having their own view on the world, which does not necessarily match the needs or understanding of a specific consumer. Every information system is interpreting its model differently with respect to the real world and relating to different views on the real world. Though there exists some relationships among all these views on the real world (let's denote it as R), and it surely implies some relationship R' among the different models used in the different information systems, the consumers of the information cannot easily understand the relationship R , and thus can also not easily relate their models to the models of others via the relationship R' . From the viewpoint of the data providers, introducing R' introduces a new interpretation of their data with respect to the model used by the data consumer.

The Problem

Semantic heterogeneity

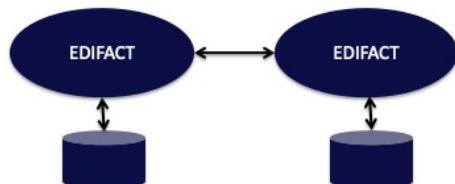
- The same real world aspect can be represented differently
- Requires agreement on the meaning of shared data
- Relating different models (and thus different representations and their interpretations) requires often human intervention
- human attention is a scarce resource !

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

Three Approaches

Standardization

- Common model



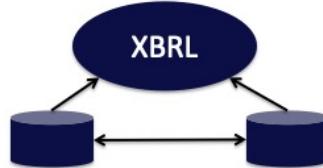
Mapping

- Mapping among models



Ontologies

- Common reference model



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

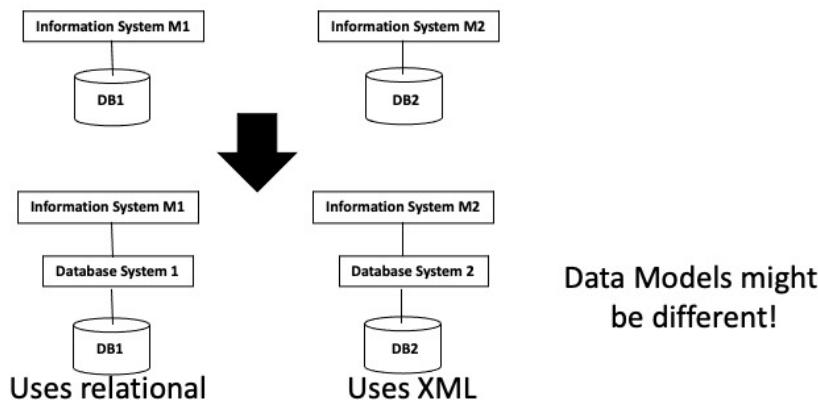
Introduction - 84

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 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, 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. A third 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.

More Problems?

Syntactic heterogeneity

- The same model and data can be represented using different data models



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

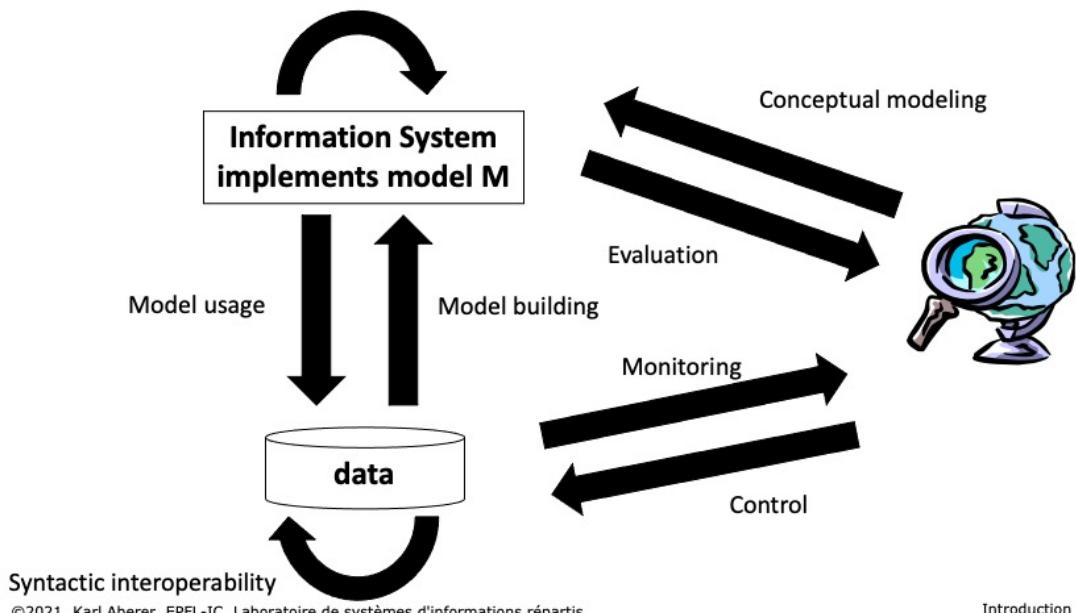
Introduction - 85

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)

Information Management Tasks

Semantic interoperability



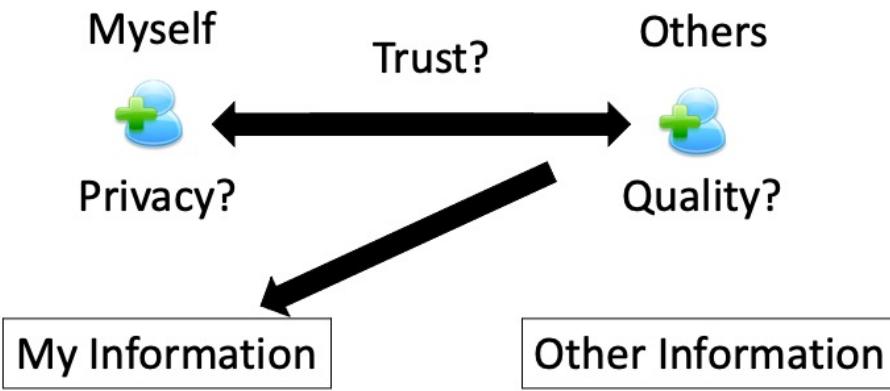
©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 86

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

Key Issues in Autonomy

The Users Problem

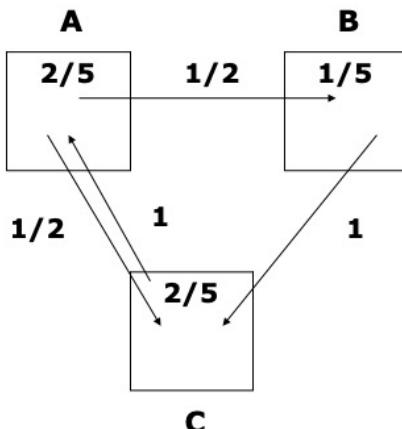


Revealing quality information increases trust,
but lowers privacy

In an setting where different autonomous users exchange information issues 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 incite us to certain behaviours. When providing information to another user a different problem needs to be considered. Can we trust the other user to use the information correctly, or will he use it in ways that could be damaging to us. This is the privacy problem, and it is receiving in the information society huge attention. The two problems of information trust and privacy are also linked to each other. The more quality information we reveal the more trust we may expect, but the more we also put out privacy in danger.

Evaluating Quality of Information

Recommendations (e.g. Google PageRank)



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 88

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.



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 89

One 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

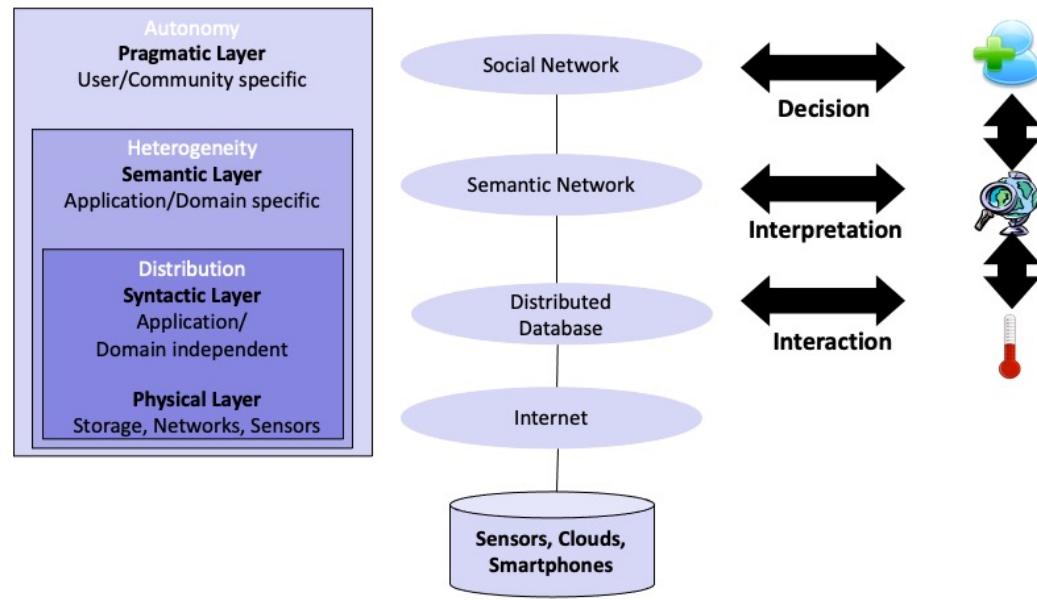
Example: location privacy – obfuscation methods

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

	1	2	3	4	5	6	7	8	9
1									
2									
3							●		
4									
5									
6									
7									

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 the 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



©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 91

Trust is ...

- 1. a quality of information**
- 2. a quality of a user**
- 3. a quality of the relationship among user and information**
- 4. a quality of the relationship among users**

7. ABOUT THE LECTURE

©2021, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis

Introduction - 93

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)

- Relate different texts among each other

Information extraction: extract facts from text (part 3)

- Relate text to structured models

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)
 - Requires data integration as preprocessing
- Data integration: determine relationships among models (part 3)
 - Uses frequently data mining techniques

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

- Natural language as mechanism to share information

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.

Exercise

Big Data is often characterized by the four concepts of Volume, Velocity, Variety and Veracity

- 1. Inform yourself what is meant by those concepts**
- 2. Identify from this lecture four problems / methods that are related to each of those four concepts**

As data is a central notion to the lecture, and Big Data is a common terminology used to characterize many of the aspects exposed in this introduction to distributed information systems, attempt to relate the different “V’s” of Big Data to the concepts discussed in this overview.