Distributed Information Systems
An Overview

Overview

1. What is an Information System?
2. Data Modelling
   2.1 Models
   2.2 Data
   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?

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

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 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.

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.

Ask ChatGPT

An information system (IS) is a collection of hardware, software, data, people, and procedures that work together to facilitate the gathering, processing, storing, and dissemination of information within an organization or a specific context. The primary purpose of an information system is to support decision-making, coordination, control, analysis, and visualization of information to help individuals, groups, and organizations achieve their goals effectively and efficiently.

Information systems can vary widely in scope and purpose, ranging from simple systems used for personal tasks like managing contacts or calendars to complex enterprise-wide systems used by large organizations for functions like customer relationship management, supply chain management, and financial reporting.

Key components of an information system include:

1. **Hardware:** The physical devices and equipment that process and store data, such as computers, servers, storage devices, networking equipment, and peripherals.
2. **Software:** The programs and applications that enable users to interact with and manipulate data. This includes operating systems, databases, communication software, and various applications tailored to specific tasks.
3. **Data:** Raw facts, figures, and symbols that are collected, processed, and stored by the system. Data can come in various forms, such as text, numbers, images, and multimedia.
4. **People:** The individuals who use, manage, develop, and maintain the information system. This includes end-users, IT professionals, managers, and other stakeholders who contribute to the system's functionality and success.
5. **Procedures:** The formalized methods, rules, and guidelines that dictate how the system is used, maintained, and controlled. Procedures ensure consistency, security, and proper functioning of the system.

Information systems can be categorized based on their scope and focus:

- **Personal Information Systems:** Used by individuals for personal tasks such as email, document management, and time tracking.
- **Enterprise Information Systems:** Used by organizations to support various business functions like sales, marketing, finance, and human resources. Examples include Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP) systems.
- **Decision Support Systems:** Provide tools and information to assist in decision-making by analyzing data and generating insights.
- **Knowledge Management Systems:** Focus on capturing, storing, and disseminating an organization's collective knowledge and expertise.
- **Geographic Information Systems (GIS):** Specialized systems for capturing, managing, analyzing, and presenting geographic or spatial data.
- **Transaction Processing Systems (TPS):** Handle routine, day-to-day transactions within an organization, such as order processing and inventory management.

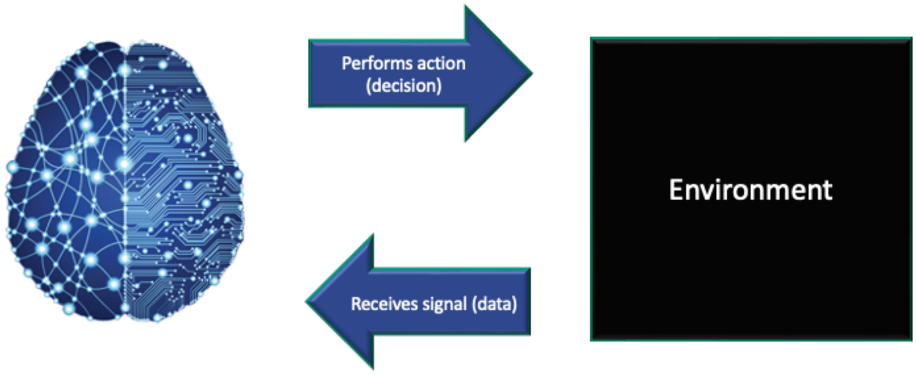
Overall, information systems play a crucial role in modern businesses and society by enabling efficient data management, improved communication, streamlined processes, and better decision-making.

(pretty similar to the WikiPedia article)

ChatGPT relies in its answer largely on the article found in WikiPedia.

A Systems Perspective

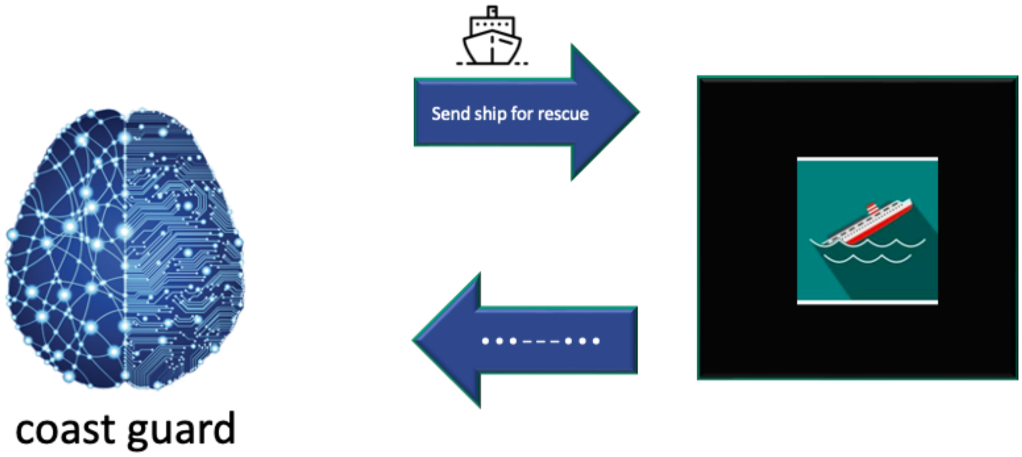
Information Processing



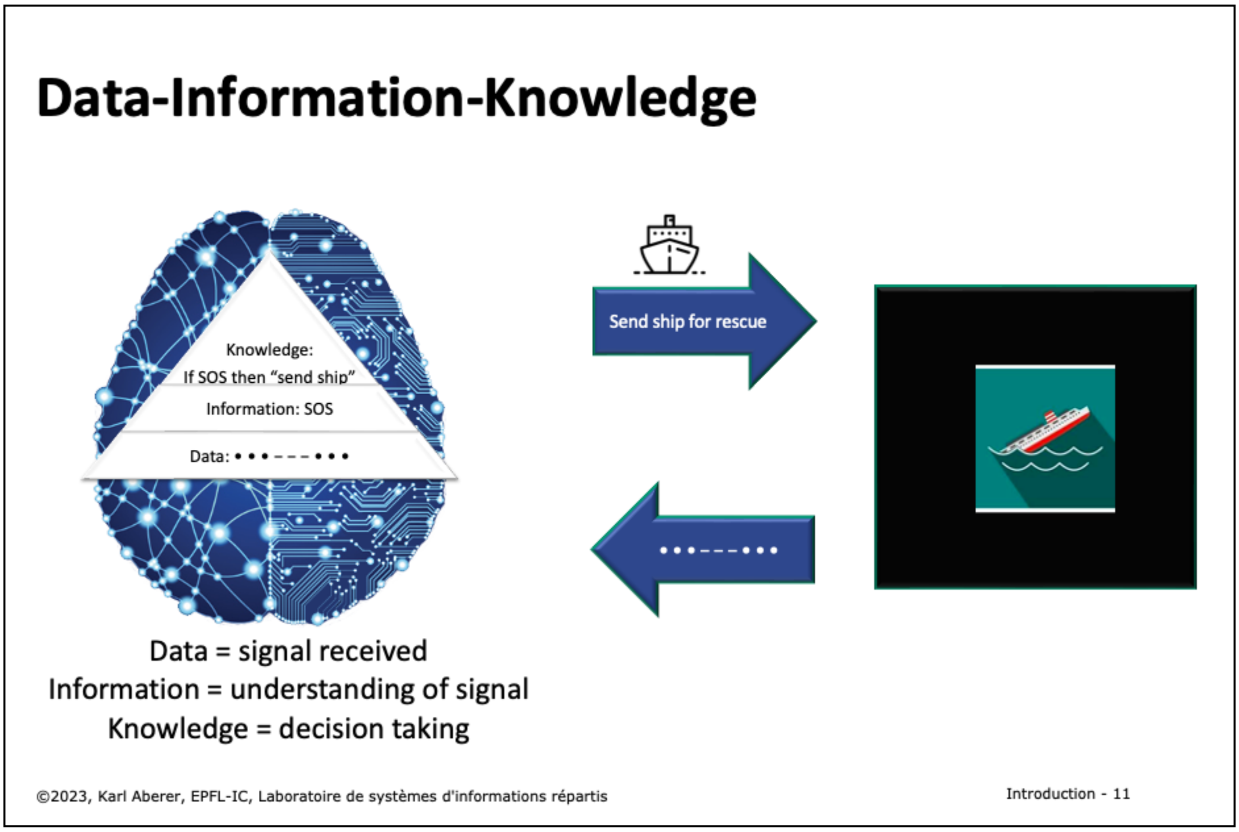
This model applies to all types of systems: organisms, species, humans, organizations, societies, etc.

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.

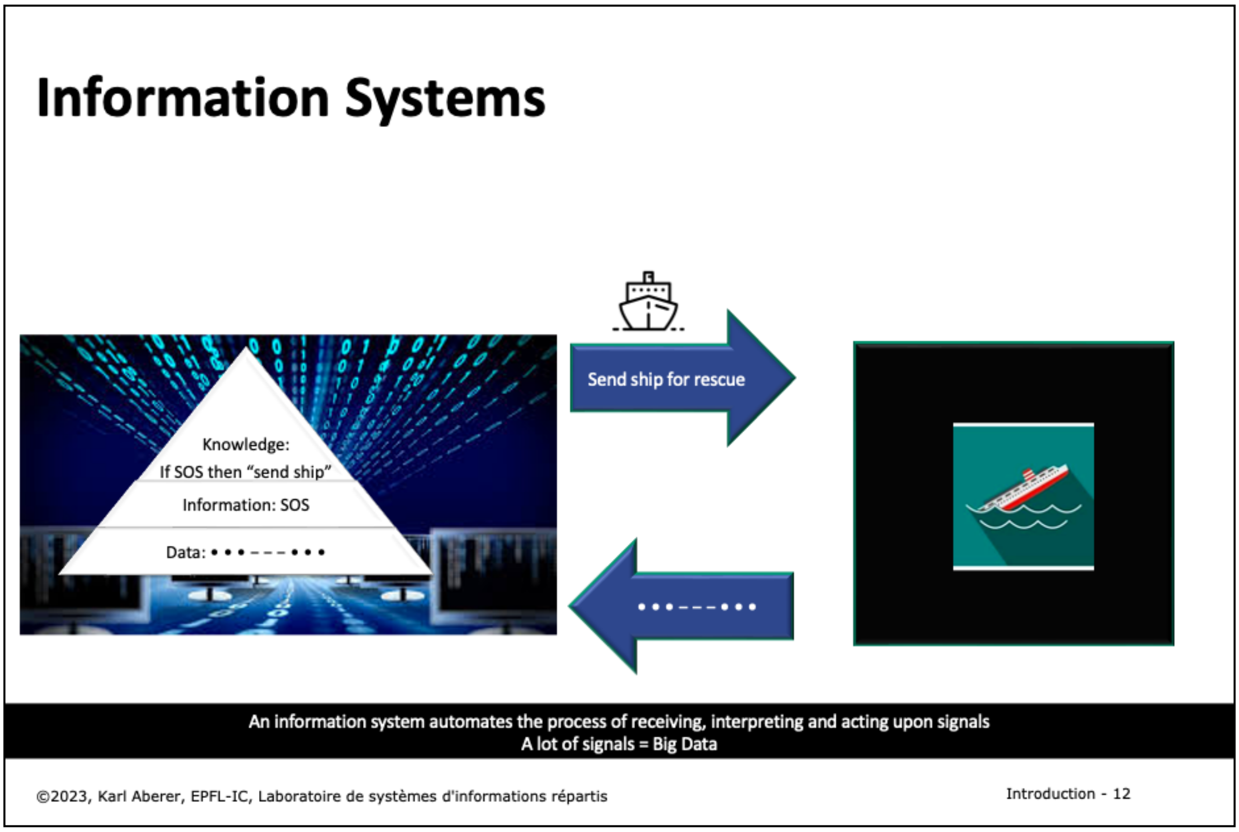


Looking in more detail on what is happening, we can identify different aspects of this process.

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.

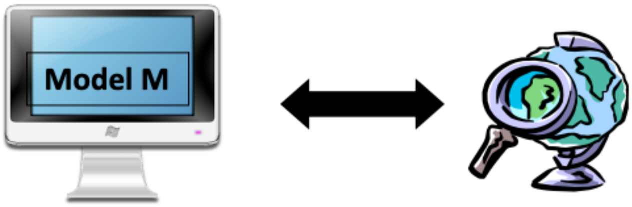


The process described in the example, applies to any entity that is performing information processing in the context of an environment, e.g., an organism, an organization, or a computing system. Performing information processing in the context of computational systems is what we will be studying in this 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 *by using a representation of a **model** of its **real-world environment***
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).

2. DATA MODELING

2.1 What is a model?

Mathematical representation of a real-world phenomenon

Example: travel relationship

Relation : $travel \subseteq Person\ x\ Location\ x\ Location$

Axiom: $travel(p, l1, l2) \land travel(p, l2, l3) \rightarrow travel(p, l1, l3)$

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.

These 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.

Data

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)

Representing a single trajectory tr as data, a set of samples: $tr(t_0) = s_0, tr(t_1) = s_1, \ldots$

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 using a function, as it is frequently used to describe, for example, vehicle trajectories recorded using a GPS.

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.

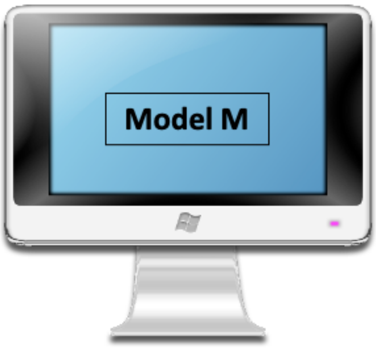
In our trajectory example, we can use both methods. We could describe trajectory by (measured) samples, or by a function generating a trajectory.

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

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**.

2.2 Data

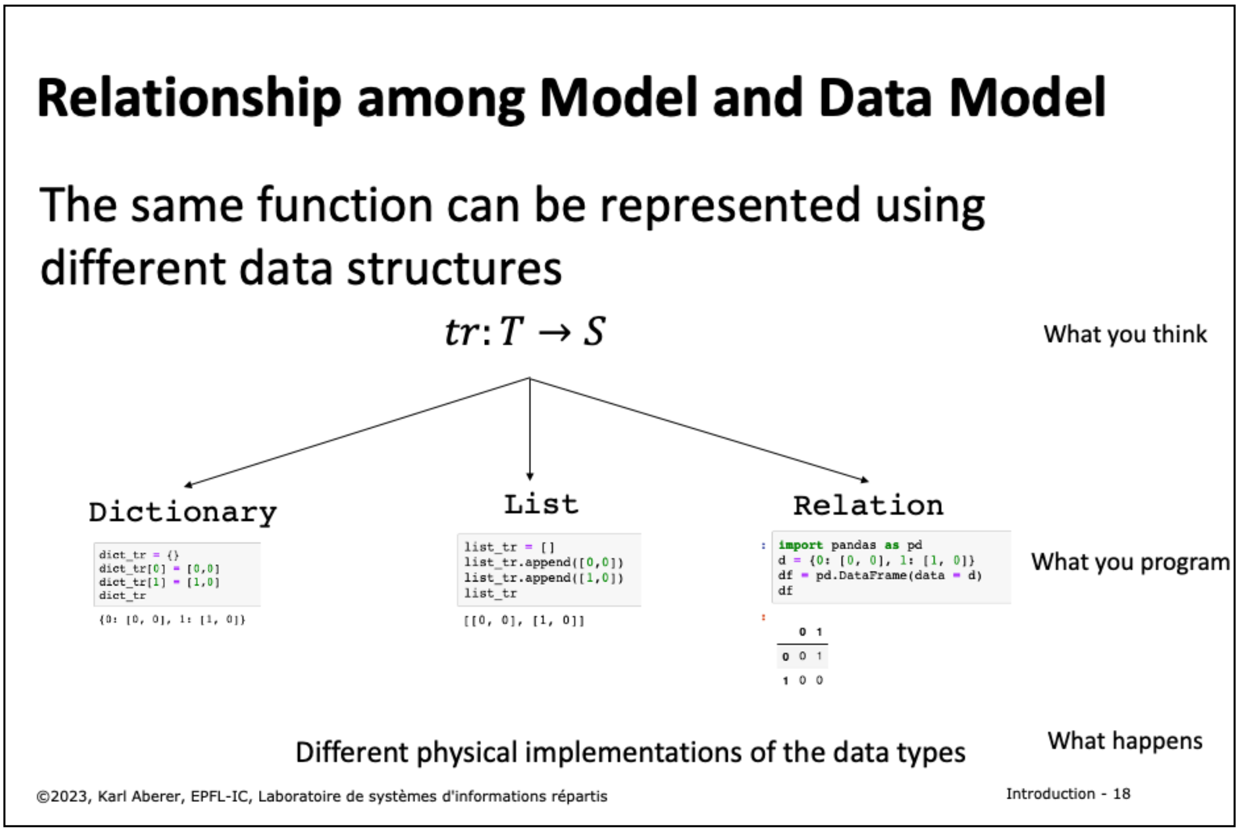
How is a model represented in a computer system?



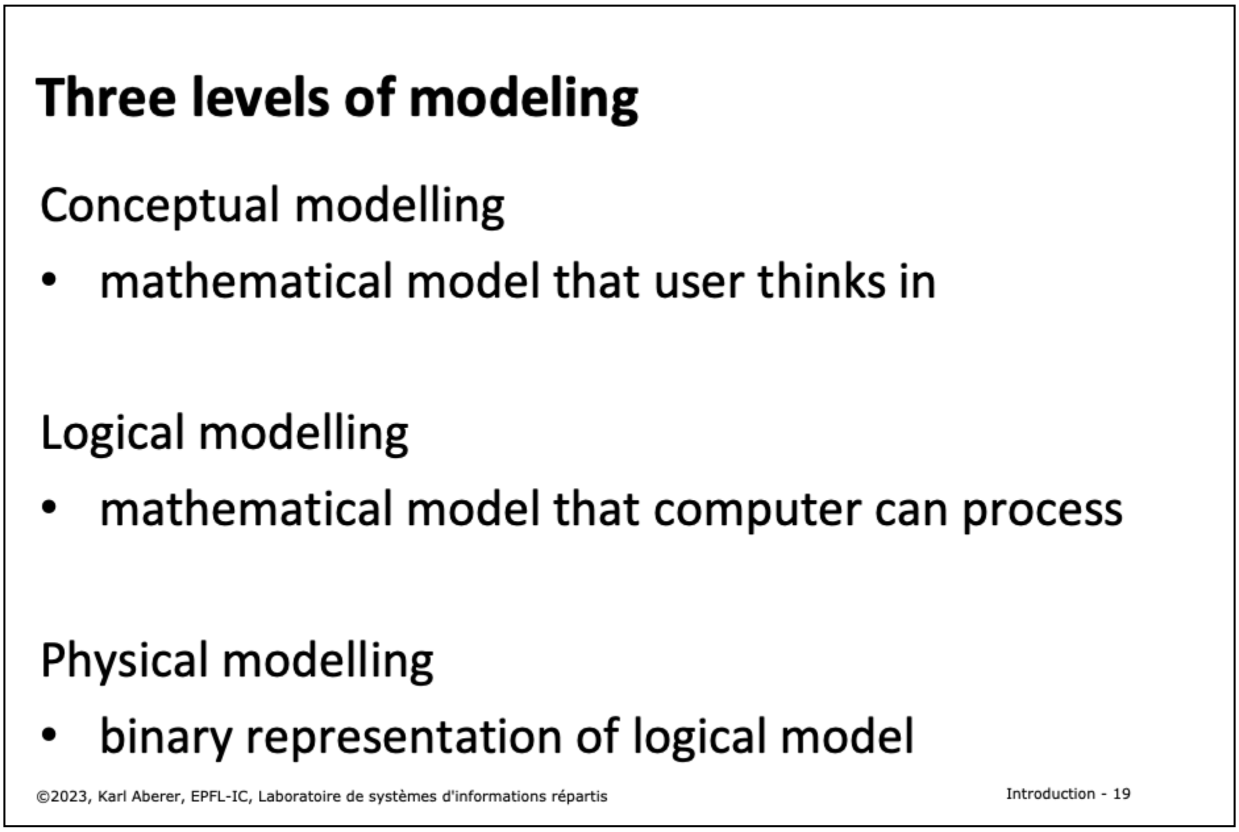
A mathematical model M is represented using a **data structure D**.

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.



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.



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

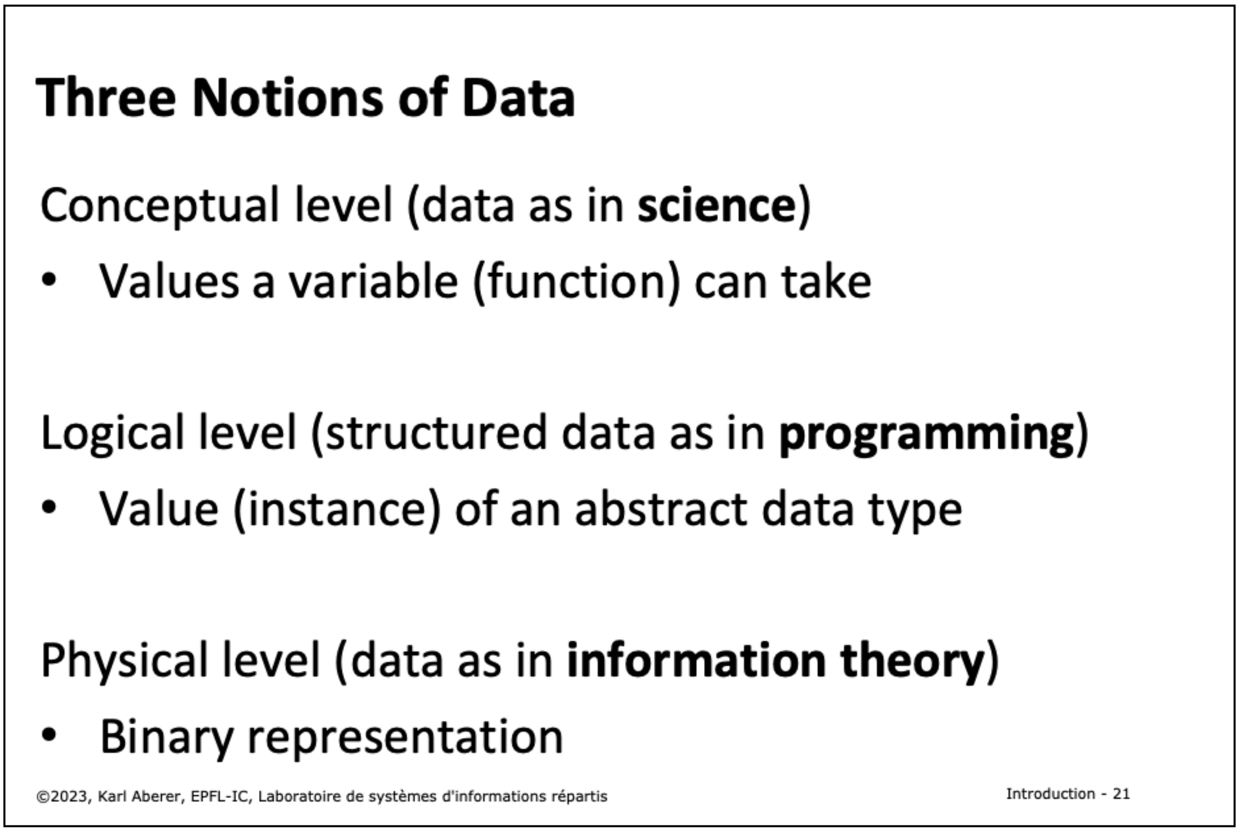
These three levels have been identified as a formal standard in the ANSI-SPARC architecture for abstract design standard for database systems in 1975.

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.

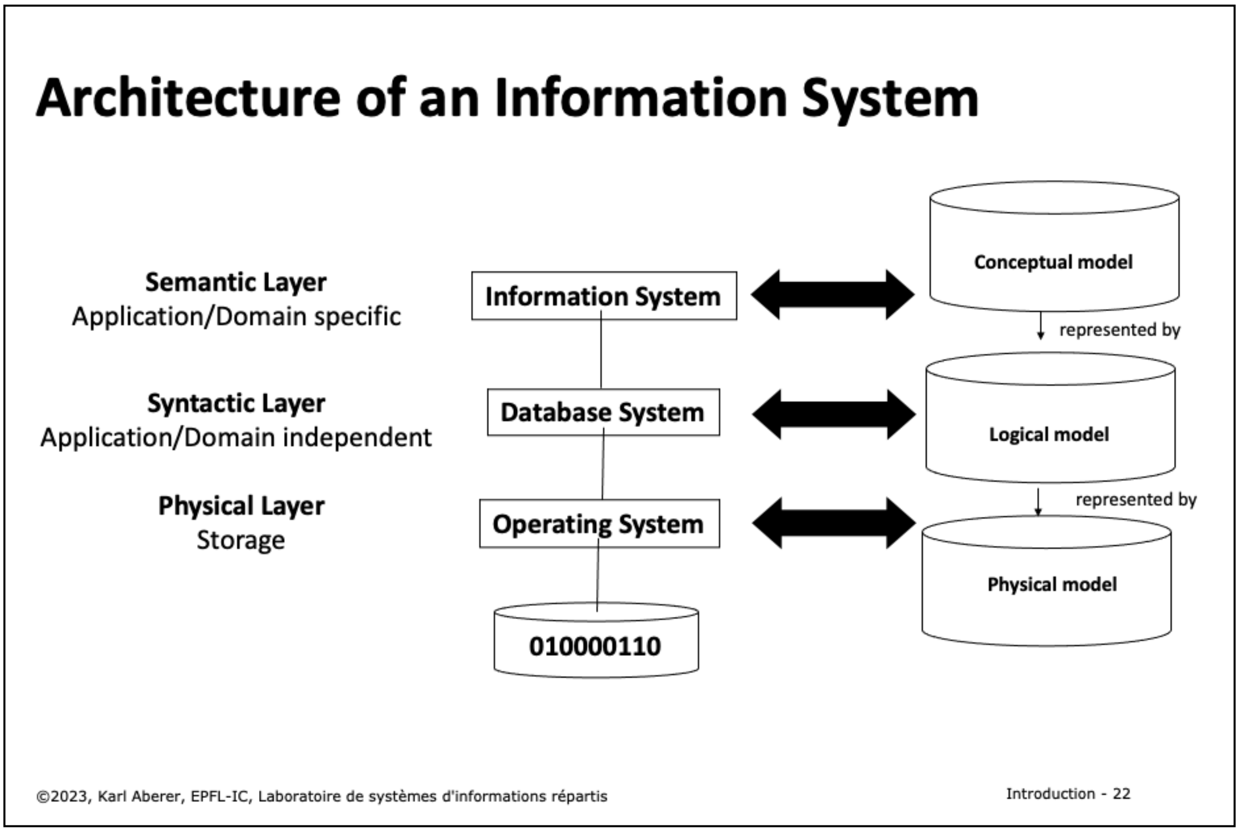


We can also relate the concept of "data" to these three modeling levels. This reveals the interesting fact that we are continuously using the term data with varying meaning (without necessarily being aware of that).

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)



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.

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** $H: X \to Y$ 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.

Binary Representation

Data structure: `list(int)`

Example: `1,8,3,` …

Binary representation:

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

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

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

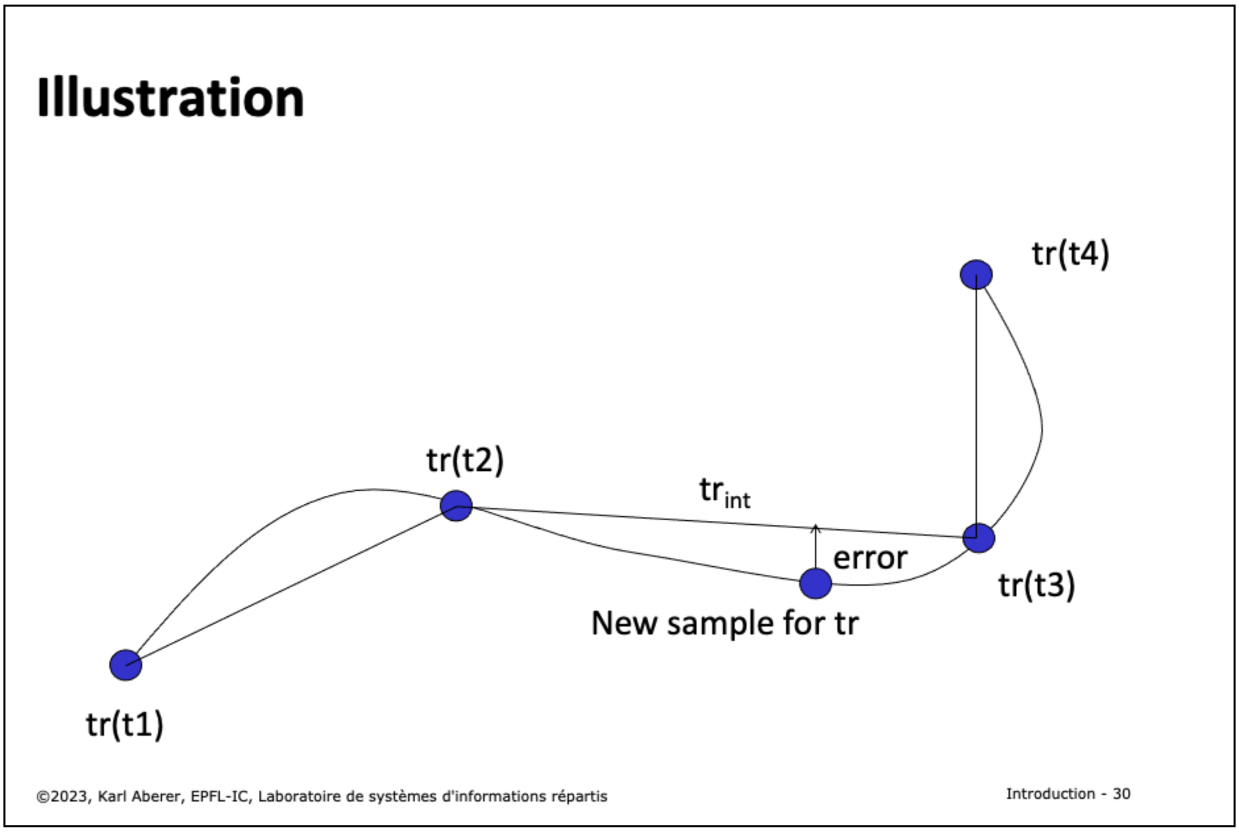
Representing an empirical mathematical model, by another mathematical model

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

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

- 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.



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**

Representation

Information System implements model M ⟷ 

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

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.

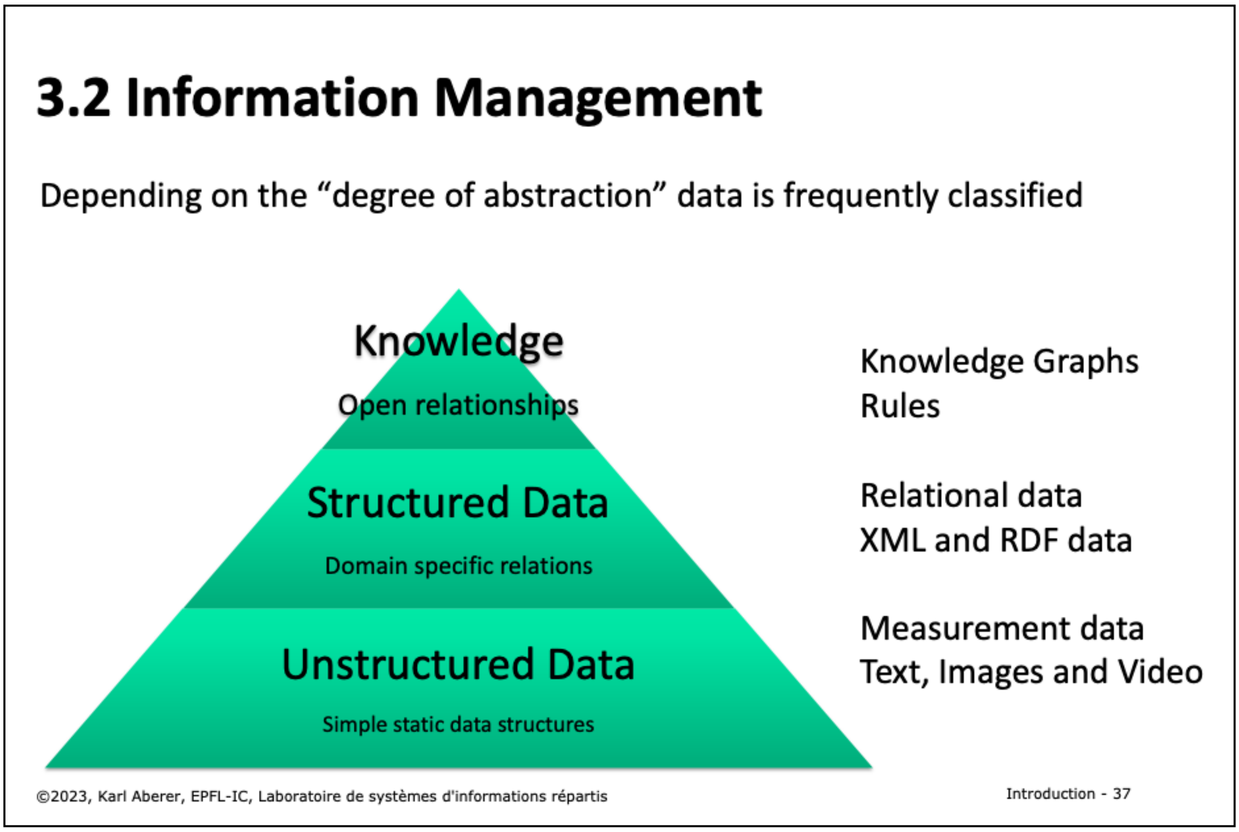
Example: DataFrames

	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.



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.

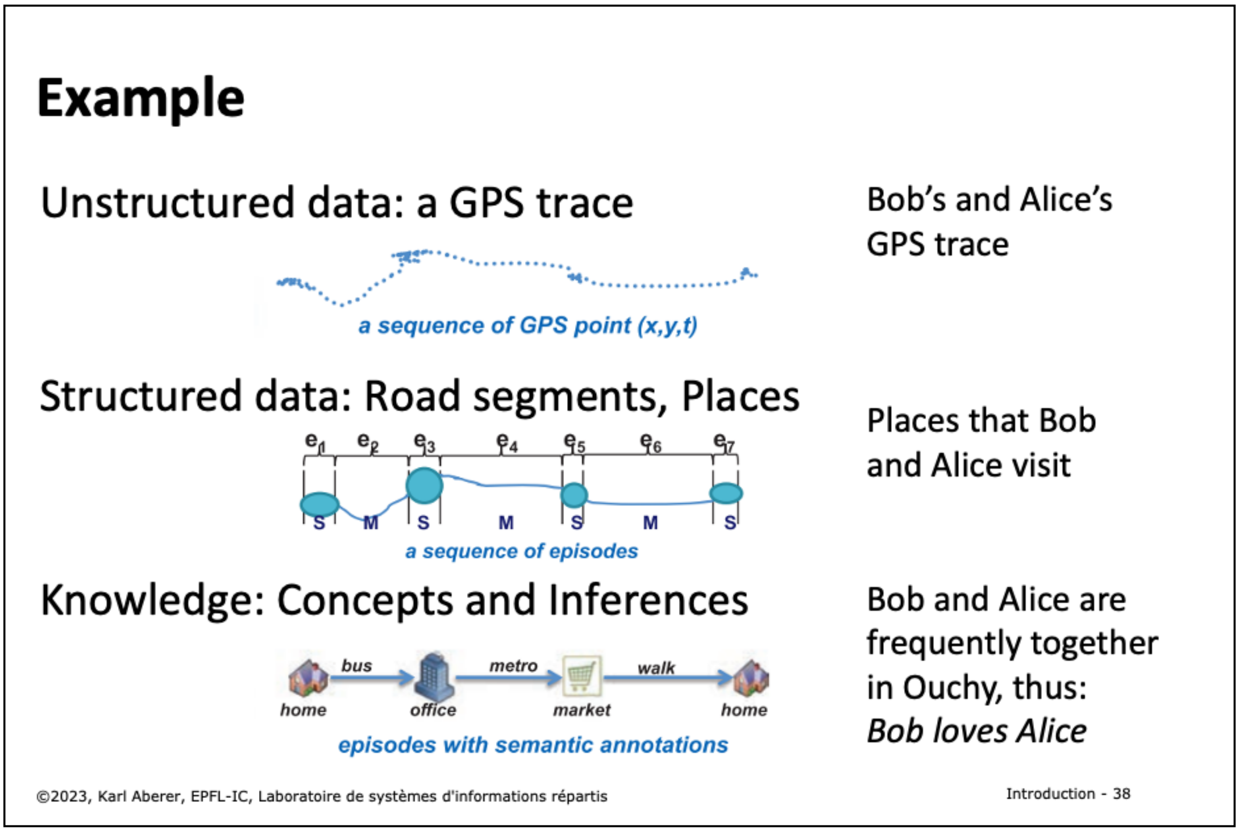
This convention for classifying different types of data refers a priori to data in the sense of conceptual data. For each of the three levels, however, specific logical data models (e.g. graphs, relations) are preferably used.

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.



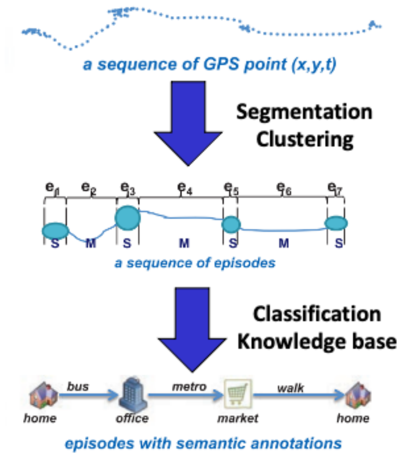
This examples illustrates how such abstractions can look like in practice. We can consider GPS traces as raw, unstructured data. Unstructured expresses here the fact that apart from the physical location we have no further understanding of the meaning of the data. By using automated analysis (e.g. segmentation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

New Models generate New Data

Model

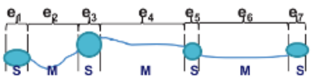
Data

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

Obtain data from measurement

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

Derive new model from data

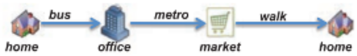


a sequence of episodes

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



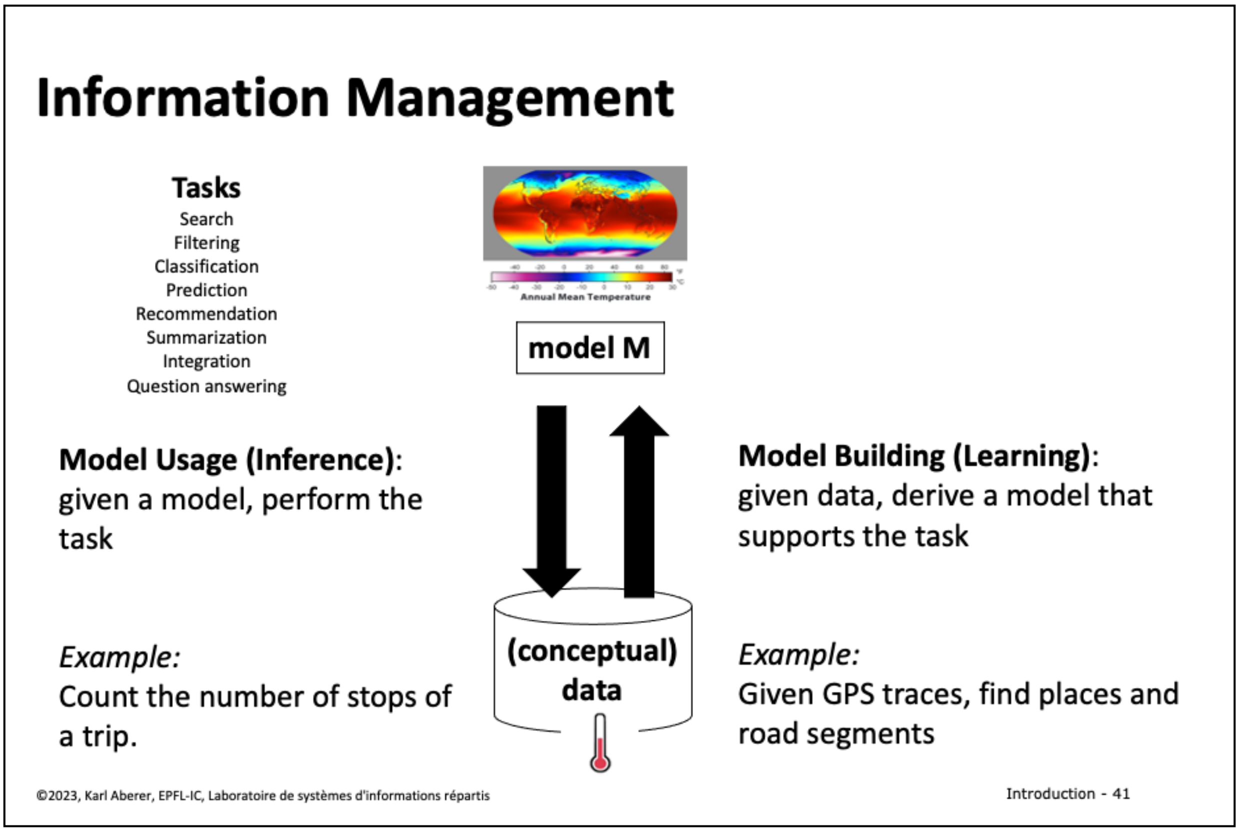
episodes with semantic annotations

Data corresponding to the model

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

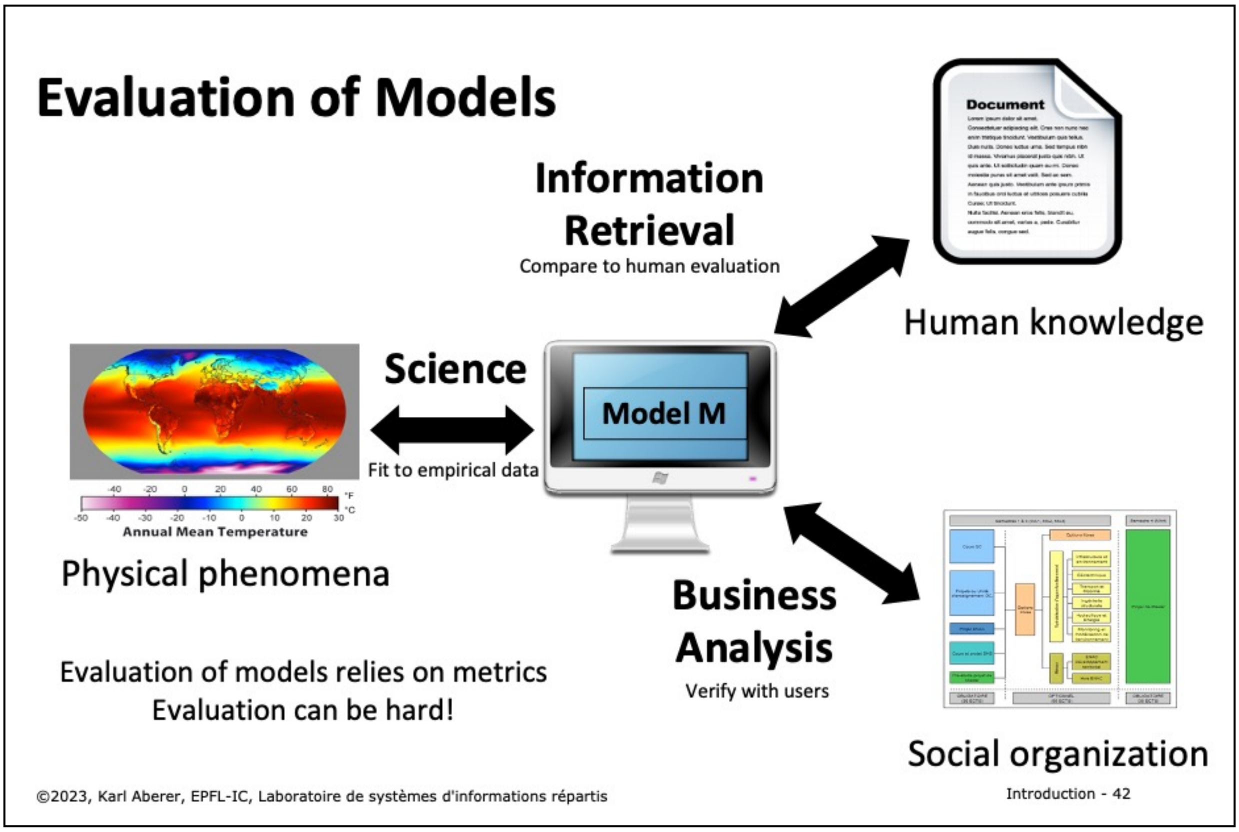
Introduction - 40

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.



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.



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 as they are not falsified (as we know since Popper we can never proof that a model is correct!).

Quality vs. Cost Tradeoff

Computational cost of deriving models from data can be high

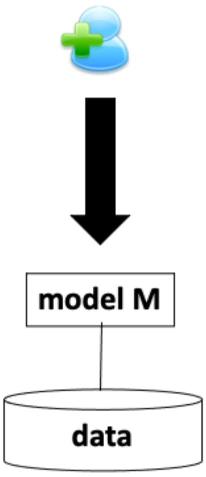
- Generally, better models have higher costs
- A trade-off

Example:

searching documents with keywords vs.
searching using a language model

Whereas information management is concerned with the issue of creating and using the best possible models of real world phenomena, in practice this cannot be done without ignoring the capabilities and costs of the underlying computing system. As a general rule, better models, i.e. models that more accurately and comprehensivly represent the world come at a higher cost. Therefore the gain in quality of presentation has to be outweighed with the computational, and hence energy cost incurred.

Utility



Users need information system to take **decisions**

Utility of information linked to the value achieved

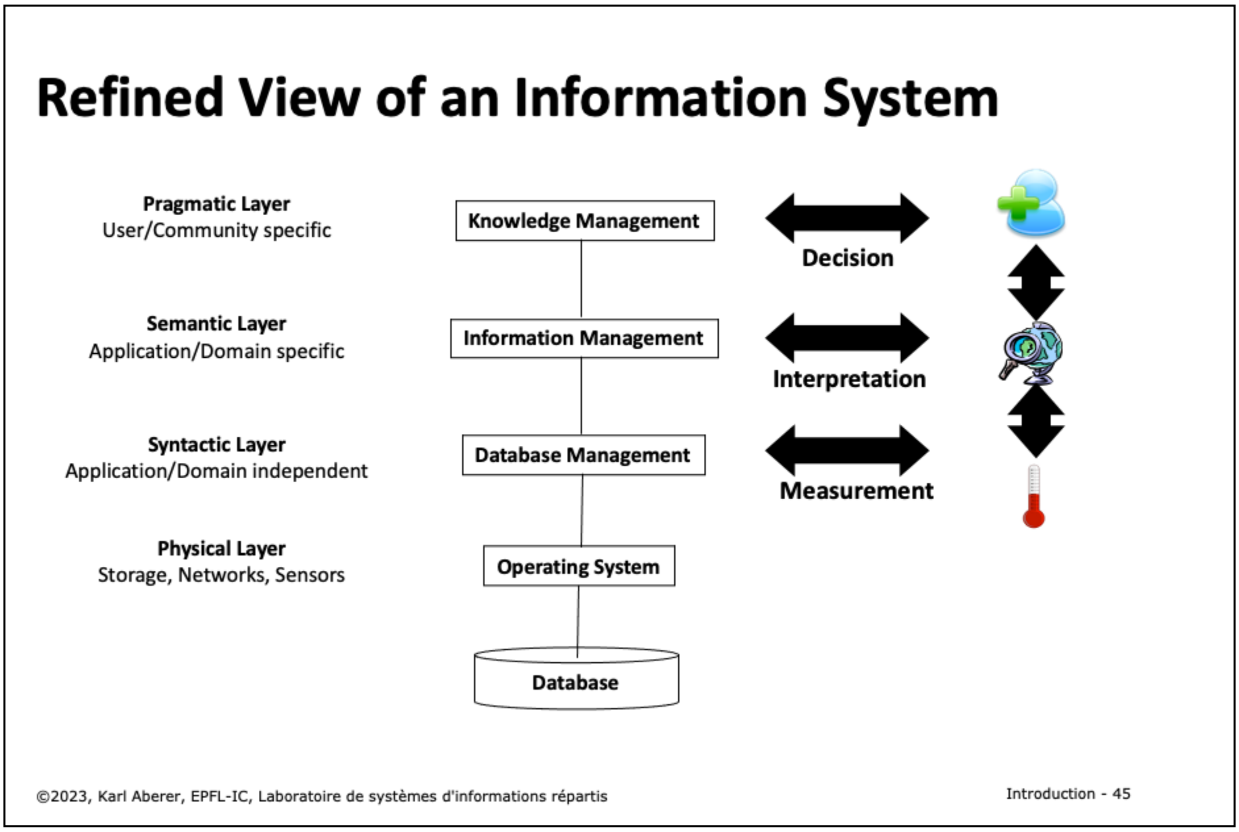
Value depends on

- 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.



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.

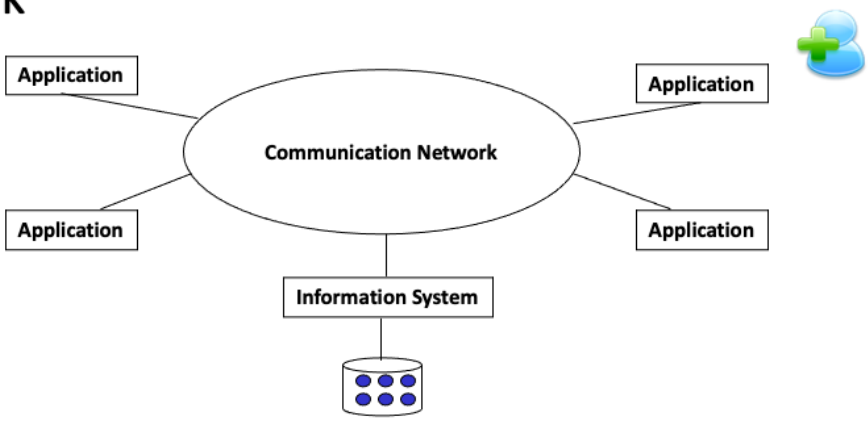
3.3 Distributed Information Management

Distribution can occur at different levels

- Network
- Data
- Models
- Control

Centralized Information System

Centralized Information System on Computer Network

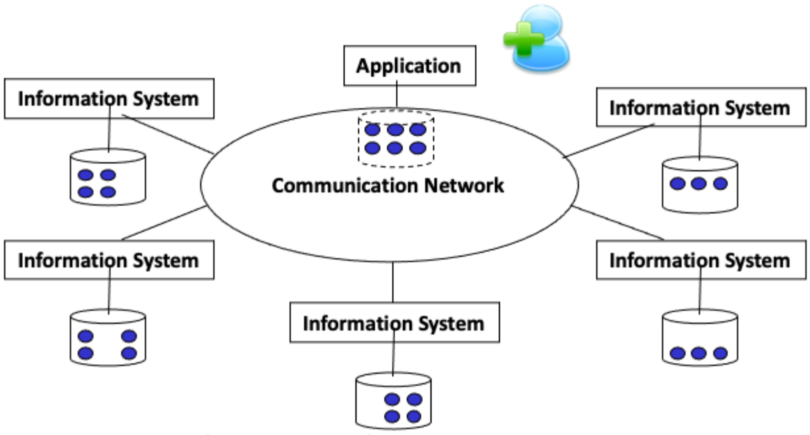


Except in the very early days, information systems had always been used in computer networks. From the perspective of information management this has no significant implications, as long as the information system itself is centralized, i.e., running on one physical node under a single authority. The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

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

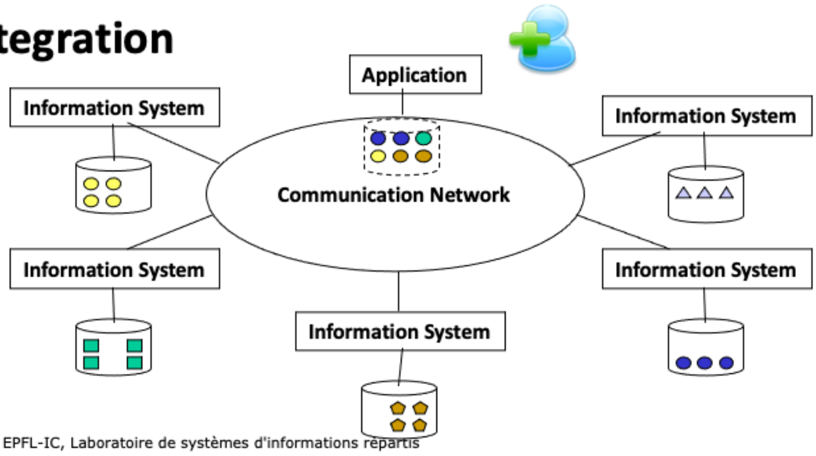
Distributed data management deals with similar questions as centralized data management, namely optimizing the storage of the data and the processing of accesses of the data. The new key element is that data can now be stored at different nodes in a network, and that the cost of data transmission over 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.

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.

Key Issues with Data Integration

The same real world aspect can be represented differently

- Requires agreement on the meaning of shared data
- Relating different models requires often human intervention
- human attention is a scarce resource!

Syntactic heterogeneity

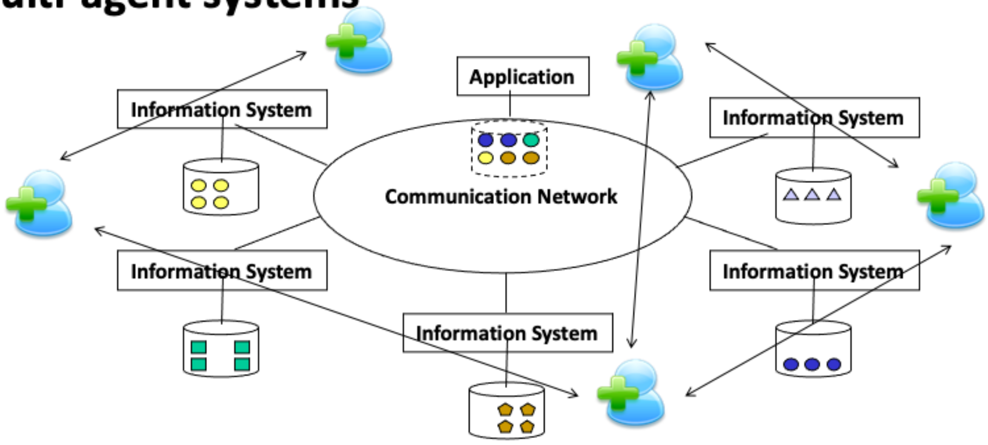
- The same conceptual model can be represented using different logical data models (relational, XML, …)

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.

Autonomy – Distribution of Control

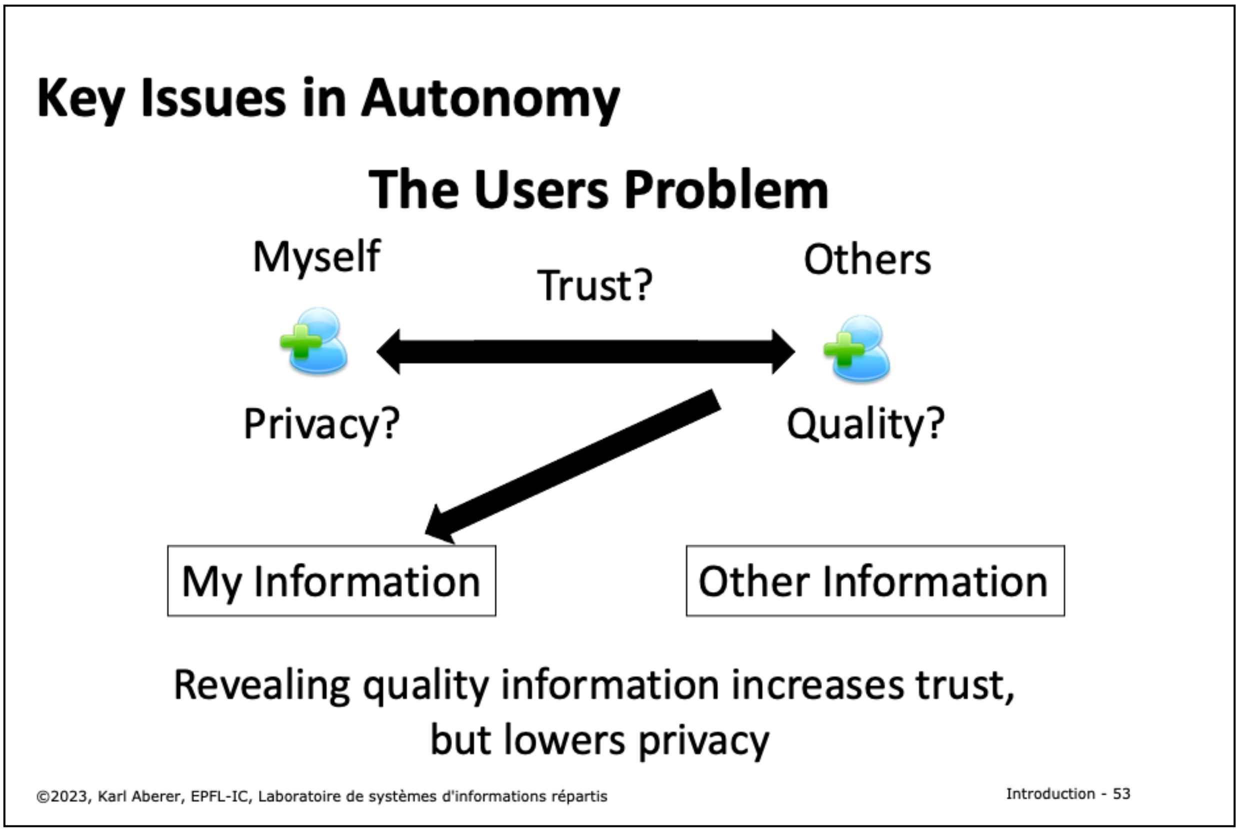
Independent users have to collaborate, coordinate, negotiate, to perform information management tasks

Multi-agent systems



 Introduction - 52

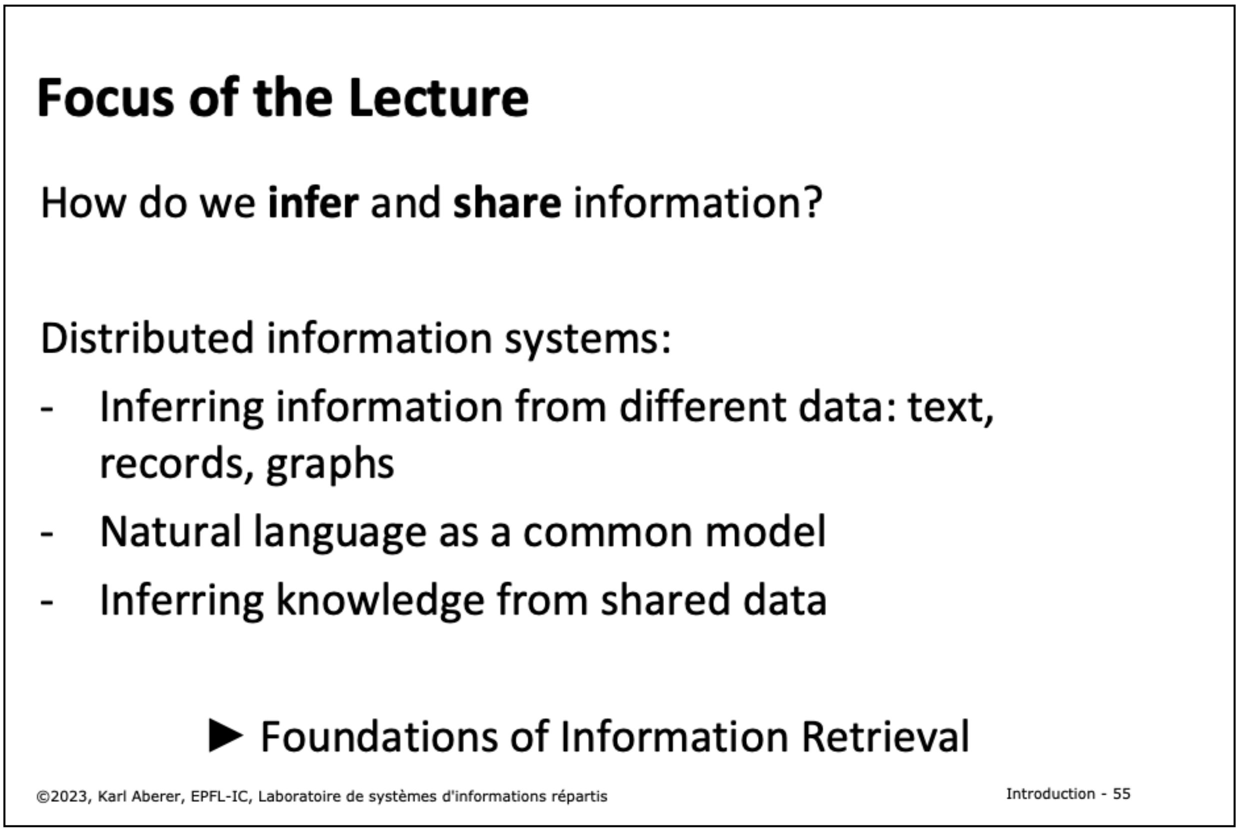
Finally in heterogeneous information systems we must deal with the problem that the different information systems are under the control of different **autonomous** authorities. This poses problems of coordination, mutual trust, and privacy protection. The information systems can be considered as independent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



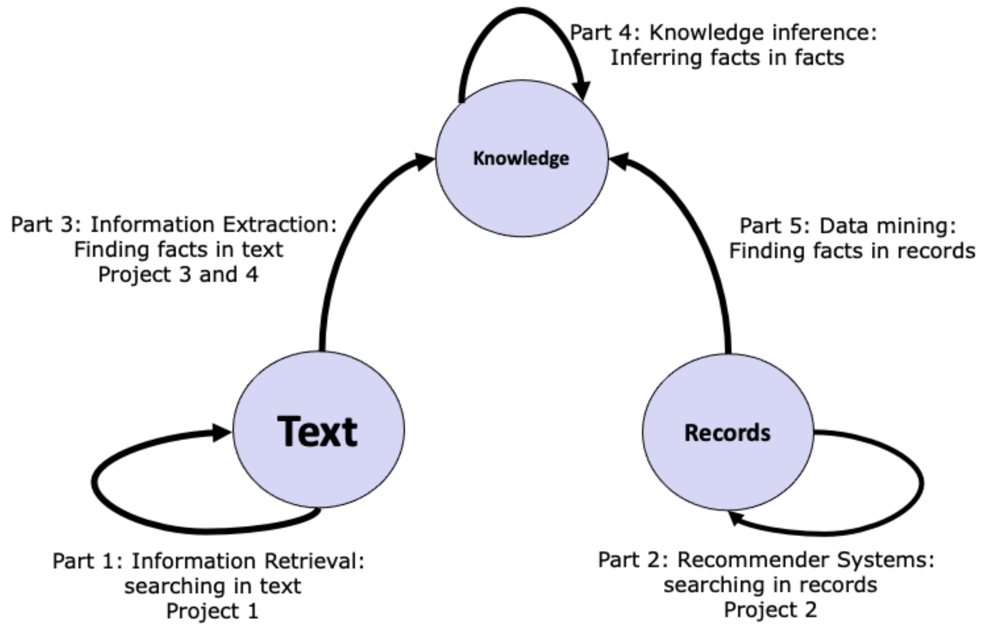
In a setting where different autonomous users exchange information, questions related to the utility of information come into play. As users have their own private interest, they have to consider them in interactions with other users. For example, when receiving information from another user, a fundamental 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.

4. ABOUT THE LECTURE



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 on the question of inferring and sharing of information. Most of the methods that we will cover in this course are related to the field of information retrieval in a general sense. With the rise of powerful language models text retrieval is considered a problem typically solved by such models. However, earlier methods remain relevant for two reasons: (1) understanding the basic mechanisms and (2) as more efficient alternatives in large scale data processing. Since we will mostly be studying fundamental methods from the field of information retrieval the course could also be called Foundations of Information Retrieval.


Part 4: Knowledge inference:
Inferring facts in facts
Knowledge
Part 3: Information Extraction:
Finding facts in text
Project 3 and 4
Part 5: Data mining:
Finding facts in records
Text
Records
Part 1: Information Retrieval:
searching in text
Project 1
Part 2: Recommender Systems:
searching in records
Project 2