Introduction:
Distributed Information Systems - An Overview
Today's Questions

1. What is an Information System?

2. What is a Distributed Information System?
What do you think?

- What is an Information System?
1. Information Systems

• An information system is a piece of software that manages facts about some aspect of the state of the real world for a specific purpose
  - the facts are represented as data using a data model
  - the relationship to the real world is established by an interpretation

• Information = Data that is interpreted

This is a definition of information systems in a nutshell: “An information system is a piece of software that manages facts about some aspect of the state of the real world for a specific purpose”. This definition contains a number of concepts that require explanation:

facts: They are called data. Data is given as some mathematical structure – called then the data structure (that describes what is the domain, the relationships among the objects in the domain, and possible predicates and operations).

real world: The real world is not necessarily our physical real world. It can be anything from human concepts (e.g. a legal information system) to virtual worlds existing in a computer system or network (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.). In order to do so it performs an interpretation of the data, this is where the interpretation comes into play. The interpretation relates the structure (also called the syntax) with the “real world”. The interpretation can be a formalized relationship, if the “real world” is formalized, or just exist informally.

aspect: this implies that there exist many different ways to construct information systems.

What “managing” means and what is the nature of the software that performs this management we want to explore in the following in more detail.
These examples show that in particular there not always exists a unique interpretation and/or purpose for the same facts. It is also not the case that always humans are necessarily involved in the interpretation of the facts. For example, if in a hypothetical fully automated bank the clerks and programmers would disappear, it might be possible that the only interpretation of the bank account data is performed through transaction programs.
We may consider the available interpretations, together with the specifications of the data structures, as the knowledge that is in use. This is illustrated by the above figure from a famous paper from Gio Wiederhold, dating from 1992, (Mediators in the Architecture of Future Information Systems. IEEE Computer 25(3): 38-49 (1992)) It illustrates how in the data loop data is gathered, stored and maintained through state changes, and then when interacting with knowledge (mostly humans, but not exclusively) new interpretations are created through selection, integration, and abstraction. These new interpretations serve new purposes of decision-making and action. From this, new knowledge is gained (via experience and education), which further feeds back in the knowledge cycle. Also, some of the assumptions in Wiederhold’s paper are in the meanwhile made obsolete by the progress of information technology, and in particular the Web, the fundamental model still holds true.
There exists a lot of confusion between the notions of data and information management. The reason is that the two issues are inherently connected as we have seen before, and that the distinction is a matter of emphasis.
What do you think?

- What are typical problems of *data management*?
Data Management Issues

- Problem 1: Data modeling
  - suitable data structures to represent facts (list, set, graph, matrix, relation)
  - data without further structure is called content (text, images, audio etc.)
  - operations for input (update) and output (querying)

- Problem 2: Management of large amounts of data (Scalability)
  - efficient storage schemes (e.g. sparse - dense)
  - efficient access schemes (e.g. indexing)
  - exploiting different media (e.g. memory - disk)

- Problem 3: Safe storage and updates of data
  - on some persistent medium (e.g. memory, disk, tape)
  - independent of lifetime of programs, access pattern, failures
  - consistent updates

- Data Management, Database Systems, Database Management Systems

One key function of an information systems is to “manage” the facts: managing means to maintain them and to perform operations on them, i.e. to maintain a state and to perform state transitions. This is what also transient programs do, but different to them the state should be maintained independently of any programs lifetime, and therefore some medium is required to save it over time. The amount of data that is stored is usually very large, which poses serious problems of efficiency. This aspect of information systems is covered by data management systems.
With respect to data modeling two main categories of models can be distinguished: unstructured and structured. Note that data models are not only defined by the structure, but also by the operations (some examples are given for illustration) and “semantic” constraints. Constraints exclude certain structures that would be syntactically possible but are not allowed.
Structured Data Models

- Structured Data Models
  - Application specific types are captured in Schemas
- Schemas for representation of real-world concepts as types
  - "real-world semantics"
- Advantages of schemas
  - maintain integrity of data
  - optimize access to and storage of data
- Relational data model made the race
  - clean mathematical semantics: optimization, normalization
  - simple to understand

PGA Tour

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Earning</th>
<th>Driving Dist</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T. Woods</td>
<td>6496</td>
<td>294</td>
</tr>
<tr>
<td>2</td>
<td>S. Garcia</td>
<td>2373</td>
<td>285</td>
</tr>
<tr>
<td>3</td>
<td>E. Els</td>
<td>3180</td>
<td>279</td>
</tr>
<tr>
<td>4</td>
<td>D. Toms</td>
<td>2534</td>
<td>271</td>
</tr>
<tr>
<td>5</td>
<td>J. Daly</td>
<td>838</td>
<td>350</td>
</tr>
<tr>
<td>6</td>
<td>C. Dusty</td>
<td>489</td>
<td>290</td>
</tr>
</tbody>
</table>

Among the data models the structured data models have been the by far most important ones. They allow to create for each different application different structures (typically different data types), such that the specific type of structure captures “real world semantics”. In other words the interpretation of the data cannot only be defined for individual facts, but for whole classes of facts, and can be done thus more efficiently at a higher “abstraction” level. Among all structured data models the relational data model was by far the most successful.
Problem: Scalability for Large Databases

- Store data on available storage medium in an efficient manner
  - Blocks, files, network nodes, ...

- Provide efficient access to data for specific addressing methods
  (Indexing)
  - attributes, predicates, paths, ...
  - Data access structure (tree, hash table etc.)

- Example: B+-Tree
  - tree nodes match block size of storage systems
  - tree is balanced
  - all operations (search, update) logarithmic

Database research has devoted an enormous amount of research on ever increasing the efficiency of database systems. Two main problems are addressed: how can one make optimal use of the existing infrastructure by exploiting its physical characteristics, and how can methods be built to efficiently locate data. B+-Tree’s are just one, but one of the most important, examples illustrating both aspects.

However, it must not be neglected that a huge amount of work related to efficient access to data has also been invested in the area of information management. The reason is the following: it is not only the physical characteristics of the environment that can be exploited for efficiency, but it is also knowledge on the nature of the data that can be exploited to more efficiently access it. This knowledge can however only be obtained if there exists an interpretation of the data. For example, in text retrieval knowledge on the characteristics of textual data is exploited in order to devise better storage and access schemes. This work is also closely related to information theory, e.g. aspects related to compression.
B+-Tree

insertion of 6

rectification of tree

©2007/8, Karl Aberer, EPFL-IC, Laboratoire de systèmes d'informations répartis
Problem: Safe storage and updates

- Maintain consistent state
  - in the presence of multiple users (Concurrency)
  - in the presence of failures (Recovery)

- Example
  - acc1 is a data object (tuple, element, object, ...)
  - T1 and T2 are different users, what goes wrong?

<table>
<thead>
<tr>
<th>Transactions T1:</th>
<th>T2:</th>
<th>State acc1</th>
</tr>
</thead>
<tbody>
<tr>
<td>read(acc1)</td>
<td>read(acc1)</td>
<td>20</td>
</tr>
<tr>
<td>acc1 := acc1 + 10</td>
<td>acc1 := acc1 + 20</td>
<td>40</td>
</tr>
<tr>
<td>write(acc1)</td>
<td>commit</td>
<td>30</td>
</tr>
<tr>
<td>Commit</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The second major line of research on databases has been on transactions, i.e., how to run a data management system safely. Here we illustrate just one possible problem that is dealt with by transaction management.
What do you think?

- What are typical problems of *information management*?
Information Management Issues

- **Data overload**
  - more databases, more connectivity, disintermediation
  - more information = disintermediation

- **Information starvation**
  - problem: data supply does not match data demand
  - interpretations used by data provider is different from interpretation used by data consumer

- Human attention is the scarcest resource!

The second aspect of information systems is related to the problem of having multiple interpretations. There exists an abundance of data. The problem is whether the user (or the applications) can interpret it properly. I.e. as long as no proper interpretation exists the data is of no use. This is what is called information starvation. In particular, human attention is the scarcest resource that exists, and humans require the data in a form that they are able to interpret, which is not necessarily the available form.
Supporting Interpretation(s)

- Proper interpretation for supplied data is required!

This figure illustrates the situation: there exist many information systems supplying data, but having their own proper interpretation which does not necessarily match our purpose. There exists some relationship $R$ between the real world aspects the data suppliers relate their data to and the real world aspects we are interested in. Given this relationship (which is typically not formalized) the problem is now to provide a corresponding relationship $R'$ between the data we are supplied with and data that we are able to interpret. Since $R$ is normally not well specified, finding $R'$ is in general a difficult problem. From the viewpoint of the data suppliers introducing $R'$ introduces a new interpretation of their data, and thus creates a new “real world aspect” that they can relate their data to. Thus by creating a proper interpretation of supplied data we implicitly also connect the data suppliers to “new worlds”.
These examples illustrate of how from existing data (supply denotes the interpretation of the supplier), new data (demand denotes the required interpretation of the demander) is created for new purposes. Note that the necessary operations can be everything from simple manipulations of data structures to complex algorithmic procedures such as text or image analysis.
Deriving New Models

• The “database approach”
  - consult the users in an application
  - develop a conceptual model
  - develop, implement and use the logical model
  - re-consult the user and start over

• The “data mining approach”
  - take a learning dataset
  - build a model from it
  - take a test dataset
  - compute how well the model matches

• The “information retrieval approach”
  - ask human users for the relevance of information for a problem
  - model relevance by a retrieval algorithm
  - compare the results: recall, precision

The problem when creating new interpretations (which we have identified earlier as being the central task of information management) is to be able to assess whether the interpretation created match the expectations. Again we can distinguish different methods of doing that (which does not exclude that there exist others).

Both the database and information retrieval approach perform the assessment (mostly) by involving the human user. In contrast, in data mining formal criteria for the evaluation of an interpretation are set up in advance, and then checked once an interpretation has been derived. This makes sense, since in data mining the interpretation relationship exists between existing data and data derived from the existing data, and thus can be formally analysed.
Problem: Semantics (Interpretation)

- Databases: Express the information need by a query explicitly
  - typically using first-order logical statements

\[ \text{Pro} = \{ p \mid \text{Earning}(p) > 6000 \text{ and Driving Dist}(p) > 280 \} \]

- Data Mining: discover hidden dependencies in the data
  - typically using machine learning methods

\[ \text{Earning}(p) > 1000 \Rightarrow \text{Driving Dist}(p) > 270 \]
  (confidence 100%, support 66%)

- Information retrieval: express information need by similarity
  - typically using statistical techniques

In order to create new interpretations of data we can distinguish three kinds of approaches:

1. the database approach: here from an explicitly available structure another explicit structure is generated. To specify the mapping usually first order logic used (though also other functions might be involved, such as manipulation of multidimensional arrays in online-analytical processing). This new structure is associated with an interpretation (e.g. “Pro” is a real-world concept)

2. the data mining approach: here the interpretation is based on the content. Implicit structures that are detected in the content are turned into explicitly available structures, for which presumably an interpretation exists. The example shows an example where the structure is given in form of a rule (effectively a kind of an additional constraint on the data) that has an interpretation.

3. the information retrieval approach: it makes use of implicitly available structures in the content without making these ever explicit. The standard problem that is solved in that way is to provide a relevance relationship among contents that is interpreted in the “real world” as the two contents are relevant to each other (whatever does this mean ?). This relevance relationship is then used to locate starting from one content more related content.
This example shows how information retrieval effectively uses the implicit structure of the content.

- Analysis of term-document matrix reveals the most important dimensions in term space.
- Mapping into low-dimensional concept space.
- Concepts are used to retrieve similar documents.
Data Mining: Real Example

Amazon provides information on « similar items »
- customers who bought this book also bought ...
- customers who read this author also read ...
Therefore
- how can Amazon find out which are the « most similar » books and authors ?

Amazon is a heavy user of data mining technology today.
Anything Else?

• Practical issues
  - building systems
  - using systems
Abstraction plays not only a role when creating new interpretations, but also when architecting an information system. This leads to different abstraction principles that are used to organize the systems that manage the information in a modular way and make them easier to manage. In particular (relational) database systems developed several extremely successful abstraction principles in the system design. Some of these principles are being also applied to other types of information system, more concerned with information management.
Finally, each of the sub-disciplines developed over time its methodologies of how to approach standard problems in a systematic way. We mention here two typical and widely used examples of such methodologies.
This is a somewhat simplified overview of the main problems in data and information management: we can classify them into issues related to modeling (i.e. what are possible representations for the facts we are dealing with), issues related to algorithms (i.e. how can we perform operations on the models efficiently and reliably), and issues related to the implementation (how can we build and use real software systems for using the models and algorithms). For each of these aspects, we can further distinguish two main categories: in modeling there is a clear distinction whether we are interested only in the syntactic issues (i.e. not considering the interpretation of the model) and semantic issues (taking into account the interpretation). In algorithms we can distinguish between issues of reliability (how can one safeguard against all kinds of adverse effects) and efficiency (how can the systems made scalable). With respect to implementation there are issues of architecting the systems and of using the systems in a systematic way. In the following we summarize the main “inventions” that have been made with respect to each of those aspects.
What do you think?

- What is a distributed information system?
Except in the very early days, information systems had always been used in computer networks. This does not imply any fundamental problems in addition to those we have discussed up to now as long as the information system is centralized, i.e. running on one node under a single authority.
There exist however many reasons not to keep all data in a single node on the network. Some of these reasons are related to the improved use of existing resources. We might want to move data close to the node where it is processed, we might take advantage of parallel processing of the data, and we might want to avoid bottlenecks in order to improve scalability of the systems. Thus, we want to distribute the data. Nevertheless we don’t want to give up the abstraction that we are still accessing a single information system that is running under a single authority.
Things can become more complex if we give up on the assumption, that a distributed information system should appear like one single information system when it is distributed over a network. In fact, we might want to make the distribution visible in order to enable multiple authorities to infuse their knowledge into their network, in order to make information systems that where formerly separated working together, and in order to make different interpretations of data available. Thus we are considering distribution of knowledge. In order to make use of such a logically distributed information system we need to support the application/user in order to overcome some of the problems related to logical distribution, without necessarily making the fact that the data is logically distributed completely transparent.
What do you think?

• What are specific problems to be solved to handle *logical distribution*?
Logical Distribution: Technical Problems

• Logical Distribution
  - autonomy of nodes can lead to syntactic heterogeneity (models, schemas, data) and semantic heterogeneity (interpretation)

• Overcome heterogeneity: information integration
  - Problem 1: overcome syntactic heterogeneity
    - common data model $\rightarrow$ semi-structured data
  - Problem 2: overcome semantic heterogeneity
    - schema mapping and integration
    - agreement on common schemas $\rightarrow$ ontologies
  - Problem 3: evaluate semantic closeness $\rightarrow$ distributed retrieval

Dealing with logical distribution poses both new algorithmic and systems issues.
Problem: Syntactic Heterogeneity

- Semi-Structured Data Models
  - representation of relationships: hypertext graphs, semantic networks
  - self-describing: schema-less data
  - volatility: changing schemas
  - standardization: represent data from any structured model
  - serializability: exchange of documents

With the emergence of the web a new class of models gained importance, semi-structured data models. There exist various reasons for their success, all due to issues related to distribution and communication:

- serializability (i.e. the possibility to represent the data as a string that can be sent over a communication channel) is a more obvious reason
- the relaxation of using schemas, which are very useful if the interpretation of the data is stable, but not if the same data is used under different interpretations. Still allowing to maintain some schematic information is supported
- simple representation of relationships (referencing is one of the fundamental needs in a distributed information system)
This figure illustrates the use of serialization: within information systems data is kept in potentially complex structures. For communication it has to be converted into a representation that is “serial”, i.e. a string. Data models that provide a canonical serialization, such as XML, are thus a natural choice in a communication intensive environment.
Running Example

The Amazon Website contains semistructured data
- mix of regularly and irregularly structured data
- navigation
Therefore
- requirement to store, query and navigate this kind of « non-relational » data efficiently
Problem: Schema Mapping

- Assume all data represented in canonical data model (e.g. XML)
  - detect correspondences
  - resolve conflicts  => a methodology!
  - integrate schemas (mapping)
- Mappings are frequently expressed as queries (e.g. XML Query)

One important approach to overcome semantic heterogeneity is to match between different data schemas (we could do the same also at the level of individual facts/data, but this would be less efficient). The standard methodology is to first identify which constituents of a schema correspond to each other, then to identify whether there exist differences in the way the same information is represented and find ways to overcome them, and then to construct a mapping for the data. In databases the mapping is typically performed using a query language.
This example illustrates the standard architectural approach for integrating information systems: first syntactic heterogeneity is overcome, by wrapping the data and transforming it into a common format (e.g. XML). Then semantic heterogeneity is overcome by mapping the various schemas into a common global schema. Nowadays, many of these systems use a data warehousing approach, i.e. the data is physically stored in the integrated information systems. Among the most popular applications of these systems are prices comparison web sites.
Amazon is also subject to the “treatment” of prices comparison sites, such as www.addall.com.
Problem: Ontologies

- Ontologies are an explicit specification of a conceptualization of the real world (Gruber, 1993)
- Ideally
  - different information systems agree on the same ontology
  - relate their model/schema/data elements to the ontology
  - mapping can be constructed via the ontology
- Issues
  - ontology languages (e.g. OWL), mappings and engineering
- Problem: requires agreement on the conceptualization of the real world!

In order to obtain a better handle on the problem of dealing with different interpretations of data, a possibility is to make the interpretation a formally described entity. This requires the use of some “proxy” for the real world that is formally describable. These “proxies” are called ontologies.

Example: The reader of this text may consider the text itself as an attempt to create an ontology for the area of distributed information systems.
Reconciling distributed knowledge is another important aspect of logical distribution of information. The most popular example for this is Google, which uses the link structure of the Web (where the interpretation comes from the fact that setting a link expresses something about the importance of the referenced site) in order to derive information on the relevance of documents.
What do you think?

- What are specific problems to be solved to handle *physical distribution*?
Physical distribution deals essentially with the distributed counterparts of the standard problems in central data management. Physical distribution introduces however one aspect that is essentially different from a centralized setting. Whereas in a centralized setting data is maintained (communicated) only over time, in a distributed setting it is additionally communicated over space (between the nodes). This leads to a variety of possibilities of how the relationship between data sources and data sinks can be organized, which we can characterize as different information dissemination models.
In distributed storage management the key question is of how to position the data in the network most effectively.
Having data that is dependent on each other (e.g. replicated) in the network poses the problem of performing operations on the data consistently in a distributed setting, which leads to the problem of distributed transaction management.
If data is distributed over a network, in order to retrieve it the information system has to solve the problem of how to efficiently locate the nodes that contain the desired data and to retrieve the data from there.
Running Example

Amazon operates with distributed Web sites
- accounts are global (.com .de .uk .ch)
- delivery from respective country
  therefore
- data both locally available and globally maintained
- data distribution optimization problem
The dimension of communication opens a range of possibilities of how changes on the data state are propagated in the network (this is a problem that does not occur in a central information system, since there exists only a single state). We can classify those changes according to [Ozsu, Liu, Yang 1998] with respect to three dimensions, related to communication: control of the exchange, event leading to an exchange, and communication model.
In the classical Client-Server setting a client sends a request to the information system, and receives as a result a response.
In mobile broadcast a server broadcasts periodically all the available information to all clients (within its range). The clients select from the incoming data stream the data they are interested in.
In a publish-subscribe system the clients first deposit what information they are interested in (profile), and whenever new data arrives that is matching the profile the server sends the data to the client.
Amazon provides a number of possibilities to subscribe for email notifications: special occasions, new books etc.
In P2P information systems no server exists. Every client propagates search request to local neighborhood, which further spreads the rumor till eventually some node can provide the requested data.
Scope of DIS

- knowledge
- data
- non-distributed
- distributed
- heterogeneous
- autonomous
- model
- algorithm
- system
Robust Technology

- (Relational) Database Servers
- Transaction Monitors and Web Application Servers
- Information Retrieval Engines and Web Search Engines
- WWW Standards
- Data Warehousing and Mining Tools
Evolving Technology

• New Forms of Physical Distribution
  - Mobility, Personalization
  - Embedded Information Systems
  - P2P Systems
  - Sensor Networks

• Semantic Interoperability
  - Semantic Web
  - Ontologies

• Self-Organization
  - Adaptability and Learning
  - Information Agents
  - Information Economy