Aim of the Talk

- This is not a prediction of future database systems
  - "Prediction is very difficult, especially if it's about the future."
    [Mark Twain/Niels Bohr]

- It is not a trend analysis
  - No statistically provable results

- It’s only a personal view on what kind of functionality future databases should provide
  - Research challenges
  - Not comprehensive, only some examples
  - I have more questions than answers
The 3 Challenges are …

to overcome the following limitations

- Schema first requirement
- Data integration problem
- Data access/retrieval is for experts only

1. Challenge: Schema first requirement

- Data can be stored only after the schema design but …

- Database modeling is a challenge
  - Some data comes with a description (metadata) others not
  - Difficult to understand other peoples’ data
  - How about the structure (content, linguistic, format) of textual data?

- Data model evolves over time
  - semantic drift (meaning of data changes)
  - schema growing process is manual

- How do we get rid of the schema?
  - Or at least automate schema design and management
1. Challenge: Example

*Do you understand this schema?*

And this schema has only 36 annotated tables

The Vision

*No manual modeling or schema design required*

*DBMS has*

- no Schema or
- defines and manages Schema automatically
  - Schema derived from (example) data stored
  - Automatic schema evolution

*Research so far*

  - deals only with structured data (from OLTP Systems)
  - uses triples (id, property, value) for storing data
- D. Maier. Profiling Dataspaces:Understanding (and Using) Other People’s Data, Klaus Dittrich Memorial Symposium, Zurich, CH, 2008
  - reports on a study to find the schema for a medication list RxList and related standards like NDCD, RxNorm with help of Quarry metadata explorer (RDF-like data model) and other data profiler tools
The Vision

No manual modeling or schema design required

Some ideas

- Use meta information from objects to get structure info
  - Examples: obj.class(), obj class instanceVariables class
- Use DTD or XML Schema info for XML documents
  - Example: 
    ```
    <xml version="1.0" standalone="no">
    <!DOCTYPE hello SYSTEM "hello.dtd">
    <hello>Hello world!</hello>
    ```
- Use layout/linguistic information from sample text/htm/xml documents with known semantics
- Use statistical information to find the most likely data type or coding
  - Example: always ASCII digits → integer
  - always ASCII digits plus punctuation → decimal

Aim

Challenges

1. Vision
2. Vision
3. Vision

source: blackberry.com
The Vision

- No manual modeling or schema design required

Some ideas

- Stay data type tolerant
  - Example: 9/10 digits, 1/10 characters → store as string
- Stay schema tolerant
  - Example: unclear relationship
    - then use no relationship and analyze query profile
  - Example: 9/10 1:* relationship 1/10 *:* relationship
    - then allow both until you can rule one out

Owner

Car

2. Challenge: The Data Integration Problem

Example: Complex ETL process in Data Warehousing

... and this is only one data source
2. **Challenge: The Data Integration Problem**

*Example: Complex ETL process in Data Warehousing*

- What you really need to integrate …
- … Your data sources are everywhere!

Aim

**Challenges**

1. Vision
2. Vision
3. Vision

Data must be copied and integrated into a schema (periodical ETL-process)

- Data is always out of date
- Much of the data is never used

Data provenance and ownership is lost

- Control over data is lost
- Data quality unknown

How can we avoid the integration of highly scattered but interrelated data?
The Vision

- **Database is virtual, but manages an interrelation schema or at its best an integration schema**

- **In situ Storage**
  - Data remains where it is created
  - Ownership and provenance preserving

- **Data may be cached for performance**
  - Trade off between performance and consistency, resp. data freshness

---

Aim

1. Vision
2. Vision
3. Vision

---

Research so far

  - introduces dataspaces concepts, in situ data, collection of relationships

  - reports on experiments with object caches, in situ queries, column-wise storage and memory blades, incremental data loads
The Vision

Database is virtual, but manages an interrelation schema or at its best an integration schema

Some ideas for storage:
- Data stays at its source location
  - Preserve data provenance
- Select as early as possible
  - Requires query decomposition
- Move data only for performance (caching, hoarding)
  - Needs high bandwidth with low latency

Some ideas on performance:
- How to accelerate access beyond caching data and high bandwidth?
  - Work in parallel
  - Smart indexes, e.g.
    - Index for metadata identifies redundant data
    - Index knows cost function for data access
    - Index on stored query results
  - Relaxed consistency requirements with regard to time dimension
    - Store and forward
    - Semantically equivalent query rewrite
The Vision

- Database is virtual, but manages an interrelation schema or at its best an integration schema

- Some ideas: taming the data flood with a hierarchical interrelation schema

Aim

Challenges

1. Vision
2. Vision
3. Vision

3. Challenge: data access and retrieval only for experts

- Only syntactic queries possible
  - Absurd join operations possible
  - Need to learn SQL and know the schema

- How about information retrieval in text documents?
  - Only syntactic pattern matching

- ACID transaction model is not adequate
  - Long, nested transactions [Korth/Stegele, Wang/Peng]
  - Serializability is often too restrictive
    - Semantic transaction steps [Garcia-Molina, Farrag/Özsu]
    - Multirevision reconciliation [Phatak/Nath]
    - Escrow serializability [Laux/Lessner]

- How can we make the database more “intelligent”?
3. Challenge: data retrieval

Example: What is the average width for images of Mark Twain?

```
Select avg(width) from ???
```

What is SQL?

Keyword search

Average width image Mark Twain

What you expect, is a number

But Google returns …

? Niels Bohr Average = Albert Einstein?
3. **Challenge:** data retrieval

**Example:** What is the average width for images of Mark Twain?

No images found for keyword query.

---

**The Vision**

"Intelligent" data access

**Some ideas**

- Exact results, where possible
  - Regular structure
  - Irregular structure
    - Keyword query
    - Semantic query
- Intentional query (approximate results)
  - Query is not bound to a schema
    - No SQL, but declarative
  - Query must "understand" your intention
    - Ontology, query profile needed
- Active database, agents
  - Signal/show new information
Some innovative examples

Keywords query on structured data

Example: SQAK [Tata, Lohman] builds SQL aggregates using keyword query like:

Mark Twain average image width

results in a SQL query derived from the schema similar to:

```sql
select avg(width) from
(select img_width as width
from image join user ...
where user like 'Mark Twain%'
union
select oi_width as width
from oldimage join user ...
where user like 'Mark Twain%')
```

Linguistic search

Some innovative examples

WebTable search [Cafarella et. Al]
- Extracts relational information from Web-pages including Metadata
- Schema auto-completion via table-headers and column name matching
- Synonym finder or translator via correlated tables

Ideas for concurrent data access

Splitting transaction into compatible atomic steps [Gracia-Molina, Farrag/Özsu]

Example: Money transfer from account a to account b. T1 and T2 are in conflict, but T1(a) and T2(b), T1(b) and T2(a) are compatible

Interleave compatible portions

Reorder portions for compatibility
**Vision**

- Ideas for semantic transaction processing

- Semantic classification of transactions that allow escrow serialization [Laux/Lessner]

- Example: Money withdraw from account a. T1 and T2 are in conflict, but T1 and T2 are escrow serializable (reconcilable).

- Open question: getting performance and semantics?

**Discussion**