Databases in Software Engineering: a Roadmap

Klaus R. Dittrich

Department of Information Technology
University of Zurich
Database Technology Research Group

Winterthurerstrasse 190, CH-8057 Zürich
e-mail: dittrich@ifi.unizh.ch, http://www.ifi.unizh.ch
Tel.: ++41-1-635 4312, Fax: ++41-1-635 6809

©2000 Klaus R. Dittrich

outline

- what is the problem ?
- databases & database technology: what is it, after all ?
- DB support for SEE: what is required ?
- what has been achieved ?
- are we done with this ?
- where to go, what to do, and who ?
what is the problem?

- software engineering involves lots of information that is being produced, has to be maintained and managed, and will eventually be consumed/inspected/...
  (comes in various kinds, with various processing requirements etc.)
- database technology provides lots of (proven!) powerful and efficient solutions for information management
- looks like an ideal solution, doesn't it?
- however (as usual): it's the small things that cause problems ("the devil is in the detail")

database

A collection of interrelated data with the following properties:

- permanently available (i.e. explicitly controllable lifespan)
- potentially large
- integrated (across multiple applications with overlapping information; controlled redundancy)
- usable independently of program that created it (data independence: mutual change immunity AP - DB)
- multiuser operation ("parallel access")
- consistent, safe, secure ("data qualities")
- comfortably, flexibly and efficiently usable ("associative access")
- (possibly) distributed in computer network (→ "transparency")

database technology

The entirety of concepts, methods, systems and tools for the organization and operation of databases
**database management system (DBMS)**

software for the management and operation of databases with the above-mentioned properties

implements among others:

- **a data model:** conceptual framework for logical data organization means for data structuring/data description ("data definition"/DDL $\rightarrow$ schema, views)
- **means for data access and data manipulation:** DB-operations (DML) query language
- **control mechanisms:**
  - transaction model (incl. recovery, synchronization)
  - consistency maintenance
  - trigger
  - access control
  - backup and archiving
  - secondary storage management (access paths etc.)
  - distribution management in network

**DB technology today**

- **DBMS are a huge commercial success**
- **data models, query models and transaction models** have been researched extensively
- **some recent and current developments:**
  - from structured to (also) unstructured and semistructured data
  - new data types (multimedia), object-oriented/object-relational models
  - extended transaction models: from concurrency to (also) cooperation
  - advanced query models, including IR-style retrieval
  - DBMS-support for advanced requirements like time, versioning, rule-based triggering, ...
  - distribution, federation, integration, ...
DB support for SEE: what is required?

- software engineering environment (SEE):
  set of tools supporting tasks of the SE process,
  complemented by appropriate infrastructure

- tools produce and consume/inspect data

- thus we need
  - representation and persistent storage of software artifacts
  - integrity and consistency management
  - query and retrieval facilities
  - version and configuration management
  - cooperation support and workspace management
  - process support (workflow management and enactment)

what has been achieved?

- over the years, many efforts have been undertaken towards "software engineering databases"

- typical (simplified) architectural view of SEE:
what has been achieved?

- tricky issue: what goes where (tools vs. repository system)?
- which kind of repository system?

- **database management system (DBMS)**
  artifact query & retrieval, concurrency control, transactions, consistency/integrity/access control, ...

- **object management system (OMS)**
  artifact representation, (some) integrity control, low-level navigation interface

- **file system**
  persistent storage
  (some) concurrency & access control

what has been achieved?

- OMS vs. DBMS?
- proprietary vs. general repository system?
- commercial system vs. research prototype?

(subjective) summary:

a lot of work, without much agreement, conclusive results, or real success stories
are we done with this?

- certainly not ... !
- ... for at least two reasons:
  - advances in database technology give hope for better results in the future
  - cannot imagine that current state is satisfactory for SE community (outsourcing of tasks works elsewhere, too)

- what we do need: clear conception of exactly what DB services are needed, in which form

where to go, what to do, and who?

- a fresh look at "DBMS": collection of services instead of monolithic "dinosaur"
- component technology, extensibility
where to go, what to do, and who?

- databases and software engineering:
  - a marriage of love?
  - a marriage of convenience?
  - a love-hate relationship?
  - a non-relationship?

- interdisciplinary research is advocated so much, and even works between computer science and others; why not try it more between subareas of computer science?

- closer cooperation between communities could achieve a lot!