1. Motivation

Special Requirements of Data Management for Engineering Applications
Overview

• What are typical/conventional applications?
• Why is it necessary to talk about engineering data?
  – What is different about engineering data?
    • Typical structure
    • Special requirements
  – What is different about how data is accessed in engineering applications?
    • Queries
    • Updates
  – How is engineering data used differently?
    • Transactions
    • Processes
    • Applications
• What is the current state of the art for DMEA?
Conventional Applications

- Document-oriented (files), e.g.
  - Textprocessing
  - Desktop Publishing
  - Web
  - Spreadsheets
  - Media: Image, Video, Audio

- Structured Data (Databases), e.g.
  - Administration of Company / Organization
  - Enterprise Resource Planning
  - Customer Relationship Management
  - Finance and Banking

Schallehn: Data Management for Engineering Applications
## Conventional vs. Engineering Data

<table>
<thead>
<tr>
<th>Conventional Data</th>
<th>Engineering Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple (flat) records</td>
<td>Complex Objects</td>
</tr>
<tr>
<td>• One record represents one real-world object</td>
<td>• <strong>Hierarchical structures</strong>: an object may contain other objects, etc.</td>
</tr>
<tr>
<td>• Simple relationships (e.g. by identifying keys)</td>
<td>• <strong>Network structures</strong>: objects may form new structures from complex relations</td>
</tr>
<tr>
<td>Records represent current state</td>
<td>Need to represent development of objects as versions</td>
</tr>
<tr>
<td>Records as a single representation of a real-world fact</td>
<td>Need to represent concurrent/parallel manifestations of one object as <strong>variants</strong> or <strong>configurations</strong></td>
</tr>
<tr>
<td>Records have a fixed structure</td>
<td>Engineering data is unpredictable and requires <strong>flexible structures</strong></td>
</tr>
<tr>
<td>Different records are most often loosely connected along relations</td>
<td>There are <strong>strong dependencies</strong> between different objects, often existential</td>
</tr>
</tbody>
</table>

Schallehn: Data Management for Engineering Applications
Hierarchical Structures

- Typical **product structure**: a complete product consisting of parts or assemblies, where assemblies again may consist of assemblies or parts.
- Each simple part or composition may be complex itself.
- E.g. input for production planning (Bill of Material) etc.

Schallehn: Data Management for Engineering Applications
Network structures

- Several components may be connected in a network and form a higher level unit.
- Each component as well as the relationships can be complex.
- E.g. electronical engineering, telecommunications network, embedded components connected via a fieldbus, etc.

[Source: Philipp Ludwig]
Versions

• Temporal sequence of different development states of one object
• For internal usage: milestones or consistent states for possible rollbacks
  – Work in progress
  – Fitness for certain development steps (simulation, mockup, field test, etc.)
  – Alpha, beta, release candidate
• For external usage: release of an improved product of the same development line

Schallehn: Data Management for Engineering Applications

VW Golf I
VW Golf II
VW Golf III
…
Variants

- Alternative states of one object existing in parallel
- Internal variants may exist due to concurrent design activities
- External variants exist, e.g. to address different market segments or different application scenarios
- Different external variants may have different properties, offer different functionality, and fulfill different requirements

Schallehn: Data Management for Engineering Applications
Versions and Variants

• Both represent different alternative representations of one design object
• Are often managed in a common context
• Specialized systems offering according operations, e.g.
  – Check out
  – Branching: creating a new variant
  – Merging
• Certain naming und numbering systems to identify versions and variants internally and/or externally

Schallehn: Data Management for Engineering Applications
[Source: http://en.wikipedia.org/wiki/Microsoft_Windows]
Flexibility

• Conventional data within one universe of discourse often has a fixed structure (records, attributes, relationships, etc.)

• The structure of engineering data on the same real-world object may vary widely for
  – different applications
  – different companies
  – different product categories
  – different process instances
  – different versions or variants
Strong dependencies

• Strong, e.g. existential, dependencies are more common in engineering applications due to complex product structures

• Existential dependencies:
  – Relationship among objects, where a dependent object may not exist without a
  – E.g. rooms are existentially dependent on the building they are in

• Non-existential dependencies:
  – Objects in a relationship may exist independently of each other
  – E.g. students and lectures
(Non-)Existential Relations in ER and UML

**Non-Existential Relationship**

- **ER**
  - Student
  - Course
  - 
  
- **UML**
  - Team
  - Member
  - 

**Existential Relationship**

- **ER**
  - Building
  - Room
  - 
  
- **UML**
  - Book
  - Chapter
  - 

Schallehn: Data Management for Engineering Applications
## Predominant Access Characteristics

<table>
<thead>
<tr>
<th>Access to Conventional Data</th>
<th>Access to Engineering Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple access patterns for read operations</td>
<td><strong>Complex access patterns</strong> for read operations</td>
</tr>
<tr>
<td>• Retrieval of single records (exact match)</td>
<td>• Retrieval of complex objects (entire or partial hierarchies or networks)</td>
</tr>
<tr>
<td>• Partial match or range queries</td>
<td>• Queries including complex <strong>spatial conditions</strong></td>
</tr>
<tr>
<td>Update operations on single records</td>
<td>• <strong>Navigational access</strong> along relationships</td>
</tr>
<tr>
<td>Small amounts of data in one query</td>
<td>Updates may involve huge fractions of data</td>
</tr>
<tr>
<td>Small to possibly huge numbers of users</td>
<td>Typically big to huge results</td>
</tr>
<tr>
<td></td>
<td>Small groups of users</td>
</tr>
</tbody>
</table>
Spatial Access

- Access to geometrical (2D or 3D shapes, geographical, architectural, etc.) often based on their position in space, e.g. objects in current viewport to be rendered
- Conforms to multi-dimensional range query, e.g. 2D-window
  \[ xmin \leq x \leq xmax \]
  \[ ymin \leq y \leq ymax \]
- Special support in databases

Navigational Access

- Access pattern of following relationships to retrieve objects
- E.g. computing the transitive closure or traversing a tree
- Often a programming pattern, but also carried out by users interactively

[Source: PERT Chart by Thomas Baier]
# Predominant Usage Patterns

<table>
<thead>
<tr>
<th>Usage of Conventional Data</th>
<th>Usage of Engineering Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isolated work of single users</td>
<td><strong>Collaborative work</strong> of groups of responsible engineers</td>
</tr>
<tr>
<td>Short time required for single tasks (transactions)</td>
<td><strong>Long sessions (transactions)</strong> to carry out engineering tasks</td>
</tr>
<tr>
<td>One-shot read and update accesses</td>
<td><strong>Interactive and iterative</strong> re-finement of complex objects</td>
</tr>
<tr>
<td>One or few application(s) used to access the data</td>
<td><strong>Diverse applications for different tasks</strong> accessing the same or strongly related data</td>
</tr>
</tbody>
</table>
Collaborative Work

• Members of groups of engineers concurrently and collaboratively working on
  – the same design or
  – different but related aspects of the same design (maybe using different applications/tools)

• May result in possible conflicts, data loss and inconsistencies

• Resolution mechanisms integrated with management of internal versions and variants, workspaces, check in/check out, etc.

[Source: The DLR Concurrent Engineering Facility]

Schallehn: Data Management for Engineering Applications
same time
synchronous

Face to face interactions
decision rooms, single display
groupware, shared table, wall displays, roomware, ...

Continuous task
team rooms, large public display,
shift work groupware, project management, ...

Time/Space Groupware Matrix

same place
colocated

different place
remote

Remote interactions
video conferencing, instance messaging, chats/MUDs/virtual worlds, shared screens, multi-user editors, ...

Communication + coordination
e-mail, bulletin boards, blogs, asynchronous conferencing, group calendars, workflow, version control, wikis, ...

Long Transactions

• Product development as creative process: activities typically consistent of long sequences of step-wise and iterative modifications of engineering data
• Processes very loosely structured
• Consistent state may be reached after hours, days, weeks or even longer
• Exclusive access to avoid side-effects data may be prohibitive because of decreased potential for parallel activities
Diverse Applications

- Along product life cycle (including product development phases) engineers may use many different applications with slightly different requirements.

- Focus here on:
  - CAD (- Design)
  - CAM (- Manufacturing)
  - CAE (- Engineering)
  - PDM (Product Data Management)
  - PLM (Product Lifecycle Management)

- Collaboration (within phases) and processes (across phases) requires interoperability:
  - Integration of data (e.g. in one database system)
  - Formats/standards suitable for data exchange
Schallehn: Data Management for Engineering Applications

[Dankwort, Weidlich, Guenther, Blaurock, 2004]
Schallehn: Data Management for Engineering Applications

Summary (Motivation)

• Engineering applications with very specific requirements regarding data management
  – More complex structures
  – Access patterns focusing on complex objects
  – Usage in creative, interactive and collaborative processes

• Conventional solutions for data management are commonly used in Engineering applications
  – File systems with standardized/proprietary file formats
  – Relational Database Management Systems (RDBMS)

• Advanced data management solutions provide some suitable concepts and are frequently used
  – Object-Relational DBMS
  – Object-Oriented DBMS
  – NoSQL DBMS