Adabas —
Concepts of a Database for Mission-Critical Applications

Friedrich-Schiller-Universität Jena
Lehrstuhl für Datenbanken und Informationssysteme
Prof. Dr. Klaus Küspert

— February 1, 2011 —

Software AG
Arno Zude, R&D ETS Adabas
Agenda

1) Software AG at a Glance  
2) About the History of Adabas  
3) Adabas Concepts  
   ▪ Data Model  
   ▪ Data Structures  
   ▪ Recovery  
4) Accessing Adabas  
5) New Developments  
6) Success Factors
Software AG at a Glance

Founded in 1969
2nd largest German software company
- Headquarters in Darmstadt
4th largest European software company

Ca. 1.2 billion (10^9) € revenue in 2010
Ca. 5,700 employees worldwide
- >2,000 in Germany
- ca. 850 in R&D

Represented in 70 countries
More than 10,000 enterprise customers

Ca. 29% owned by Software AG Foundation
Software AG at a Glance — Products

- 1st relational-like, transactional high-performance database
- One of the first 4th-generation programming languages
- First Business-to-Business server and SOA integration platform
- First business process analysis & design platform

Adabas
Natural
webMethods
Software AG at a Glance — A Global Company

R&D Labs in All Major IT Hotspots
About the History of Adabas

“Adaptable Database System”

Adabas development started in 1969
- first on mainframe
Relational-like, transactional, multi-user
Used across all industries
- Banking, insurance, logistics, government, etc.
More than 2,000 customers worldwide
Developers in Germany, US, UK
- Darmstadt, Reston/VA, Denver/CO, Derby

Runs on mainframes and open systems
- z/OS, z/VSE, z/VM, BS2000/OSD
- Unix, Linux (incl. z/Linux), Windows
Adabas Concepts — Data Model

Adabas database
- Multiple files in a database
  - Multiple records in a file
    - Multiple fields in a record

Linking different files via common fields
- no pointers between records
  - like in relational model
  - unlike in hierarchical, network models

Read & update operate on individual records
- not sets of records

Search operations create lists of record IDs
- not protected against parallel update

Differences to relational model
Adabas Concepts — Data Model II

Various data types for fields
- Character, binary
- (Un)packed decimal numbers
- Fixed-point, floating-point
- Large objects (e.g., multi-media)

Fields can be defined to have multiple values
- Multiple-value fields

Groups of fields can be defined to have multiple occurrences
- Periodic groups

Related data is kept close together
- Efficient data access
- Differences to relational model
Adabas Concepts — Data Model III

Fast read & search access via indexes
- called “descriptors”

Several types of descriptors
- (Simple) descriptor
  - One full field
- Subdescriptor
  - Part of one field
- Superdescriptor
  - Combination of parts of several fields
- Hyperdescriptor
  - Combination of several fields
  - Value(s) determined by user exit
- Collation descriptor
  - Sort sequence defined by user exit

Flexible, efficient data access
Adabas Concepts — Data Structures

Container datasets (disk files)
- Fixed block sizes, direct access
  - Block = minimal I/O unit
- DATA — Data storage
  - The actual payload data
- ASSO — Associator
  - File directory
  - Field definition tables
  - Indexes
  - Free space management
- WORK — Work file for temporary use
  - Journal of updates for restart recovery
  - Search results
- PLOG — Protection log
  - Journal of updates for archive recovery
  - Copied off for archival when full
Adabas Concepts — Data Structures II

Data Compression
- Field data typically stored in compressed form
  - Character: No trailing blanks
  - Numeric: No leading zeros
  - Null-value suppression option
- Compact storage, efficient access

Data storage (DS)
- Variable-length data records
- One or more records per block
- Record moved to other block if it becomes too large to fit
Adabas Concepts — Data Structures III

Address converter (AC)
- Each record identified by its “ISN”
  - Internal Sequence Number
  - A unique number, usually serial
- AC maps ISNs to DS blocks

Index
- Index maps descriptor values to ISNs
- Stored as inverted lists
  - B*-tree
Adabas Concepts — Data Structures IV

Finding data records — from index via address converter to data storage:
Adabas Concepts — Recovery

Restart recovery

- Performed if database server fails
  - Power failure
  - Canceled by operator
  - System problem
  - Software problem

- When server is restarted ...
  - Protection data on WORK is used to
    - detect need for restart recovery
    - repair inconsistent internal data structures (e.g., index trees)
    - redo committed updates that were not written out prior to the failure
    - undo uncommitted updates that were written to the database
Adabas Concepts — Recovery II

Archive recovery

- Performed if database gets damaged
  - Disk failure
  - Application misbehavior
  - Software problem

- To repair damaged database or file(s) ...
  - Restore archive copy of database/file(s)
  - Use the protection data on the PLOGs to
    - replay all updates performed between the time the archive copy was created and the time the damage occurred
  - Alternative: Use the PLOGs to
    - roll back all updates between the current time and the time first damage occurred
Accessing Adabas

Application programming interfaces to Adabas

- Direct call interface
  - For example, from COBOL, C
- Natural (Software AG’s 4GL)
- Adabas SQL Gateway
  - ODBC
  - JDBC
- Adabas SOA Gateway
  - SOAP, REST
Accessing Adabas — Direct Call Interface

Set up Adabas Control Block and up to five Adabas buffers with ...

- Adabas command to be executed
- Data to be searched, read, or inserted/updated/deleted

Example (COBOL):

```cobol
MOVE 'S1' TO ADACB-COMMAND.
MOVE 'I' TO ADACB-COP2.
MOVE DBID-FNR TO ADACB-DBFILE.
MOVE 'FIND' TO ADACB-CID.
MOVE 'AA,8,A,AE,20,A,AJ1-3,20,A.' TO ADABAS-FORMAT-BUFFER.
MOVE 'AE,9,A,D,AJ,5,A.' TO ADABAS-SEARCH-BUFFER.
MOVE 'KUESPERT JENA ' TO ADABAS-VALUE-BUFFER.

CALL 'ADABAS' USING ADABAS-CB,
     ADABAS-FORMAT-BUFFER,
     ADABAS-RECORD-BUFFER,
     ADABAS-SEARCH-BUFFER,
     ADABAS-VALUE-BUFFER,
     ADABAS-ISN-BUFFER.
```
Accessing Adabas — Natural

Natural is ...
- Software AG’s 4th-generation programming language
- for application development

Example for calling Adabas:

```
FIND PERSONNEL WITH NAME = 'KUESPERT'
   AND CITY = 'JENA'

DISPLAY
   'ID'       PERSONNEL-ID
   'Surname'  NAME
   'Location' CITY

END-FIND
```
Accessing Adabas – Natural II
Accessing Adabas — SQL Gateway

Adabas SQL Gateway

- Calling Adabas via an SQL interface

  ```sql
  SELECT  personnel_id, name, city,
  FROM    PERSONNEL t1, PERSONNEL_ADDRESS_LINE t2
  WHERE   name = 'Kuespert' AND city = 'Jena'
  AND t1.isn_personnel = t2.isn_personnel
  ```

- Executed by accessing only one Adabas file (table)
  - PERSONNEL_ADDRESS_LINE is periodic group in PERSONNEL file

- Embedded into host programming language

- Calling Adabas via an ODBC or JDBC interface
Accessing Adabas — SOA Gateway

Adabas SOA Gateway

- Calling Adabas via a request to a Web service URL
  

- Calling Adabas via a SOAP request

- Callable from programming languages, Web browser, Excel, Word, etc.
New Developments

- **Application**
  - Java
  - .NET
  - 3GL
  - Natural

- **Data Management**
  - SQL
  - Java
  - XML
  - SOA

- **Data Access**

- **Data Distribution**
  - VSAM
  - DL1
  - DB2
  - Apps

- **Tamino XML Server**

- **Text Retrieval**

- **Administration**
  - Security
    - External e.g. RACF
  - Monitoring
  - Statistics
  - Utilities

- **Performance**
  - Windows, Unix, Linux, Mainframe

- **Cluster**
  - • Availability
  - • Load Balancing

- **Data Organization**
  - • Partitioning
  - • Save/Restore

- **Data Distribution**
  - • Replication
  - • Transaction

- **Cross-platform**
New Developments — Event Replicator for Adabas (2005)

Replication of production data

Close to real time
From Adabas (source)
To ...
- Adabas (destination)
- relational databases
- data warehouses
- applications
New Developments — Data Archiving for Adabas (2010)

Archival of production data

Resource savings
- Automation
- Lower-cost hardware

Regulatory compliance
- Storage
- Robust search capabilities

Performance
- Relieve pressure on production database
New Developments — Data Masking for Adabas (2011)

Generation of Test data
- “production-like” data for ...
  - application testing
  - end user training
  - demonstration
  - research

Automated process
- Less resources to create test data

Regulatory compliance
- Privacy
- Security

Application quality
- Production versus generic data
Success Factors

Flexible data model
- Supports relational structures, entity-relationship model, others

Excellent performance — Key factor
- Helped by data model, record-level locking, exploitation of system functions
- Economical use of disk space via data compression, saving on I/O operations

High availability
- Little downtime for maintenance (planned outages)
- Quick & reliable recovery (unplanned outages)

Simple administration
- A few database administrators take care of many Adabas databases

Several platforms
- Present across mainframes and open systems
Adabas
A database for mission-critical applications