

Relational Database Interface for Existing HP e3000 Applications

Harry Jungk

Manager R&D

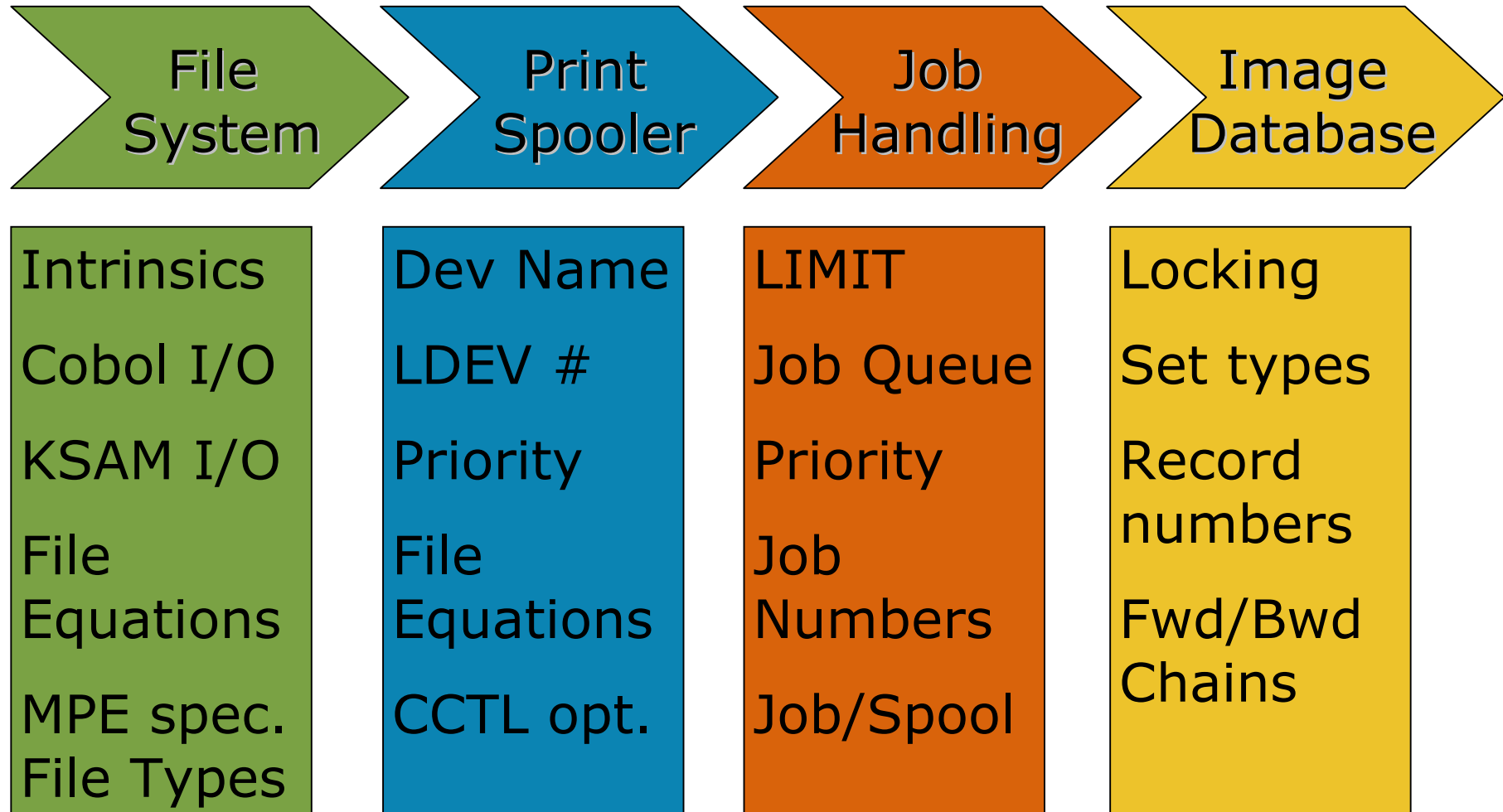
ORDAT Germany



The migration issue

- HP3000 based ERP-System
- Decision to get independent from MPE & Image in 1992
 - 5000+ applications
 - Millions lines HP-Cobol code
 - Installed base (100+ Customers)
- During migration
 - further on developing
 - cover all support requests (bug fixing)
- Performance
- Customers may won't pay for migration

The 4 Steps for Success



TurboImage Key Features

- DBLOCK features
 - Locking single records
 - Locking whole table or database
 - Locking with operators (<,>,<>, >=, etc)
- Record numbers
 - Direct reads by record number
 - Reading chains within detail datasets (fwd/backwd)
 - Keeping record numbers for reallocating chains
- Chain Handling
 - Sorted/unsorted chains
 - Multiple chains per detail

The TI vs. Oracle Pitfalls

■ Transactions in Oracle

- Cursors instead of TurboImage chains
 - Frozen result set after creating the cursor
 - Changes not recognized by the application
- Rollback segment usage
 - Snapshot too old during chained reads
 - No „dirty“ reads like in TurboImage possible
- Locking vs Transaction Handling
 - Locking is more serialization than transactioning
- Scrollable Cursors
 - No backward reads in Oracle prior to Oracle 9i.

Embedded SQL vs CLI

■ Embedded SQL

- String handling in Cobol is very complex
- Parameter passing is not simple emb. SQL
- Embedded SQL is not really database independent
- VARCHAR handling requires a lot of appl. changes
 - 01 MY-VARIABLE PIC X(50). changes to
 - 01 MY-VARIABLE.
 - 03 MY-VARIABLE-LEN PIC S9(4) COMP.
 - 03 MY-VARIABLE-ARR PIC X(50). and
 - All MOVEs have to contain the length field
 - Redesign of the entire application may be required

Embedded SQL vs CLI

- Call Level Interfaces
 - Allow TurboImage 'like' call structures
 - Create database independent call structures
 - Get best performance out of RDBMS
 - Be 100% TurboImage compatible
 - Create a compatibility layer
 - Create a performance layer for optimization
 - Requires minimum changes within the application
 - Allows a 100 % TurboImage compatible lock monitor

TI2SQL – an example for CLI

- One solution that solves all the problems
 - All Image data types can be supported
 - The Image lock features are completely covered
 - The master/detail relationship will be continued
 - Automatic master sets are supported
 - Chain handling, previous, next records is supported
 - Performance can be improved with minor changes
 - Database access is 100% in native mode
 - Full use of all RDBMS features – e.g. views - is possible
 - No changes within the application in compatibility level
 - Multiple database/application servers are supported
 - Multiple RDBMSs can be supported simultaneously
 - Only minor changes in performance level
 - All enhanced features are in an Image like shape
 - Embedded measurement interface for database access

TI2SQL example for CLIs

■ Turbolimage based

DO-THE-FIND.

```
MOVE „PART-NO“      TO KEY-ITEM.
MOVE „4711“          TO ARGUMENT.
CALL „DBFIND“ USING  BASE, DSET, MODE, STATUS,
                     KEY-ITEM, ARGUMENT.
```

LOOP.

```
CALL „DBGET“ USING  BASE, DSET, MODE, STATUS,
                     LIST, BUFFER, ARGMENT.
```

```
IF THE-STATUS > 10 OR < 20
  GO TO LOOP
```

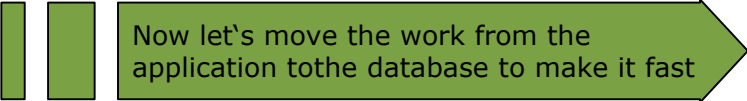
END-IF.

```
IF THE-USER IS NOT = „MGR“
  GO TO LOOP
```

END-IF.

```
NOW WE PROCESS THE RECORD.
GO TO LOOP.
```

THAT-S-IT.



Now let's move the work from the application to the database to make it fast

■ Performance level in TI2SQL

DO-THE-FIND.

```
MOVE „PART-NO“      TO KEY-ITEM.
MOVE „4711“          TO ARGUMENT.
```

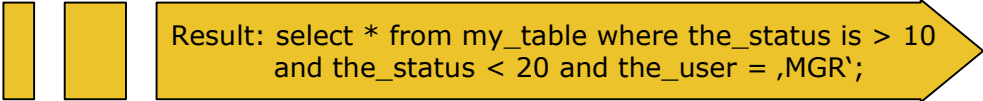
```
MOVE „THE-STATUS“   TO SEL-ITEM (1).
MOVE „>“             TO SEL-OP (1).
MOVE 10              TO SEL-VAL-INT16 (1).
MOVE „THE-STATUS“   TO SEL-ITEM (2).
MOVE „<“             TO SEL-OP (2).
MOVE 20              TO SEL-VAL-INT16 (2).
MOVE „THE-USER“      TO SEL-ITEM (3).
MOVE „=“             TO SEL-OP (3).
MOVE „MGR“           TO SEL-VAL-INT16 (3).
```

LOOP.

```
CALL „DBGETSEL“ USING BASE, DSET, MODE, STATUS,
                     LIST, BUFFER, ARGMENT,
                     HANDLE, SEL-BUFFER.
```

```
NOW WE PROCESS THE RECORD.
GO TO LOOP.
```

THAT-S-IT.



Result: select * from my_table where the_status is > 10 and the_status < 20 and the_user = „MGR“;

Questions?



Thank You
Very Much
For
Your
Attention



Interex, Encompass and HP bring you a powerful new HP World.

