ConditionsDB – Lisbon API

Wide access to CondDB data and schema

A real CondDB implementation
for the real world usage

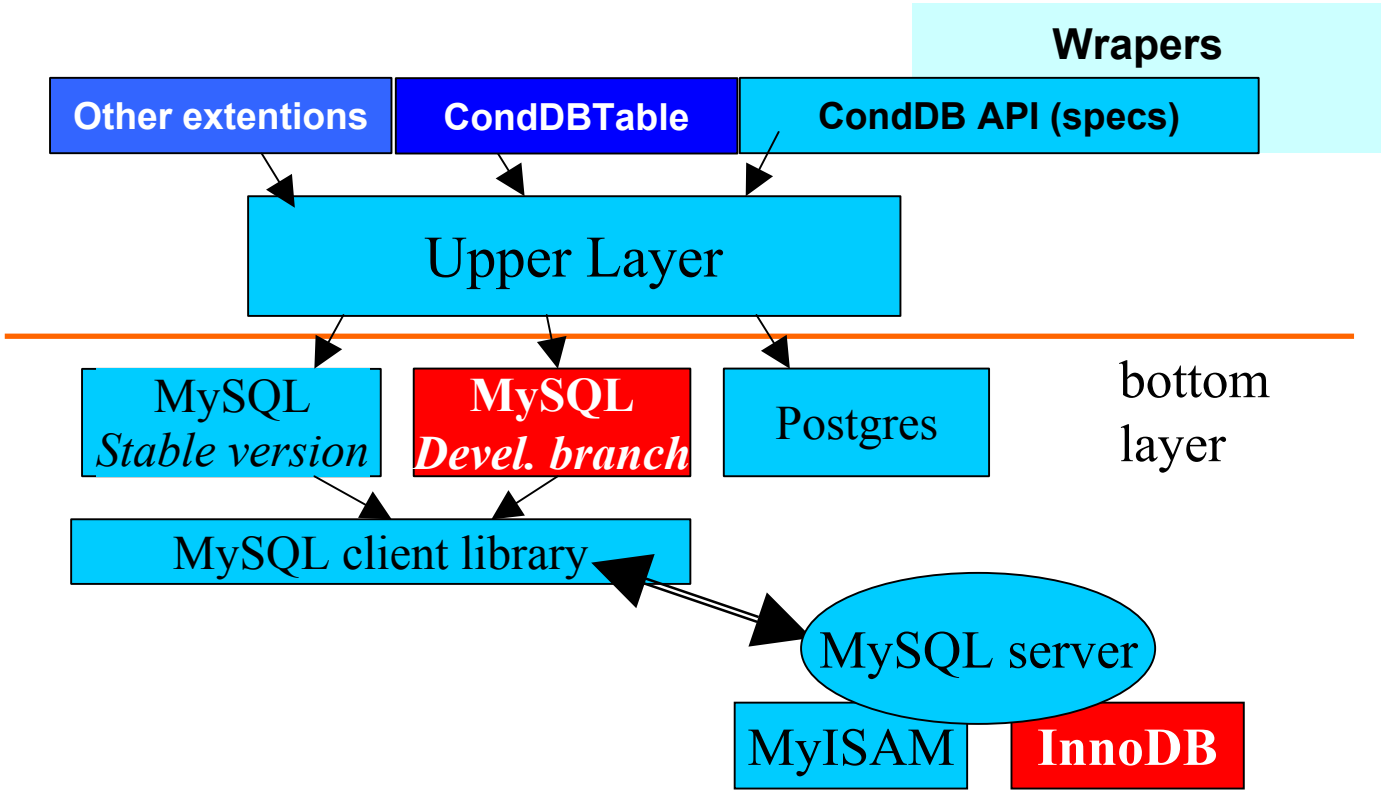
Far beyond the BLOBS…

LCG Conditions DB Workshop

Outline

- Overview
- Implementation evolution
  - Performance issues
- Implementation Status
  - The new CondDBTable object
- Interfaces
  - DCS
  - Online Software
  - POOL (Athena)
- Tools
  - Simple web browser
  - Examples and test programs
- Conclusions

ConditionsDB (mySQL) implementation



Overview

- **Some Features**
  - **Backend independency**
  - **Data volume Scalability**
    - We have made it federated by design
  - **Trying to minimize dependencies on external packages**
  - **One slide installation approach**

Implementation evolution

- **First delivered implementation (0.2.6x):**
  - Finished August year ago
  - Based on the ORACLE implementation (IT)
  - Uses the same standard (IT) interface
  - Standard SQL without transactions
  - MySQL 3.23.xx as RDBMS
  - Very fast and promising

	ORACLE	mySQL
Read 10,000 simple Objects	4m 37.32s	0m 7.93s

Implementation evolution

- **The extended version (0.3.x)**
  - **Was delivered 8 months ago**
  - **Was used in Muon (and TileCal) 2002 test beams**
  - **It's available in the ATLAS and Lisbon repositories**
  - **Supports the same interface and also:**
    - Tiny objects storage and retrieval
      - Objects that do not need versions
      - New functions implemented and added to the interface
  - **Plus:**
    - PVSS manager to interface with DCS
    - Available on windows platforms
  - **DB schema modified - for better performance**
  - **Standard SQL without transactions**
  - **MySQL 3.23 or 4.0 as RDBMS**

Implementation evolution

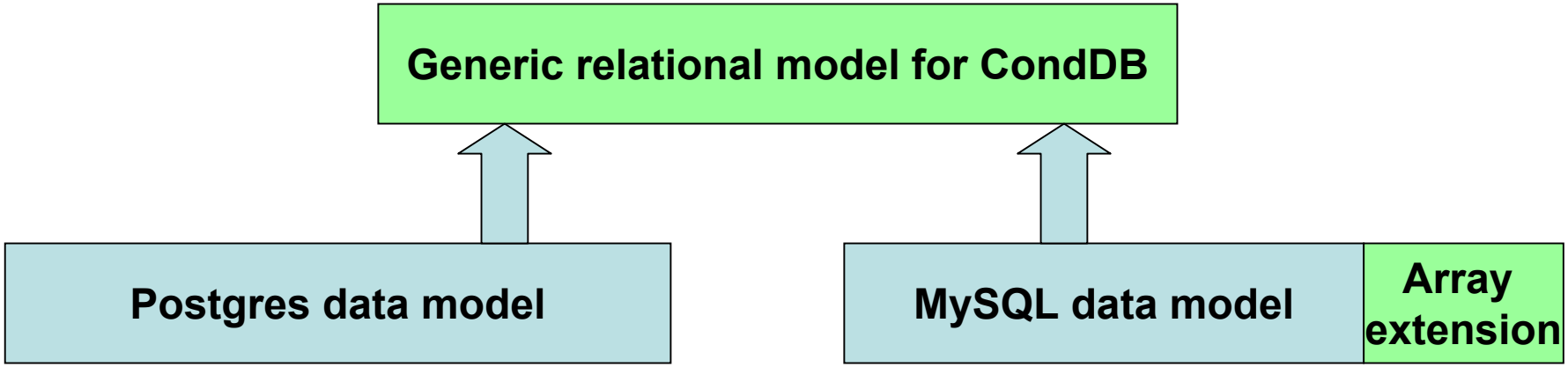
The extended version (cont)

- **New building approach**
  - Using standard GNU approach → *./configure, make, make install*
- **Extensions for the "tiny" interface**
  - Full support for objects with structures (including arrays)
  - Single/Field arrays
  - New DB schema
- **Logging mechanism for debug purposes**
  - Stores API actions in one table

	Single / Multi-line arrays	Single / Field arrays	Structures (incl. arrays)
PVSS	√	X	X
PVSS Online extended	√	√	√

Implementation evolution

- **The version 0.3.5**
  - **Introduces binary array support extending MySQL**



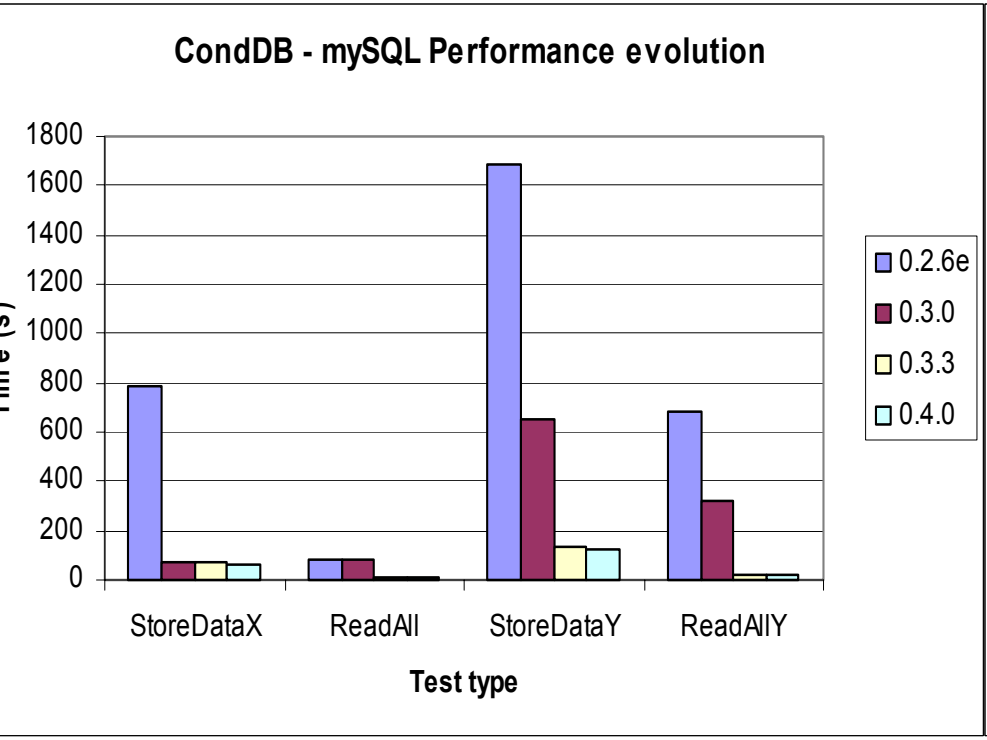
  - **It also allows several “columns”**

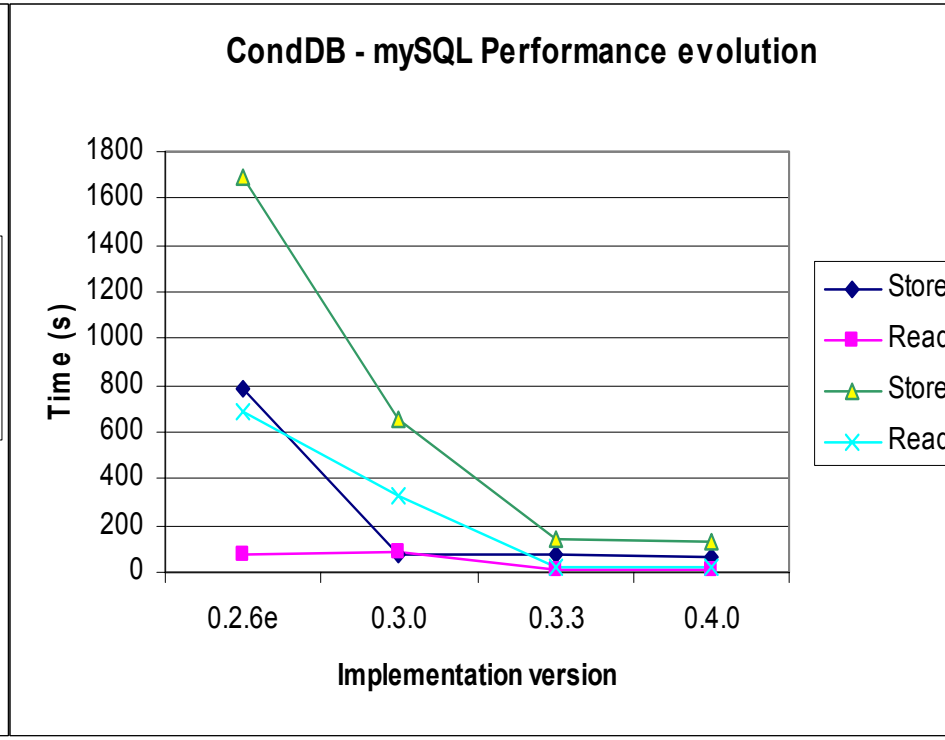
Implementation evolution

- **0.4.x version**
  - **Beta version available**
    - http://kdata5.fis.fc.ul.pt/cgi-bin/cvs/viewcvs.cgi/ConditionsDB/
    - **Call for beta-testers:**
      - **Download it from the site**
      - **Type: ./configure, make, make install**
    - **Final (test-beam) version will be released when the DCS and Online SW interfaces are ready**
    - **Some (minor) things might yet change**
  - **Access to the data using BLOBS and CondDBTable**

Implementation evolution - performance

- **Performance comparison (10.000 objects)**
  - **Standard (BLOB) interface – PIV 2,6 GHz, 1GB RAM, gcc 3.3**





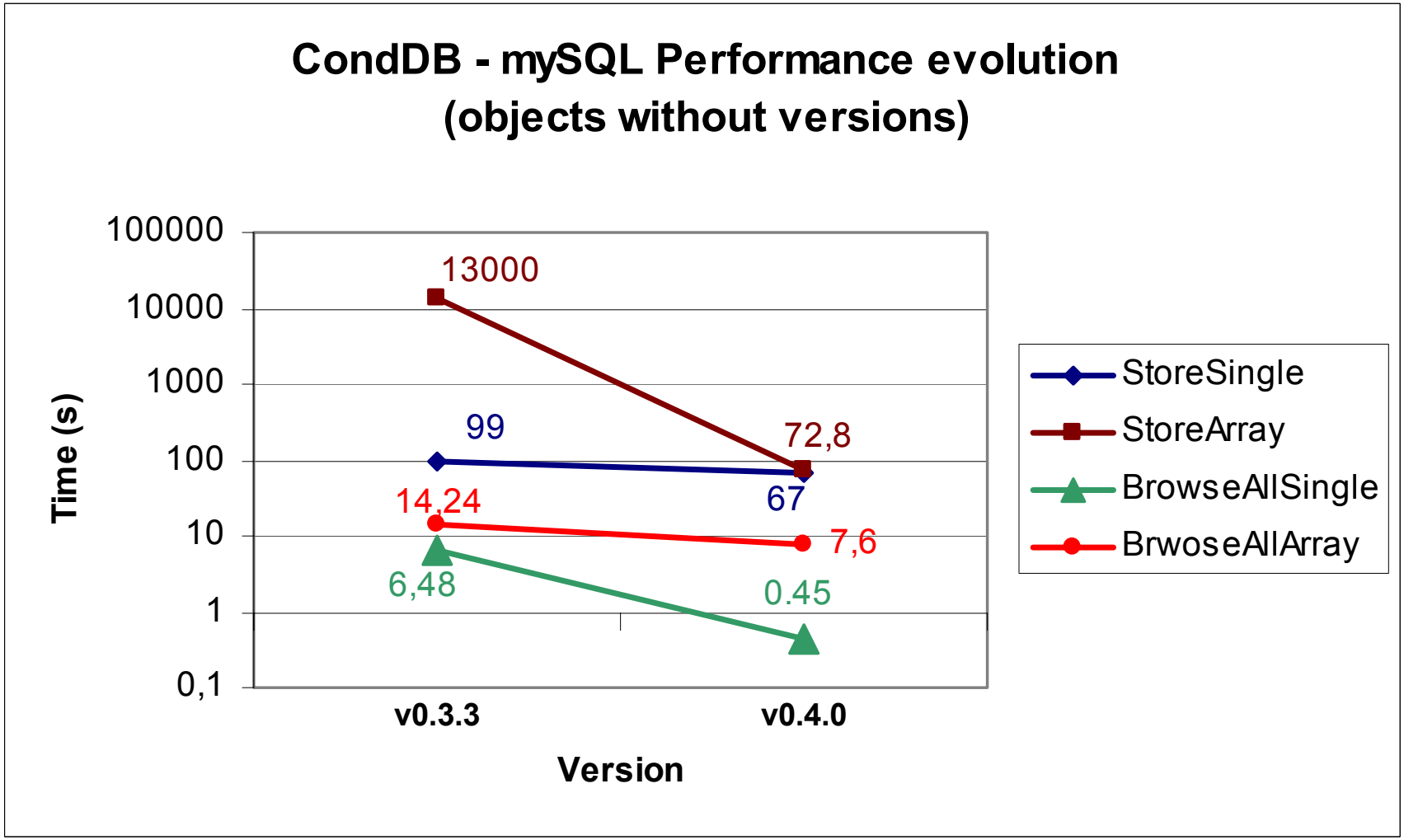
Approach: Prepare to give up the first one.

Conclusion: Factor 10 better; We had to do prototyping because the pure design and discussion was leading nowhere.

Implementation evolution - performance

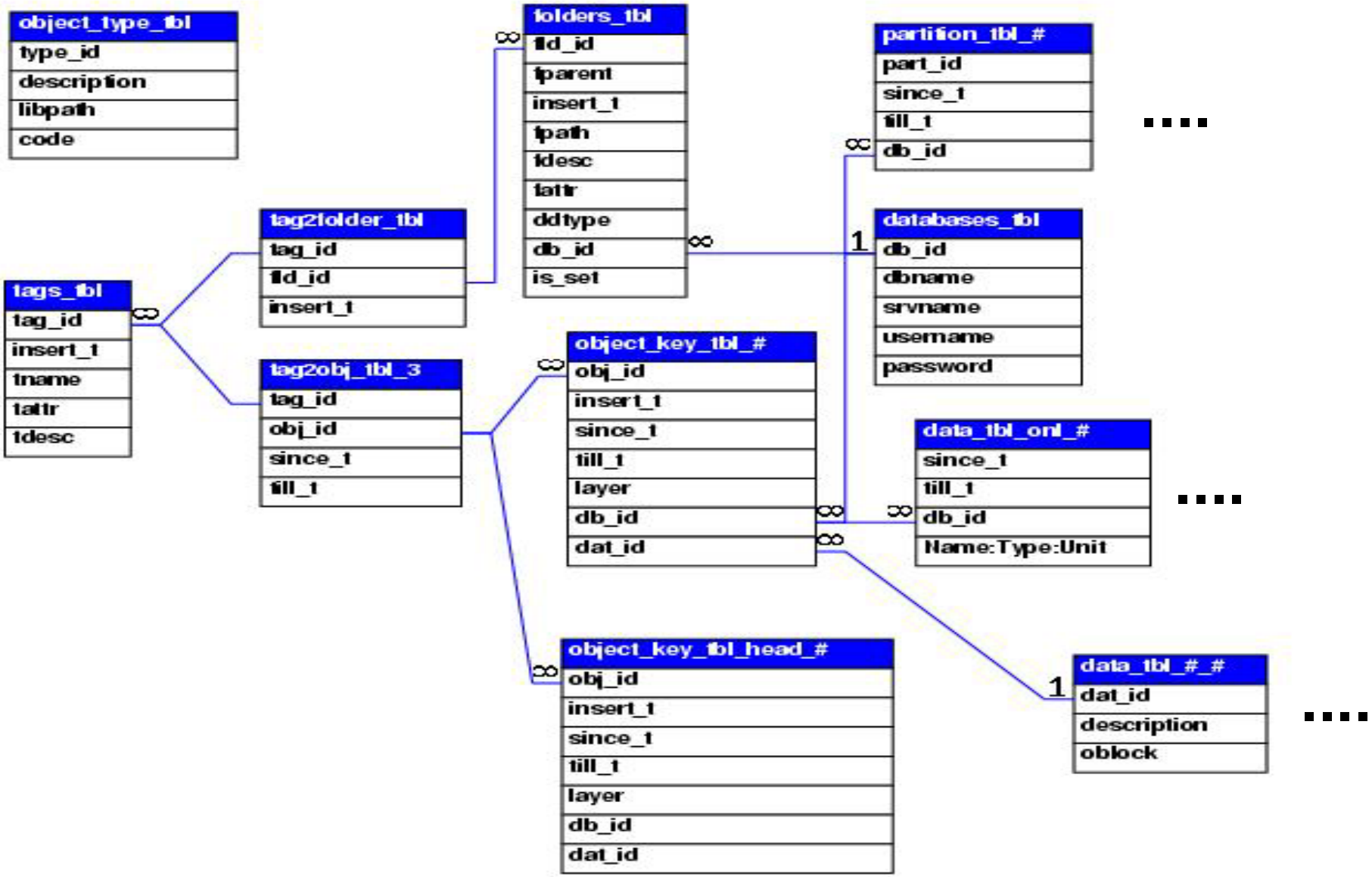
Performance comparison

- Online interface – without versions Athlon 2GHz, 1GB RAM, gcc 3.3
- Test for 50.000 objects; Arrays of 20 Floats



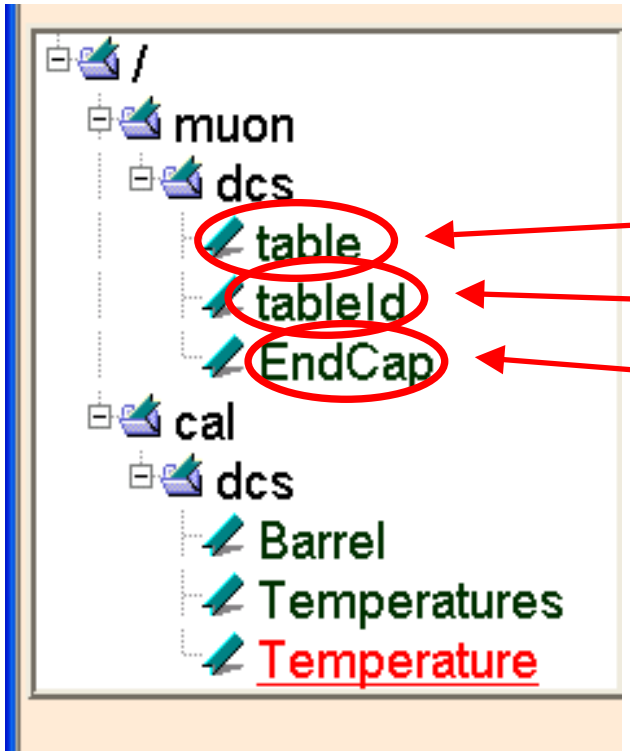
Implementation status

- **0.4.x version**



Implementation status

- **0.4.x version**
  - **7 types of data storage:**
    - BLOBS (with / without versioning)
    - CondDBTable (with / without versioning)
    - CondDBTable with Id (with / without versioning)
    - POOL



Each folder contains an object of a given type

CondDBTable

CondDBTableID

BLOB

Implementation Status

- **Whats new in the API**
  - Changes to the interface

Overload for CondDBTable: storeCondDBObject
findCondDBObject
createCondDBFolder

New functions:
browseHistory
getFolderType

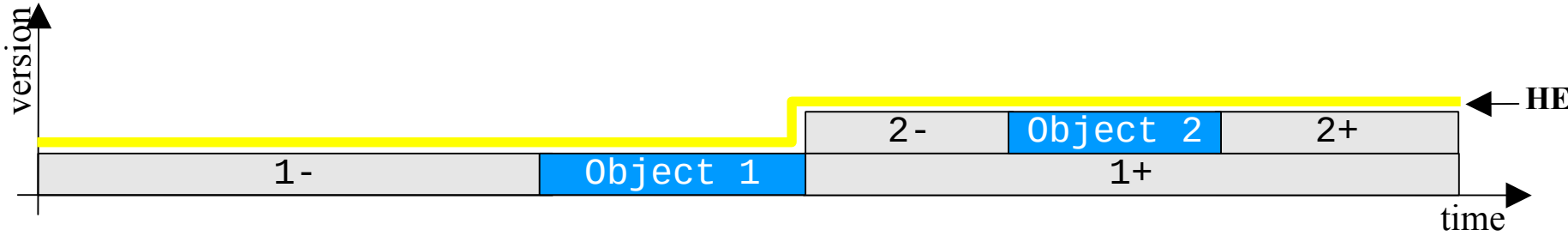
New class:
ICondDBTable

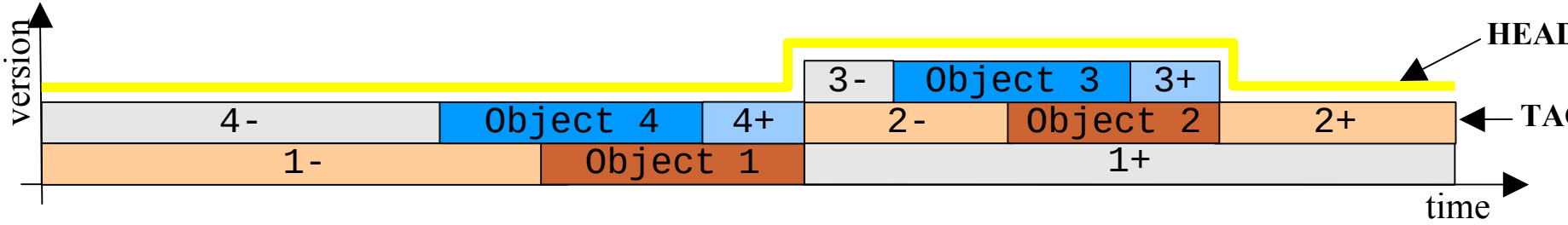
Removed:

all functions and methods which are PVSS specific

Implementation Status

- **Still limited tagging mechanism**





Browse objects in HEAD

Browse objects in tag TAG1

The CondDBTable

What is the CondDBTable

- **The CondDBTable is a transient object. It is possible to update it in memory and then commit it to the database (the CondDB persistent storage facility).**
  - the table structure is easily mapped from memory to database
- **Database know about the object structure**
- **Restricted Transient model to support the Relational model**
  - Exploring the “Relational” tuple
  - DBMS independent

The CondDBTable

- **What is the CondDBTable**
  - **Basic CondDBTable functions**

setName	getNames
setType	getTypes
getRow	
setCell	getCell
setColumndata	
getNumRows	
getNumColumns	
setSinceTime	setTillTime
changeSinceTime	changeTillTime
getSinceTime	getTillTime

The CondDBTable

- **How to map blobls (xml type) into a CondDBTable**

vers	name	Jff	Jzz	Job	x	y	z	thx	thy	thz
<block>										

```
<format> %s %s %d %d %d %f %f %f %f %f %f </format>
<names> vers name Jff Jzz Job x y z thx thy thz </names>
<size> 6 </size>
<r> A BIL  3 -1  0     0.395    1.612   -0.973  0.001787  0.000038 -0.000346 </r>
<r> A BIL  3 -2  0     1.068    1.780   -1.005  0.003966  0.000969  0.000219 </r>
<r> A BML  3 -1  0     1.162    7.637    1.866  0.001507  0.000204  0.000100 </r
<r> A BML  3 -2  0     1.002    7.511    4.831  0.001779 -0.000156  0.000196 </r
<r> A BOL  3 -1  0     1.867    9.902    5.087  0.000749  0.001143  0.000255 </r
<r> A BOL  3 -2  0     0.877   13.717    8.736  0.001523  0.000962 -0.000648 </r
</block> |
```

CondDBTable with ID

t1	t2	Id	Column A (Type x)	Column B (Type y)	……. …….	Column Z (Type z)
1	Inf	1				
1	10	2	Array	Object	Array/Object	Array
1	Inf	….				
1	Inf	N				
10	Inf	2				

CondDBTable

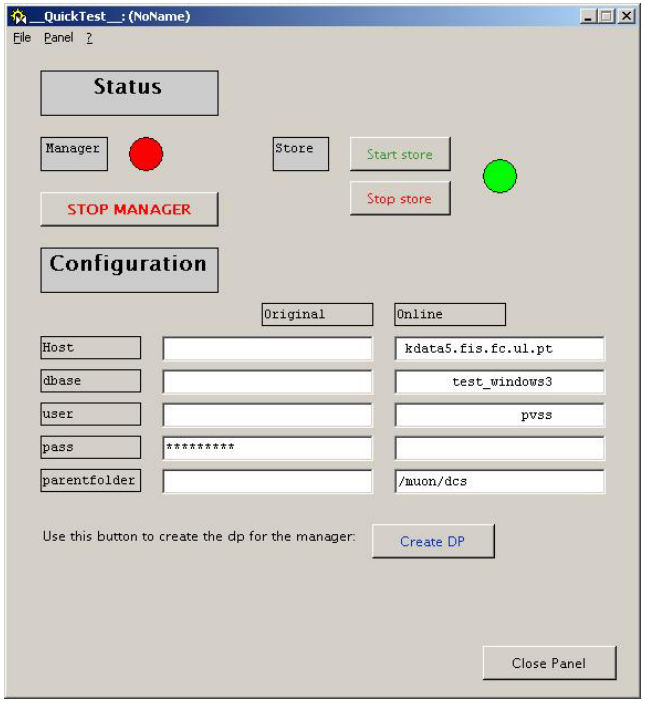
- **Some performance values**
  - **CondDBTable 4 Columns (single objects) 10.000 values**
    - **Store – 13s**
    - **Read – 0,4 s**
  - **CondDBTable 8 Columns (4 single + 4 array [20]) 10.000 values**
    - **Store – 21 s**
    - **Read – 5 s**
  - **CondDBTableID 4 Columns (single objects) 5 ID, 10.000 values each**
    - **Store – 54 s**
    - **Read – 2 s**

Interfaces - DCS

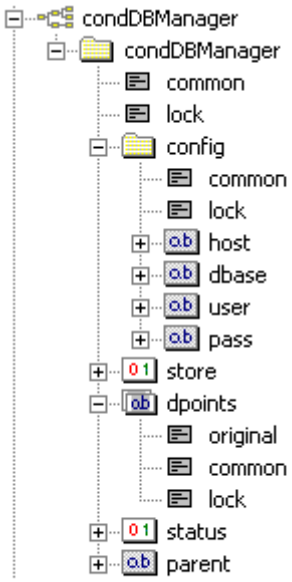
PVSS API Manager

- A new manager which uses the CondDBTable is available. This manager will be able to store structures and structures of arrays.
- PVSS 3 compatible (only after PVSS 3 candidate release)

Configuration panel



dpStruct

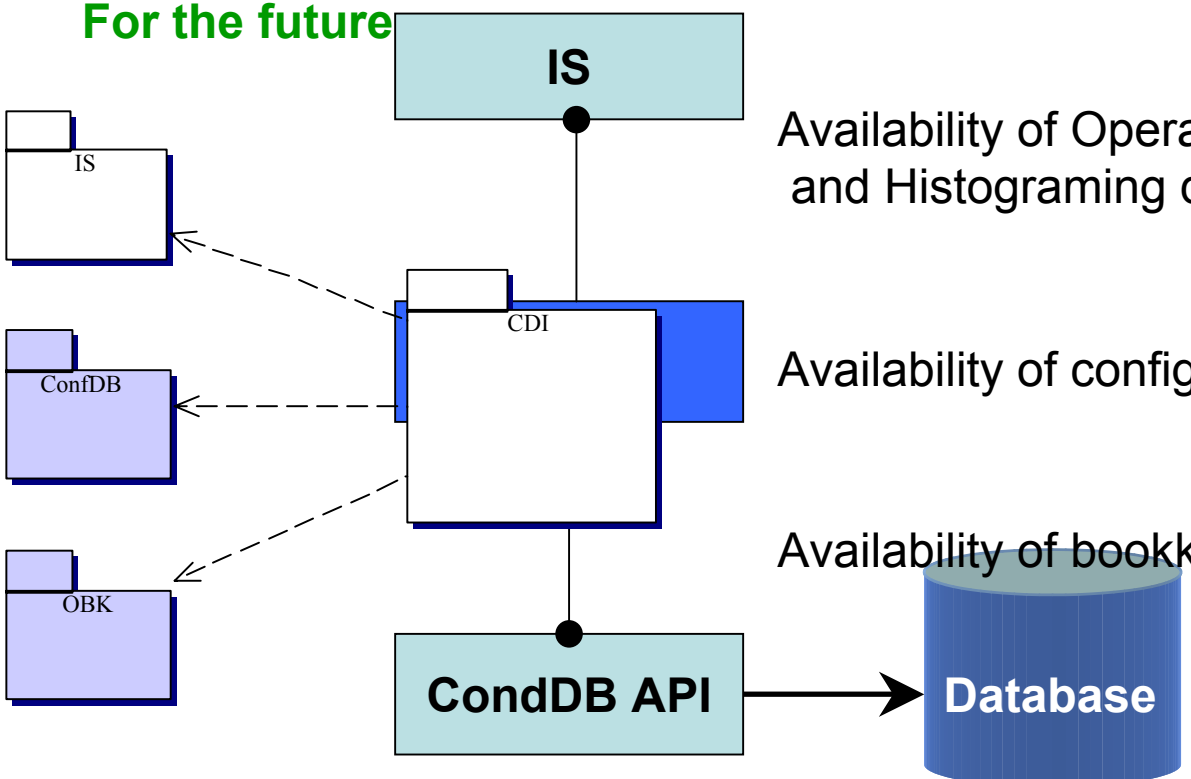


Interfaces – ATLAS Online Software

- **Online CondDB interface (CDI) – ATLAS specific**
  - **This interface stands for the TDAQ as the PVSS manager stands for DCS.**
  - **The new CondDB version will be able to cope with data coming from the IS system**

For the next test-beam

For the future



Availability of Operational monitoring and Histograming data

Availability of configuration data

Availability of bookkeeping information

Interfaces - POOL

- **Major issue**
  - It will allow interface CondDB and POOL/Athena
  - Discussion with experts on-going
  - Test beam real issue
    - CondDBTable objects needed inside ATHENA
  - First example program should be available at the beginning of May…

Tools - Simple web browser

- **CondDBrowser**
  - **Read only CondDB web browser**
  - **Uses the C++ API**
  - **PHP bind to the C++ library**
  - **Will be installed in the next test-beam offline server**
  - **Simple display facilities**
    - In the future it will have the functionalities to display histograms – (using ROOT?)

Tools – Simple web browser

- **CondDBrowser**

CondDBrowser

/
muon
cal
dcs
Barrel
Temperatures
Temperature

Host: kdata5.fis.fc.ul.pt
Database: condDB_V4_1
User: lpedro

Selected folder: /cal/dcs/Temperature

Browse with tag | Browse at point | Browse history

Tag: HEAD

Point: yyyy mm dd hh mm ss

Since: yyyy mm dd hh mm ss

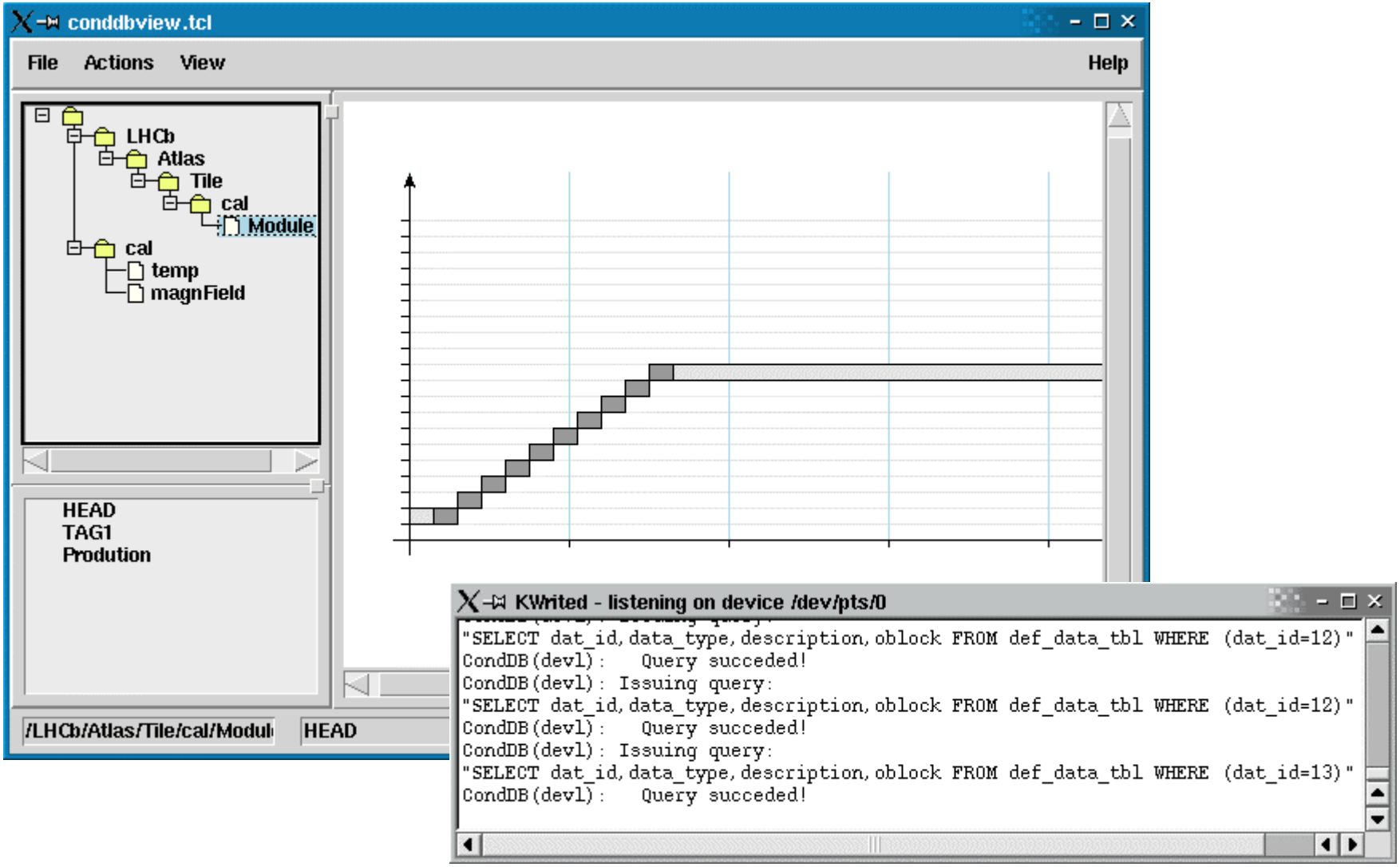
Till: yyyy mm dd hh mm ss

Submit

Since:	Till:	Insertion Time:	Layer:	Data:	Description:
1	0	2003-Nov-19 23:52:32.0 (GMT)	1		Empty object
2	1	2003-Nov-19 23:52:32.0 (GMT)	2	sample data value object	sample description
3	2	2003-Nov-19 23:52:32.0 (GMT)	3	sample data value object	sample description
4	3	2003-Nov-19 23:52:32.0 (GMT)	4	sample data value object	sample description

Tools – Simple web browser

- **TCL/TK browser**



Tools – example programs

- **Store**
  - storeDatax
  - storeDatay
  - storeTable
- **Read**
  - readData
  - readTable
- **DB initializiation**
  - basicSession
- **Tags**
  - createTags
  - testTags
- **Folders**
  - createFolderx
- **Performance**
  - performanceTests

Future developments

- **Interface**
  - **Understand the tags problem**
  - **ConditionsDB -> Condbtable -> User class include file**
  - **Stand alone browser with Administrative tools**
    - Partitioning
    - Other
  - **Migrate to the LCG CVS repository**
  - **Develop a set of common tools/tests to the ORACLE implementation**
  -

Conclusions

How to get this implementation

- **export CVSROOT=:ext:conddb@kdataserv.fis.fc.ul.pt:/usr/local/cvsroot**
- **export CVS_RSH="ssh"**
- **cvs co ConditionsDB** (password conditions)
- **WEB cvs access**
  - http://kdata5.fis.fc.ul.pt/cgi-bin/cvs/viewcvs.cgi

- **Build it**
  - ./configure --with-mysql-lib=~/mysql_lib_dir --with-mysql-inc= /mysql_include_dir/ --withconddbprofile=mysql_host:test_mydb
  - make
  - make install

- **Try it!**

Conclusions

- **What do we have:**
  - **An API that knows how to read the structure of the objects stored in the Database**
  - **Database partitioning from the API**
  - **Able to cope with object schema evolution**
  - **Restricted Transient model to support the Relational model**
    - Exploring the “Relational” tuple
    - DBMS independent
    - **CondDBTables can be used to define data sets that can be accessed as any other relational table (using keys/relations if wanted/needed…….)**
  - **Interface between DCS and ConditionsDB**
  - **Simple web browser (first version will be available really soon!)**

This is much more than a BLOB storage mechanism

Conclusions

- **What do we have:**
  - An interface ready for the real data acquisition
  - API will be used and TESTED in next ATLAS combined test beam
  - It is very important to get to a new standard interface. We are very much looking forward to any discussion that will help in this task – including adding our and more extensions.
  - What more functionalities are needed?

This is a REAL API implementation for the real World usage

The end

Questions... Comments