OBK – An Online High Energy Physics’ Meta-Data Repository

List of authors: Dr. I.Alexandrov, Dr. A.Amorim, Ms. E.Badescu, Ms. M.Barczyk, Ms. D.Burckhart-Chrome, Dr. M.Caprini, Dr. M.Dobson, Dr. J.Flammer, Mr. R.Hart, Dr. R.Jones, Mr. A.Kazarov, Mr. S.Kolos, Dr. V.Kotov, Dr. D.Liko, Mr. L.Lucio, Dr. L.Mapelli, Mr. M.Mineev, Dr. L.Moneta, Dr. I.Papadopoulos, Ms. M.Nassiakou, Dr. N.Parrington, Mr. L.Pedro, Mr. A.Ribeiro, Dr. Yu.Ryabov, Mr. D.Schweiger, Mr. I.Soloviev, Dr. H.Wolters

Presentation by: Levi Lúcio
Introduction (1) - CERN

- Founded in 1954, CERN (European Organization for Nuclear Research) is a wide international collaboration (80 nationalities);

- The objective of CERN is the experimental study of physics, in particular the study of matter and the forces that hold it together;

- Within CERN’s lifetime, several important physics discoveries have been made, along with technology breakthroughs such as the WWW.
The LHC (Large Hadron Collider) accelerator is now being built at CERN to be ready in 2007. It will be the **most powerful particle accelerator** in the world and will allow **breaking new barriers** in HEP (High Energy Physics):
Introduction (3) - Detectors

- Along the accelerator ring, several detectors (4) will be put in place. The ATLAS (A Toroidal LHC ApparatuS) is one of them:
Introduction (4) - Physics

- Two particle beams travelling in the accelerator in opposite senses at 99.9999997% of the light speed meet head on in the detector, producing **new particles**;

- The interaction (collision) of two particles and their final state products is called an **event**;

- For ATLAS, **many events** need to be collected to have strong statistics that prove the theory - a very rare particle (Higgs boson) is searched for.
Introduction (5) - Triggers

- The rate of events at ATLAS will be extremely high - 40 MHz;

- Only a fraction of those events \((1/10^7)\) is interesting - a powerful filter (trigger) is necessary;

- This still means 100 events of 1Mbyte each per second - 100MByte/s storage;

- The ATLAS is expected to produce 1PByte/year of event data.

Levi Lucio - CERN EP/ATD, FCUL
Introduction (6) - OBK

Online Book-keeper

- Part of the **Online Software** system - online *control*, *configuration* and *monitoring* of the detector and triggers (thousands of machines);

- **Records** and manages log data (**meta-data**) about the detector and trigger chain (diversified information);

- Project undertaken in 1996 by the Lisbon FCUL / ATLAS group - L.Lucio, L.Pedro, A.Amorim, A.Ribeiro

Levi Lucio - CERN EP/ATD, FCUL
Databases in HEP (1) - History

- **Before the 1980s** - database market **not mature** to handle size and complexity; **in-house solutions** in FORTRAN;

- **1980s** - relational solutions to handle book-keeping data; interest in OO persistent data model;

- **1990s** - standardization of OO databases (ODMG); investigation and consequent usage of **commercial** Objectivity/DB by LHC and other HEP experiments;

- **2002** - LHC experiments **dropped Objectivity/DB** and are **searching for alternatives** - Oracle 9i, homegrown ROOT?
Databases in HEP (2)

Today’s needs

- Management of large amounts of data (petabytes);
- Support of addition of significative quantities of data on a daily basis;
- Support of simultaneous queries;
- Support of data access over international networks;
- Flexible data model supporting versioning and schema evolution;
- Adequate interfacing to tertiary storage.
Databases in HEP (3)

Today’s trends

- Indecision between **homegrown** (OO ROOT) or **external** (OR Oracle 9i) databases;

- Not clear what **data model** to use (pure **OO, Object-Relational**?);

- Heavy research on **data distribution - replication**, interfacing with **GRID**;

Levi Lucio - CERN EP/ATD, FCUL
The OBK (1) - Definition

- Defined in the ATLAS technical proposal as the component that "archives information about the data recorded to permanent storage by the data acquisition system. It records the information to be later used during data analysis\(^1\) on a per-run\(^2\) basis (run cataloger). It provides interfaces for retrieving and updating the information."

\(^1\)After being collected, event data is analyzed “manually”.  
\(^2\)A data taking period with a given machine parameterization.
**The OBK (2)**

**Development approach**

- **Prototypical spiral (3 prototypes - OBK/Objectivity, OBK/OKS and OBK/MySQL);**

- **Well defined software development process + documentation production;**

- **Usage of development support tools:** CVS, CMT (platform management), Perl, Rose, documentation templates, etc.

---

Levi Lucio - CERN EP/ATD, FCUL
The OBK is part of the Databases super-component of the Online Software:
Requirements gathering
Main Use Cases

- **Data acquisition:**
  After being started with the Online Software, the OBK will acquire
  the specified data in an *automatically* without human intervention;

- **Information updating:**
  Users will want to add their own annotations to the acquired data;

- **Data access:**
  It will be possible for several kinds of clients, such as *humans*,
  applications or offline data analysis frameworks to access the database adequately;

- **Data administration:**
  Users will want to manage and administrate the OBK database.
High level design (1)

Package overview

- **IS** (Information System)
- **MRS** (Message Reporting System)
- **ConfDB** (Configuration Databases)

Online Software

- **IS**
- **MRS**
- **ConfDB**

OBK acquisition software

DBMS

C++ API

Web Browser

Administrative tools

Levi Lucio - CERN EP/ATD, FCUL
High level design (2)

Logical database structure

Partition: subset of the detector and triggers that can acquire data independently.

Levi Lucio - CERN EP/ATD, FCUL
Implementation

Languages and tools

- **C++ programming language**
  Used to code all OBK acquisition engines (including connections to the DBMSs) and API software;

- **STL (Standard Template Library)**
  Data containers and algorithm templates used as building blocks for C++ applications;

- **Objectivity/DB**
  Commercial distributed object oriented database management system;

- **OKS**
  In-memory persistent object manager implemented in-house to satisfy ATLAS’ needs in terms of configuration databases;

- **MySQL**
  Open source relational database management system;

- **PHP**
  General purpose scripting language, specially adequate for web programming;

- **Perl**
  General purpose scripting language;

- **Apache**
  Widely used HTTP server.

Levi Lucio - CERN EP/ATD, FCUL
Implementation

Objectivity prototype(1)

```cpp
class OBKRun : public ooContObj {
public:
    Run ();
    Run (uint32 runNumb);
    void setRunNumb (uint32 runNumb);
    uint32 getRunNumb (){
        ooRef(OBKComment) runComms[] <-> commToRun[];
        ooRef(Coordinator) runCoordinator <-> rCoordinated[];
        ooRef(LockedStatus) runToLStat[] <-> lStatOfRun[];
        ooRef(OBKConfdb) hasConfig <-> appliesToRuns[];
    }
protected:
    uint32 m_runNumb;
    d_Timestamp  m_startDate;
    d_Timestamp  m_endDate;
};
```
Implementation

Objectivity prototype(2)

Comments

▲ Objectivity/DB makes available \textit{specialized engines} to handle connections and concurrency;

▲ Very good integration between code and DMBS - \textit{minimal difference} between \textit{persistent} and \textit{transient} objects;

▲ The prototype makes use of Objectivity/DB \textit{transactions}. A new \textit{transaction} is started for each \textit{new run};

▲ Objectivity/DB’s \textit{locking mechanism} is used explicitly in the code to avoid incoherent reads/writes. \textit{MROW} (Multiple Readers One Writer) facility used to read data as soon as it is written.
Implementation

OKS prototype(1)

- Data model includes XML **data files** and **objects**;
- A data file is either **in-memory** or in **disk** (atomic loads);
- Database schema equivalent to OBK/Objectivity’s one.

![Diagram of database schema]

**Definition of a new OKS “object”**

```cpp
OksClass *PartitionInfo = new OksClass("PartitionInfo", false);
{
    OksAttribute *partitionName = new OksAttribute("partitionName",
        OksAttribute::string_type, false,
        "unknown",
        true);
    PartitionInfo->add(partitionName);
    PartitionInfo->add(inUse);
}
```
Implementation

OKS prototype(2)

Comments

▶ OKS is a **persistency C++ library**. No services other than the ones included in the library are made available;

▶ **No concurrency management** is available. In the OBK case concurrency was implemented using OS mechanisms;

▶ **No transactions** are **available**. At the beginning of each run new data files are opened and at the end of the run closed;

▶ The prototype includes **optimized accesses** to certain parameters which are very requested. They are kept in a special central data file (**cache**).
Implementation

MySQL prototype(1)

- Relational model: completely different database schema from previous OO approaches.
Implementation
MySQL prototype(2)

Comments

As Objectivity/DB, MySQL also makes available an engine to deal with queries;

Concurrency issues are managed transparently by the MySQL engine;

Transactions and atomic operations are made available by the MySQL engine - not used by the OBK though;

Indexes on certain key tables were created to accelerate queries (up to a factor of 45 speed difference);

XML used to deal with the difficulty of storing collection types.
Implementation - Data Access

- **Command line dump**
  Debug situations, not many available resources;

- **C++ API**
  Shared library; uses STL for return structures;

- **Web-based browser**
  More sophisticated, includes administrative tools. Heavier on resources than previous solutions.
Performance & Scalability (1)

Time to store an MRS start of run message

While for the OBK/Objy and the OBK/OKS store times rise (check if the run already exists), the OBK/MySQL presents low and constant store times.

Levi Lucio - CERN EP/ATD, FCUL
The OBK/OKS is the fastest - the operation takes place in memory, no I/O accesses.
While being accessed simultaneously by multiple IS servers the OBK/OKS presents the best performance - fastest IS storing time. Also, the Online Software is affected by OBK’s performance;

Worst performance in storage space by OBK/Objy - a container always allocates a predefined number of fixed size pages, even if they are not used.
Best results by OBK/MySQL, due to the efficiency of the MySQL query engine, faster than hand-coded queries in the OO prototypes;

In query 2 the OBK/Objy presents the worse performance - the parameters which are searched for are cached in the case of OBK/Objy and OBK/OKS;
Performance & Scalability

Overall Results

- **Best overall** results by the MySQL OBK prototype;
- **Strong results** also from the OKS OBK prototype, mainly due to its in-memory features;
- **Less optimal** results achieved by the Objectivity/DB prototype - requires deep know-how to be properly tuned.
Deployment

- Large scale tests of the Online Software (simulated environment):
  - 2001 (OBK/OKS): 111 nodes running on 111 machines;
  - 2002 (OBK/MySQL): 210 nodes running on 210 machines.

- Testbeams (real data acquisition with parts of the detector running):
  - 2000 (OBK/Objy): 2 Gbytes acquired;
  - 2001 (OBK/OKS): 3 Gbytes acquired;

Levi Lucio - CERN EP/ATD, FCUL
Some metrics

Requirements gathering: 2 man/month, 2 documents produced.

<table>
<thead>
<tr>
<th></th>
<th>Effort (man/month)</th>
<th>Lines of Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBK/Objy</td>
<td>18</td>
<td>4166</td>
</tr>
<tr>
<td>OBK/OKS</td>
<td>4</td>
<td>8799</td>
</tr>
<tr>
<td>OBK/MySQL</td>
<td>3</td>
<td>6193</td>
</tr>
</tbody>
</table>

Testing

<table>
<thead>
<tr>
<th></th>
<th>Effort (man/month)</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBK/Objy</td>
<td>1</td>
</tr>
<tr>
<td>OBK/OKS</td>
<td>1</td>
</tr>
<tr>
<td>OBK/MySQL</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Documentation: 3 man/month, User & Developer’s manual, Test report.
Lessons learnt

- **Software Development Process**
  Following a formal approach to the development of the three prototypes yielded:
  - easy comparison of the OBKs;
  - diminishment of the effort to build the latter prototypes;
  - delivery of a quality OBK product;

- **Technology**
  OO DBMS technology is very flexible in terms of data mapping and provides natural integration with programming languages. It is possible to follow an OO development approach both for application and database. RDBMS technology is less elegant but very efficient…

- **Interaction with users**
  Good and constant interaction with the final users of the system makes development simpler and faster. Continuous enhancement of the knowledge about the systems and the people the software interacts with is essential while putting the problem under perspective.