Available on CMS information server

CMS NOTE 2004/026

The Compact Muon Solenoid Experiment

CMS Note

Mailing address: CMS CERN, CH-1211 GENEVA 23, Switzerland

29 October 2004

Status and Perspectives of Detector Databases in the CMS Experiment at the LHC

A.T.M. Aerts

Eindhoven University of Technology, The Netherlands

F. Glege, M.Liendl

CERN, Geneva, Switzerland

I.Vorobiev

Carnegie Mellon University, Pittsburgh PA, USA

I.M. Willers

CERN, Geneva, Switzerland

S. Wynhoff

Princeton University, Princeton NJ, USA

Abstract

This note gives an overview at a high conceptual level of the various databases that capture the information concerning the CMS detector. The detector domain has been split up into four, partly overlapping parts that cover phases in the detector life cycle: construction, integration, configuration and condition, and a geometry part that is common to all phases. The discussion addresses the specific content and usage of each part, and further requirements, dependencies and interfaces.

Table of Contents

1. Introduction

The performance of accurate measurements at the high collision rate of the LHC collider with its correspondingly high data volume is clearly a challenging task [TDR2]. An important factor in the correct interpretation of the measurement data is accurate knowledge of the detector itself. For example what are the properties of the detector "in operation"? These properties may change during data taking: the detector may heat up and expand, the sensitivity of some of its components may deteriorate because of radiation damage, and occasionally a component may even fail completely. The change in properties has to be recorded so that the information is available for immediate action and so that it can be taken into account, when the measurement data are interpreted.

The view taken above on the information about the CMS detector is only one from a number of viewpoints, such as geometry, construction and integration. Each viewpoint focuses on a particular aspect of the detector and the corresponding description has to be available for the entire lifetime of the detector and beyond. CMS activities, in various phases of the CMS detector life cycle, rely on appropriate descriptions of the detector and its properties. It is important that a persistent set of descriptions of the CMS detector exist. These descriptions must be both mutually consistent, and individually designed to support a well-defined set of activities. The purpose of this CMS Note is to give an overview of the requirements, dependencies, and interfaces of the various views on CMS detector data. We will take as a point of view the existence of one conceptual model, and discuss the various descriptions in this context. In practice, each description will correspond usually to its own set of databases. As a consequence, these databases will be mutually dependent.

The combined descriptions should contain all the relevant information about the detector and document the mutual dependencies between the individual descriptions as well. We should distinguish here between the specification of the information (for example, in the form of data or object models) and the information itself that should conform to this specification. It is clear that the models should be available before the information is stored.

The set of descriptions is at present not complete. For one reason this is due to the fact that the detector is still evolving and has not reached its final state of completion. For example, there is a description of the detector geometry, but there is no description of the detector calibrations. For another reason some early versions of parts of descriptions may exist, but because of the highly distributed nature of the CMS collaboration, where a lot of work is carried out in relative isolation, they have not been brought forward yet.

This note provides a framework in which every set of detector data can be put into context and related to other data sets. In this way it makes it clear on which points data sets are overlapping, conflicting or absent.

The database descriptions should provide answers to questions about the detector, such as (incomplete list):

- structural information: detector components and their relations
- installation information: data about the hardware and software components and their configuration
- information for monitoring and debugging the detector, such as run logs, and damage reports
- operation information: auxiliary calibration measurements needed for off-line analysis.
- decommissioning information: information about parts replaced

The need for an integrated description of the complete detector comes from the fact that for a number of important activities, such as event tracking and reconstruction, component histories, and error tracking, information from all parts of the detector will have to be combined. For instance, the signals produced by the various sub-detectors, at successive moments in time, should be combined with the relative positions of the sub-detectors and their operation conditions at the corresponding times to allow for a consistent interpretation of the measurements. This information should be recorded at the time of data taking and remain available for a number of years to allow for a renewed interpretation of the data at a later date, when the detector and the physics are better understood. Error tracking combines information about various (adjacent) sub-detectors (such as locations, voltages, gains, and yields) for the analysis of signals or indications of malfunctioning of parts of the detector. A component history will comprise production data, but also functioning data such as parameter settings and conditions data during operation.

There should therefore exist a way to store, access, and relate the various bits of information, such that a consistent reconstruction of physics events can be made. A description of the complete detector also will allow one to identify information which is used or produced in the various sub-detectors or other components, or passed between them, and should be consistent. In this way proper interfaces and dependencies between the various components can be defined.

In this note, a database contains both the description of the data and the data itself. We distinguish between the following databases, each of which will support the activities around the CMS detector in a particular phase and therefore will contain information (types) that are relevant for that phase:

- the detector geometry database (Section 2)
- the construction database (Section 3)
- the equipment management database (Section 4)
- the configuration database (Section 5)
- the conditions database (Section 6)

The distinction above is a conceptual one that is common over many experiments but may be deviated from in practice. We will discuss these issues at the conceptual level and abstract from implementation technology. The relation to other databases will be discussed. In Section 7 we present some common issues, including the status of each of these databases and concluding remarks. At the end of this paper a glossary (on page 19) has been included to clarify the many acronyms and terms used in this paper.

2. Detector Geometry Database

The subject of this database is the description of the spatial aspects of the detector. The detector geometry is interpreted in a wide sense to not only cover the spatial aspects of the measuring device itself, but also that of the peripheral equipment needed for controlling the detector and for taking the measurements. The geometry is modeled as a hierarchy of volumes or slots (that we've termed CMS Slot in Figure 1) that can contain parts of the detector. We concentrate here exclusively on the spatial aspects. The description of the occupancy of the slots is the subject of section 4 where we discuss the equipment management database.

2.1. Content and clients

The detector and its supporting equipment can be modeled as a hierarchy based on the container-contained relationship between detector parts. Many detector parts can be viewed as composed of parts or components that may in turn be composite themselves. In the geometry description the volume or space that each physical detector or peripheral part will occupy is modeled, not these parts themselves. Such a volume is called a CMS Slot. The containment hierarchy is represented in Figure 1 by a Simple Tree pattern ([Gam1995, Est2000], consisting of the CMS Slot and its self referencing aggregation relationship. (We represent the models by means of UML class diagrams [UML]).

The CMS Slot represents a Bill of Materials structure for the detector to which spatial information (Nominal Location) has been attached for the location of the slots in the physical detector. The latter is given in terms of the absolute positions and orientations of all volumes with respect to the detector frame of reference.

For a complete geometrical description, also descriptive information such as the shape of the slots is needed. Since the detector itself is a fairly symmetrical construct, a Slot Type has been introduced to capture descriptive information which is common to a number of slots, such as a shape, modeled as the Solid, and possibly some other properties. This information, together with the Nominal Location, can be used to construct a 3D-representation of the detector. Note that there are some constraints to be satisfied by the descriptions. For instance, the solid of a container should encompass all solids of its contained components.

The data model can handle all information about the detector model that is used in the simulation and reconstruction software, but it is not limited to that. Its structure is suited for capturing both finer details (further decomposition), and information about peripheral structures such as the racks with the measuring equipment and power supply units that are partly co-located with the detector, partly located elsewhere (see also Figure 3 on page 9). This is made explicit in Figure 1 by the introduction of special Slot Types, such as the Detector Slot Type and the Peripheral Slot Type that represent volumes that will be occupied by a detector or a peripheral part, respectively. These sub-types will have distinguishing characteristics of their own.

The information in the database can be used for a number of purposes, such as visualizing the detector during construction, integration and operation. The CMS Slot is a core construct for a number of applications. First of all, it serves to integrate the sub-detectors. In Figure 1 it also serves as a point of reference to attach location information. In the same way, alignment (deviations from the nominal location), calibration, and configuration information can be attached to it (see also sections 5 and 6).

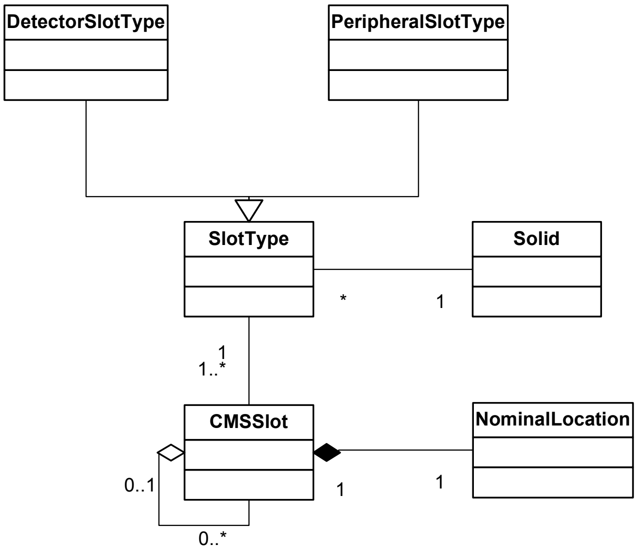

Figure 1: Detector Geometry

2.2. Requirements

The detector geometry database should represent the geometry of the detector as built. As such it can be regarded as a validated version of the detector geometry as specified in the CAD drawings, i.e., as designed.

The detector geometry should be kept synchronized with the model used in the simulation and reconstruction software. Since the latter constitutes an approximate representation of the detector, synchronization has to be limited to the slots that will contain sensitive detector components that are shared by both descriptions (see also section 2.3). These are the physical detector components that produce the measurement signals.

When the description of the detector changes after it has been put into operation, for example as a consequence of a different physical grouping of components, a new version of the detector description has to be created that reflects the new layout of the detector. This is a rather infrequent (once in a number of years) but still quite conceivable event. The old version has to be kept and has to remain accessible. In the model in Figure 1 these versions can be distinguished from each other by making use of a version attribute for CMS Slot. This kind of versioning will do, when only few, sequential versions are needed as is expected for the geometry. When a more sophisticated versioning scheme is needed, for instance when also versioning at the sub-tree level is required, the model will have to be adapted.

Performance

In its usage as a reference database, the look-up of the location of a Slot given its ID and of the ID of the Slot given its location should be efficient. Since the CMS Slot implements a container-contained relationship, identification of sibling Slots can be supported efficiently. Adjacency queries, in the case that no nearby common parent exists, are harder to do. One way of supporting such queries is to make use of database vendor specific geometric indices, such as grid files or quad trees, and the vendor supplied tools to construct them. Another, vendor independent, way would be to encode the spatial information into the component ID's.

The size of the database as indicated in Figure 1 is, in case the detector model of the simulation software is used for population, about 600 MB, including indices. Most space is consumed by the CMS Slot data (1.2 million volumes are distinguished in the CMS detector) and the location information. Not all of these volumes correspond to volumes occupied by detector components. A number of them correspond to intermediate volumes, so-called envelopes that are used for the convenient grouping of volumes. A detector geometry database populated on the basis of the CAD information will be larger.

2.3. Dependencies

The detector geometry data needs to be populated with the survey information coming from the detector assembly as constructed. A first, provisional population can be obtained from the geometrical information, laid down in the CAD drawings. Unfortunately, the CAD drawings in their present form do not support parent-child relationships. Moreover, these drawings have been constructed by multiple groups, using multiple CAD programs, and different naming schemes. At present, the transition to a new CAD system is being prepared, that may support hierarchical structures. In this transition the heterogeneity mentioned before will need to be resolved, to obtain a complete and consistent version of the detector design. From this new version of the detector as-designed, an initial population of the database could be generated that subsequently would have to be validated to arrive at a database population representing the geometry of detector as-built.

Another important model of the geometry of the detector itself has been constructed for usage in the simulation and reconstruction software [Geant4, OSCAR, ORCA]. This model has been optimized to make the simulation and reconstruction tasks feasible in terms of computing resource requirements. To this end, the detector is being described to an appropriate level of detail, making use of approximations and aggregate properties to represent finer details not explicitly included and to be able to exploit detector symmetries. Also a number of additional intermediate levels in the detector hierarchy have been introduced, to enable grouping of components to simplify manipulation in the software. A compact description of it exists in the form of XML documents [DDL2003, ML2003] and some C^{++} code, which are maintained by the detector experts. To save space the detector hierarchy is modeled in the compact description by a graph, which makes it possible to represent the information about identical components (composite or elementary) only once. This description is expanded on demand in memory by the software to obtain access to the individual detector components. The compact description is still being further optimized together with the code needed to manipulate it, under the control of a CVS versioning system.

The synchronization of the software detector model and the description in the geometry database could be achieved by embedding slot-ids for the slots into the (compact) detector description. This will on the one hand greatly enlarge the compact description, and on the other hand create a dependency of the software model on the hardware model. It will greatly facilitate the access to conditions data later on (see section 6)

The id-scheme for slots can be made geometry based on the assumption that the detector geometry will not change. The scheme will have to cover all possible slots and it should be external to the database, so that the assignment of an ID does not depend on the state of the detector information in the database. E.g., during the build up of the database, the addition of a part (insertion of a leaf in the detector tree) should not change the id-assignments. When the detector geometry changes and thereby the assignment of ids to volumes a new version of the database will have to be created. In that case a version dependent mapping of ids of corresponding parts may be generated.

2.4. Interfaces

A database prototype has been implemented using a relational database, which supports the standard query interface. The necessary data has been generated from the detector description used in the simulation software that has been saved as flat files and has been loaded into the database system.

On top of the standard interface a User Interface will be needed for data entry and validation.

In some cases, the sub-detector models can be regarded as two-dimensional views of the detector description model. For example, a part of the ECAL sub-detector, the barrel, has a cylinder shape and thus a two-dimensional surface. On this surface the sensitive detector parts, the crystals, have been mounted. The position of the crystals on this surface can be kept track of simply with a matrix (three in fact). When the only purpose of registration would be materials management, this would suffice. However, the use of specific representations for specific sets of detector parts breaks the homogeneity of the detector description as a whole. These representations should, if needed, be made available as views (i.e. as derived data).

A Web-based application is being developed for the visualization of the detector geometry that will support remote, geometry-based access to component related information.

3. Construction Database

The subject of this database, also known as Production database, is the description of the construction of a specific sub-detector up to the start of integration. This includes information about the components, and the process with which they were produced or assembled.

3.1. Content and clients

The Construction database is fragmented into a set of disjoint, independent databases that each documents the construction process of a particular sub-detector in isolation. The information covers the manufacturing history of the components, verification and initial calibration information, and workflows associated with the components. Each sub-detector has the responsibility over its Construction database.

The manufacturing history will contain details about the manufacturing process that produced the components: where and how the component was produced, which tests it was subjected to, where and by whom and what test results were produced. This information is complemented with production schedules, the tracking of movements of components between labs (workflows) for further development, and quality information. For more complicated components also information about the composition and assembly and the corresponding tests, is needed, as well as the initial calibrations and settings.

In Figure 2 [Cris1996, Cristal2, Per2003] (this Figure was abstracted from [Red2003]) an example of such information can be seen. Here Detector Components are modeled, which may be simple or composed and conform to a ComponentDefinition. This definition is used in a definition of the Workflow in which the component is produced. A WorkflowDefinition is comprised of a number of ActivityDefinitions, the outcome of which is specified in a CharacteristicDefinition. When the activities are carried out, they produce the specified outcome. Activities on components are carried out at RegionalCentres.

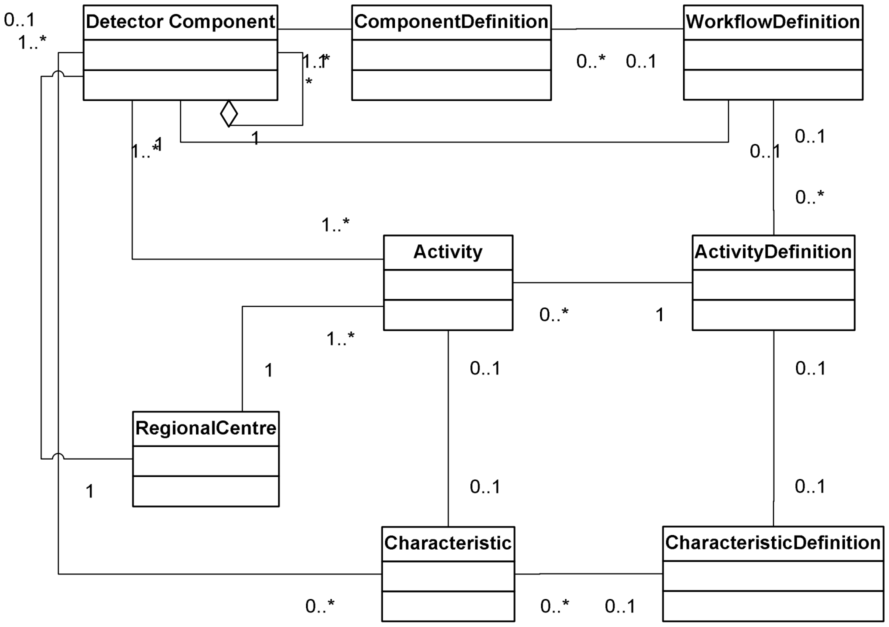

Figure 2: Example of Construction Process Information

Another example is the EMU (Endcap MUon system) construction database [Bre2001], which contains information about the workflows for producing the muon chambers and their electronics boards, the connections between the boards and the chambers, and the tests of the boards and the resulting parameters. The electronics parameters were used in further chamber tests.

The Construction databases are used by regular and Web-based applications for data entry and progress reporting. For instance, the results from the EMU chamber tests can be accessed via the Web for further analysis and possible upload into the database [Vor2003]. In some cases also automatic measuring equipment accesses the database for adding test results [Red2003, Per2003].

The data about the components and their composition will be useful for the databases in the later phases of the detector construction and operation, because it covers component descriptions, assembly, and (pre)calibration properties. Since the database focuses mostly on the individual components, there is no urgent need for data about the detector geometry.

3.2. Requirements

The construction database (including the workflows) will have to be available for the lifetime of CMS. The traceability of the production process is needed with regard to the decommissioning process later on (to fulfill the requirements of the Installation Nucléaire de Base (INB)).

Because the component-related information in the construction databases will be useful for other activities as well, the information entry points, in this case the data concerning components, should satisfy the global naming scheme. In practice this is not so.

Performance

The Construction databases vary widely in size. No high performance requirements exist for space or for access times. Long term durability has to be supported.

3.3. Dependencies

Some selected parts of the information in the Construction databases have to be available in the Equipment Management, Configuration and Conditions database. It is planned to copy the data to avoid duplication of the data entry effort. Since the construction databases have been created independently, extract, transform and load (ETL) programs have to be created for each of them to transfer the relevant data into the other databases (See [FL2004] for a CMS related discussion on this topic). This is where a lot of the integration work will have to be done, to reach a common infrastructure and consistent data content. The need for ETL tools is a good opportunity to enact uniform naming conventions over the complete CMS components dataset. After the data has been copied, the Construction database will have to be frozen, to avoid synchronization problems.

3.4. Interfaces

Since these databases have emerged independently of the detector description database as local registration tools, no specialized interfaces to other databases have been foreseen. However, most of the Construction databases have been implemented in Oracle, and some using MySQL, and thus provide an SQL interface. The Redacle and Cristal 2 databases have reported [Cristal2] a number of User Interfaces so that the technicians involved in the production processes can enter the construction and pre-calibration data and so that measurement equipment can automatically insert measurement and test data. These latter databases support data import / export facilities on the basis of XML formatted files. Access to the information in the EMU database is also possible through a Web-interface [EMU]. This interface also allows for the creation of comma delimited text files for the exchange of information between the various construction databases.

It was agreed that the Construction database will be frozen after completion of the construction process and that they will not be touched by global (system wide) queries. The relevant information will by that time have been copied to the other databases.

4. Equipment Management Database

The subject of this database, also known as the Installation or Technical Coordination database, is the first complete description of the detector and its peripheral equipment.

4.1. Content and clients

The Equipment Management Database (EMDB, see Figure 3) will contain the information about the detectors and the electronic parts, cables, racks, and crates, as well as the location history of all items. It supports the fulfillment of the INB requirements for safe disposal later on. The Geometry database is an integrated part and offers location information for detector setup.

In Figure 3 we see that the Slot hierarchy has been split up into two specific hierarchies, one for the detector itself (DetectorSlot) and one for the peripheral part (PeripheralSlot). With these slots the components are associated in a time-dependent fashion, which is given by the Occupancy. This construction takes care of the fact that the occupancy of a slot can change in the course of time, because one specimen of a component gets damaged and needs to be replaced by another one. The cabling is modeled by the Connector, which may run between a DetectorComponent and a PeripheralComponent, but, e.g., also between two PeripheralComponents.

The Equipment Management database is shared by all of CMS as it contains the first physically complete overall picture. It records every separate item installed in the detector. Since components will get damaged and will be replaced, the database will have to keep track of this as well. It should be able to present snapshots of the detector at specific times.

Other conceivable extensions to the information content are the inclusions of a magnetic (B-) field map, and of an irradiation map. An irradiation map would be pertinent for both sensitive and non-sensitive components.

B-field map is also an important part of "detector description" (Detector Geometry Database), since it is crucial for simulation and reconstruction. There is "ideal" and "real" field. The "ideal" field is calculated by specialized field calculation software, which takes as input parts of the ideal geometry (magnetic materials). The "real" field depends on real geometry and is obtained by measurement. Depending on the level of accuracy needed, the B-field information can be included in the description of the leaf-level components or in all of them.

The database will be set up by CMS integration. The sub-detector groups will enter the data and maintain it. These groups will also access the data during sub-detector configuration.

This database is also accessed by the cooling, ventilation and electrical distribution services. Also the Detector Control System (DCS) uses this database for look-up of serial numbers. The EMDB is the first database containing real, global part identification.

4.2. Requirements

The EMDB shall be maintained by the sub-detector groups that will each take care of their own setup.

The database should contain sufficient information to generate a history for each item in it, containing information such as item type, location, and movements, which could serve as a "passport" for the INB.

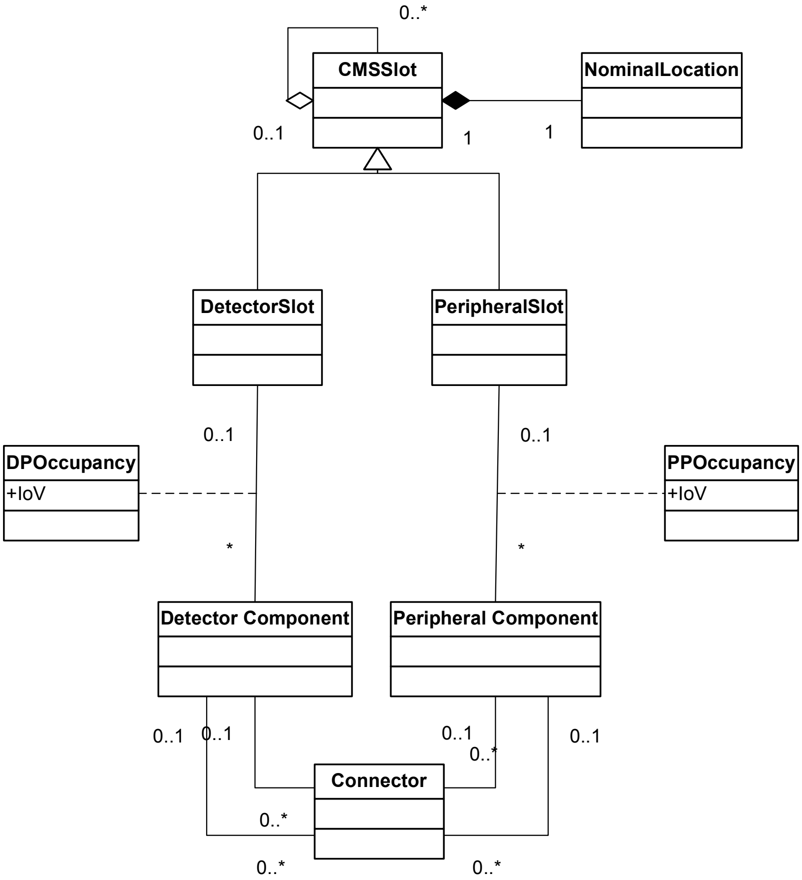

Figure 3: Example of Equipment Management Information

Performance

No high performance needs exist when it comes to data entry, since the information will be entered over a period of a few years from a number of sources, and usually by hand. Care should be taken to provide efficient reporting and look-up facilities.

This database is still relatively small with a size of the order of a few Gigabytes.

4.3. Dependencies

Part of the component information can be obtained from the production databases. Since many components began their existence in a construction database, the component history present in the Construction databases has to be copied as far as it is relevant.

The EMDB has an overlap with most of the detector geometry database. It may prove convenient to incorporate the geometry database completely into the EMDB as a read-only component.

The database will have to encompass an identification scheme that will allow for the tracking of the occupancy of particular volumes in the detector, such as the slot occupied by a particular component (see [AHB2003] for such a scheme). Several specimens of the same component may in the course of time occupy the same slot, as damaged ones are replaced.

4.4. Interfaces

The database has been implemented using a relational database with a standard interface.

There exists a user interface, called the Rack Wizard, that allows the sub-detector groups to enter the information about the peripheral components (control electronics) via the Web. The Rack Wizard provides facilities such as cable label handling (label printing and scanning). A Web export function provides cable routing to AutoCad programs. Separate Web-based interfaces support the browsing, maintenance, and export of information about the cables. These interfaces will be generalized to access all information in the EMDB.

An interface will have to be built to each of the various construction databases. Since these have all emerged relatively independently, different tools will be needed to transfer the construction data into the EMDB. Special care has to be taken, that in this transfer the CMS labeling and naming conventions [AHB2003] are enforced to the appropriate data. After entry of the construction data into the EMDB, there is no further need for these tools because the construction databases will not need to be accessed anymore from the EMDB.

The DCS will use the query interface to EMDB. This query interface will also be used by the reporting applications that will provide the component histories.

5. Configuration Database

The Configuration database holds all information required to bring the detector into a running mode. This concerns not only the hardware components such as the various on-detector boards, but also the parameters needed to configure the software components and the dataflow.

5.1. Content and clients

This database will contain sub-detector specific, even device specific information about the front end of the electronics configuration, and combined information about the DAQ , Trigger and the detector control (see Figure 4 and [TDR2, ORS2003]). Since each sub-detector has its own software and hardware control chains, this database has been divided over the various sub-detectors, and the setup and maintenance of these databases is the responsibility of the corresponding groups.

In the example of Figure 4, based on [TFE2004], the devices to be set have been modeled by means of a hierarchy to account for the big variety of both hardware and software components. These devices are grouped, into modules, and the groups have an on-detector controller. All these components have been modeled as special cases of a Detector Component. Controllers themselves are being supervised by an off-detector component, which is a special case of a Peripheral Component (a rack or crate) that is accessed by the Detector Control System (DCS) or Run Control and Monitoring System (RCMS) [Bri2003]. The device settings are given versions, where the versions can apply to a grouping (Partition) of devices that may encompass several modules.

The particular combination of versions and partitions that specifies the configuration of the detector is called a State in Figure 4.

Examples of data regarding the configuration of the detector are (see, e.g. [JCOP]): Modules on/off, high voltage, tracker global delay settings, and strip tracker Front End Driver (FED) gain settings. The database may also contain code to configure the programmable logic controllers (PLC's) of some of the sub-detectors.

The configuration database may also include initial (pre-running) CMS measurements, such as module alignment, module flatness and a survey of the magnetic field.

The use of sub-typing in Figure 4 rather than associations implies that the Detector Component model will have to be developed to sufficient detail, a finer grain of detail than is, for instance, needed for the reconstruction software. The alternative would be to use associations instead of sub-typing to express "mounted on" or "attached to" relationships between a Device and the Detector Component.

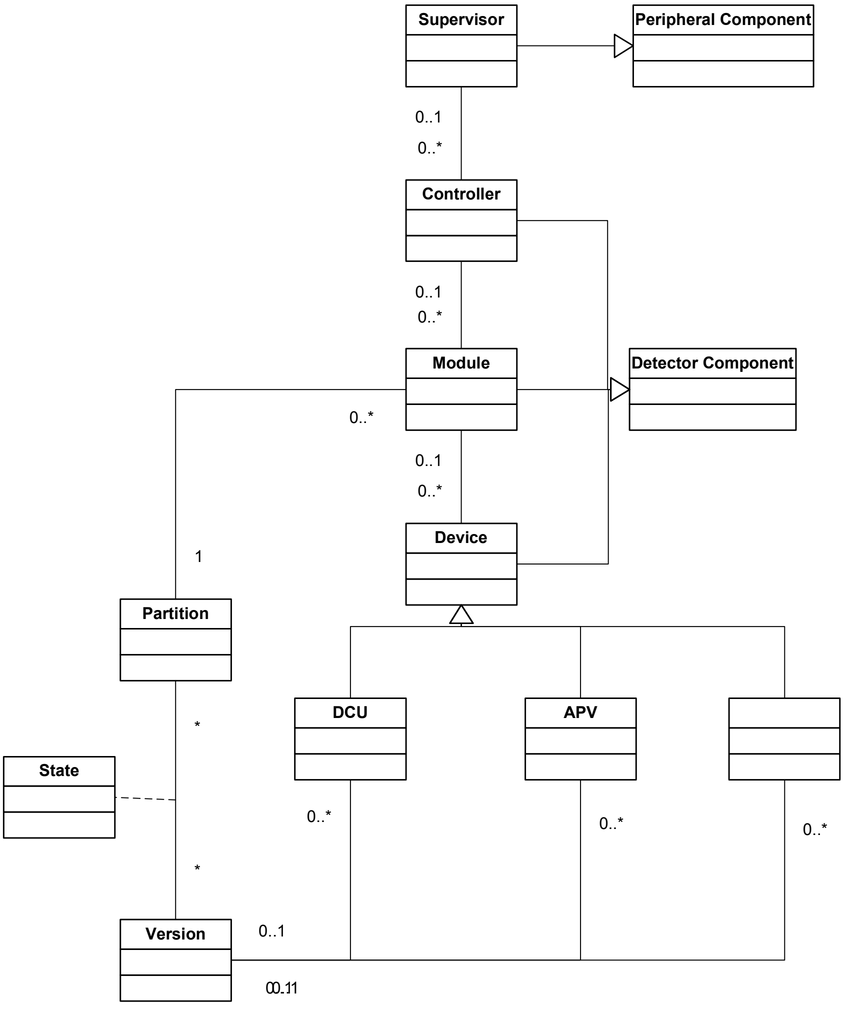

Figure 4: Example of Configuration Information

The distinction between the Conditions and the Configuration database is considered to be artificial by some. The argument is that configuration data is just a special, time independent variant of conditions data. Configuration data in this view provide a first value for conditions data, as they are the initial settings of the detector components. As soon as the settings have been read back (for checking purposes), they specify the condition the detector is in (which may differ from the intended settings). Thus the main difference is in the usage. Configuration data are used to set detector parameters and properties and in this way control the operation of the detector.

Based on the observation of the detector condition (by collection of conditions data), a change of configuration may be indicated. Since the changes in the configuration may be small, but necessary, many slightly different configurations may result from this. For instance, a detector module may stop functioning and cannot be made to function in time for the next detector start up. Such a module should be set in a switched-off state. To deal with this kind of situation, a sufficiently fine-grained versioning system may be indicated.

Some parts of the detector configuration, such as the software configuration, can be set electronically. Other parts, such as the detector alignment, are set before the detector is put into operation and can only be observed during operation.

Since the demands on the configuration data are different from those on the conditions data, it probably is useful not to hide them in the conditions data.

5.2. Requirements

The database should be able to contain different versions of the detector settings. It will take a while to understand the behaviour of the detector at LHC operating conditions, and various settings will have to be tried out. The database should support the creation of new versions by the combination and modification of existing ones.

Performance

An extreme requirement is that a single version of the configuration should be downloadable on the order of a second in order to take advantage of the best beam conditions. For the Pixel part of the detector that would mean 300MB of data to be distributed over 70M pixels. Presumably, the final requirements will be quite a bit less extreme.

5.3. Dependencies

Component information is first set by the data from the construction databases. Since the configuration database will have a structure different from that of the EMDB, a second set of data transfer (ETL) tools will have to be constructed to copy the relevant information from the construction databases [FL2004].

5.4. Interfaces

Also, the configuration data will be implemented in relational databases, which have a standard SQL interface. The hardware and software that reads and sets the detector parameters are part of the DCS (detector control system). A first interface (first prototype) to the DCS is available [TFE2004].

The part that sets the software parameters is part of the High Level Trigger and data acquisition system [ORS2003].

6. Conditions Database

The Conditions database, also known as Calibrations database, holds all information, such as calibration (including alignment), from which the condition of the detector at a given point in time can be deduced. It is used for monitoring the detector and for the support of on-line tasks such as error tracking, and is needed for event reconstruction.

6.1. Content and clients

The Conditions database will contain all parameters describing the run conditions of the detector. This information includes [CPT03]: calibration measurements of sensitive parts, high precision alignment measurements of sensitive parts or supporting parts, temperature and gas pressure measurements, magnetic field measurements, and possibly also test beam calibrations.

There are two main sources for conditions data:

- selective physics events: for example, a number of the alignment measurements are done using physics events. The interpretation of these measurements requires the reconstruction software and results in alignment corrections.
- data delivered by dedicated measurement devices like light monitoring systems, laser alignment systems, resulting in calibration constants

Examples of typical on-line conditions are opto-link gains, tracker global delays, pedestals, noise, and bad channels. These data are written in the on-line version of the conditions database and can be used, for amongst other things, error tracking. Information about bad channels can be transferred to the configuration database, where these channels can be turned off.

There is a choice between recording all conditions all the time, or only the time-dependent deviations from the configuration settings. In the first case the conditions data by itself is sufficient to obtain the operational condition of the detector at any given instant of time. In the second case the conditions data has to be combined with the configuration data to obtain the state of the detector.

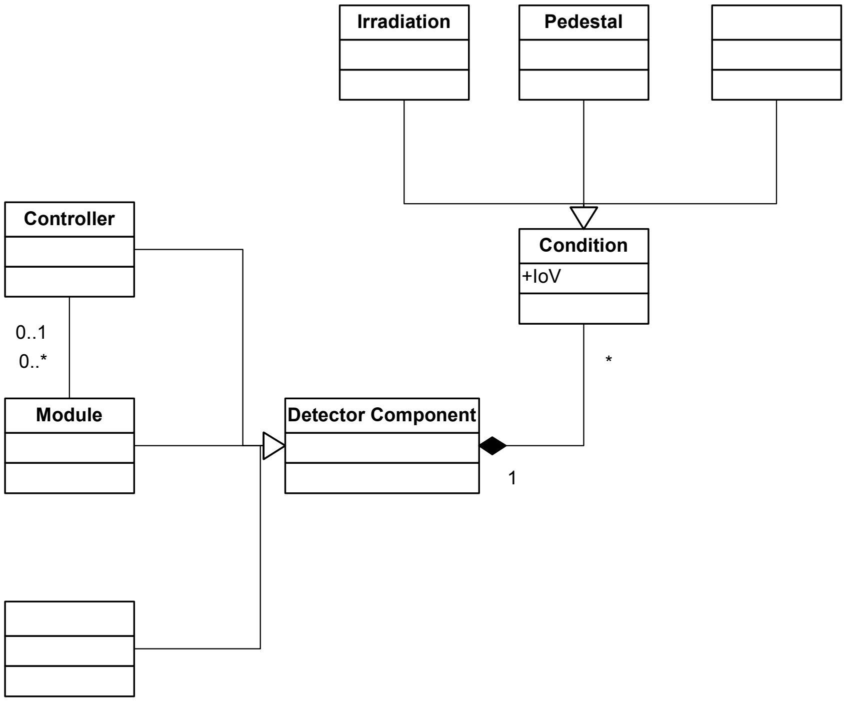

Figure 5 : Fragment of On-Line Conditions Information

Figure 5 shows an expansion of Figure 4 in adding conditions data, such as the pedestal and other directly measurable values, to the read-back detector settings that are needed for giving the actual status of the detector. Additional data, such as an irradiation map, are useful for equipment management and decommissioning. Note that these data have been associated with the detector components that are also part of the other data models to allow global accessibility of these data, for instance for reporting about component histories. The more detailed the detector component model is, the more direct the association.

Examples of off-line conditions are alignment, hit resolution, signal height, time resolution, Lorentz angle, and depletion depth. Part of this data is produced in versions of the reconstruction software and is fed back into the off-line version of the conditions database (see Figure 6). Note that also here the conditions data have been associated with the detector components.

Examples of environmental conditions are temperature, humidity, sensor leakage currents, and B-field normalization. These data are important for the detector control system that is concerned with the operating conditions of the detector as a whole.

The conditions data will be used by the High Level Trigger software on the filter farm for a first selection of interesting data. Since the load on the filter farm will be high, it seems unfeasible to use anything but periodic snapshots of the conditions database content as conditions data. The snapshots get updated during periods of low accelerator activity.

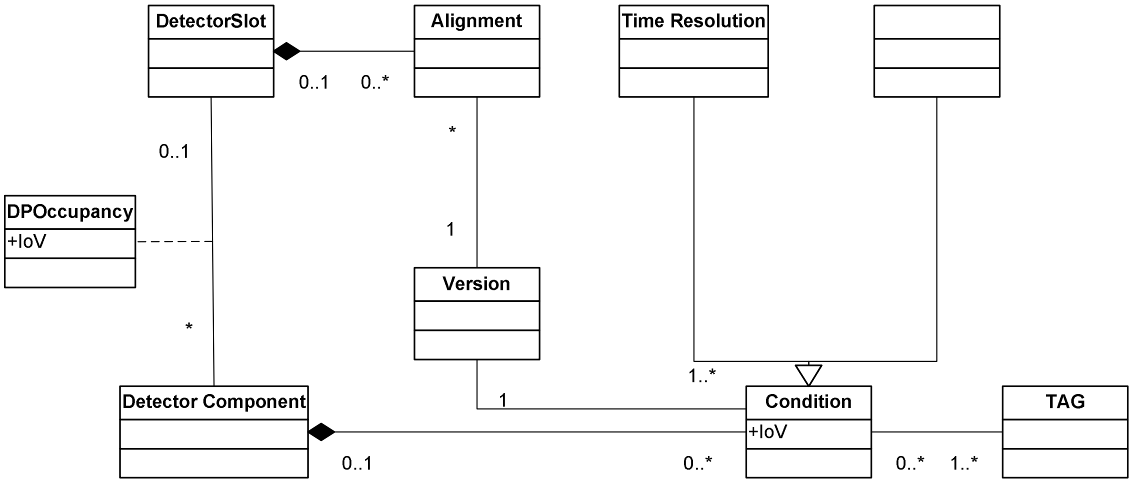

Figure 6: Fragment of Off-Line Conditions Information

6.2. Requirements

There appear to be two (conflicting) sets of requirements for this database. One set of requirements (mainly efficient reading and some writing back of (re)calibrations) come from the off-line reconstruction groups. Another set of requirements (mainly writing and some ad-hoc reading back) comes from the on-line groups. Since no database can be optimized to efficiently do both, an implementation into two databases seems to be indicated: an on-line conditions and an off-line conditions database.

The on-line conditions database will contain an enormous amount of data. It will be used for error tracking and support for configuration in the on-line system. Since this database will contain only the direct measurements, the data in this database will exhibit only a simple time-dependence.

The off-line conditions database will be comparatively smaller and will contain a condensed version of the on-line conditions data, augmented with calibration data, such as alignment data, gained from physics events. Initial versions of these data will be written by the reconstruction software that is used in the on-line setting of the High Level Trigger (HLT) on the filter farm. More refined versions will be produced by the subsequent off-line analysis. Because of its direct association with physics event data, this type of conditions data will also be given a tag for easy reference.

The conditions data exhibit large time dependence. Efficient processing of conditions data will require special attention for “temporal” operations on data. A part of this data has to be made available to the off-line reconstruction tasks, which appear to need much less data for their performance.

Performance

A Relational Database (RDB) implementation seems to be preferred for the on-line tasks. RDB’s provide support for consistency (no dangling pointers), reliability (crash recovery, hot standbys), concurrency, scalability and, not the least important, an open interface.

Size and throughput are the issues here. For instance, ECAL expects to collect 14 GB data on the ADC (Analogue to Digital Conversion chip) pedestals and crystal transparency in the ECAL barrel and end-cap sub-systems for monitoring purposes per hour [CMS2002/012]. Data production differs between sub-detectors, though. The DCS data is not included in this. The Tracker sub-detector expects to be reading out about 1 GB between fills.

The data requirements of the off-line software, used in the High Level Trigger on the filter farm, are in the area of fast data set switching. In some exceptional situations, a transition to a new set of calibration constants (for example to adapt to a noticeable change in conditions) has to be made consistently across the whole farm in a fraction of a second, in order not to lose too many events. This requires special, but relatively standard, cache management techniques. This operation is regarded as a desirable capability, but is not expected to occur very frequently. The HLT should be designed to be insensitive to calibrations that change quickly.

For each physics event that is registered, a matching (in time) set of conditions data should be available with the level of detail required for event reconstruction. This information may be versioned.

6.3. Dependencies

The initial values for the various hardware components and their settings can be obtained from the construction (ECAL) or the configuration databases or directly from the hardware (Tracker). To this, the configuration of the software will have to be added.

Suitable summaries of the on-line conditions database will have to be fed into the off-line conditions database. The level of detail that is needed and can be handled by the off-line software still has to be determined. This is somewhat dependent on the development of the computing infrastructure.

Some of the off-line conditions may be fed back into the configuration database, such as final alignment and pedestal values.

6.4. Interfaces

Since the on-line databases contain relational tables, an SQL interface for on-line querying will be available. The fraction of the conditions data that concern monitoring information about the detector will be made available via the PVSS (the commercial monitoring and control system) interface of the DCS and RCMS [Bri2003]. An interface for the PVSS system based on the detector geometry for visualizing slow controls is under construction.

The reconstruction software requires that the conditions data be available in the same way as the event data: as objects that are identified by a time and a version and that are possibly also tagged [LCGCDB]. These kinds of binary stream data have to be processed by the off-line reconstruction software to produce objects. An interface to access these data has been specified by Paoli [CDB] and implementations of this interface have been built for MySQL and Oracle by the ATLAS Lisbon group [LIS]. Note that this interface only deals with the time and version dependence and not with the internals of the streamed objects.

To supply the conditions data sets at a given point in time, an interface is needed that automatically converts relational data into binary stream data. For the binary data streams to be useful in the reconstruction software a match has to be made between the detector geometry as it is seen by the on-line systems (see Sect.2) and that in the reconstruction software. The on-line databases use a static identification of the detector slots and components [AL2004], whereas the software uses a dynamic identification scheme that is moreover subject to regular changes of versions.

7. Common issues

7.1. Sources

The main source of the detector geometry database is still the GEANT3/CMSIM code and in the CAD system (available via the Euclid server). This implies that this information is not available in a form, which is readily imported into a database. This situation is clearly untenable.

Provisions will have to be made to guarantee the quality of the databases, such as the assignment of which database has the master copy of which data and should be taken as the reference source for it. For instance, which database will act as reference for slot ids, which one for component ids, and how is replication handled. Typically, these are the databases closest to the source of the data, or they are themselves the source. An example of a change of ownership is the construction data. When the sub-detectors are put together, the construction of the components has clearly been completed. A part of the component construction data will be copied into the equipment management database that will from that point on own these data and act as the source for current component information. The original data in the construction database will be frozen to read-only.

Another issue is the cross checks between databases to ascertain whether the information is complete and consistent between databases. In the case of discrepancies, the data owner will provide the correct version.

The description of the various databases that have been implemented is hard to get to, with only a few notable exceptions [AL2004, Red2003, TFE2004]. This makes it very hard to identify and resolve integration issues.

7.2. Observations

In the CMS collaboration (and also in the other collaborations), development of the information models and the usage of the information go hand in hand. Several implementations are being developed on a trial and error basis. Only when a working version of a model has been made available, will it become clear on what points it satisfies the, up to then, tacit requirements and on what points it does not. This is too late for databases containing information that needs to be globally accessible.

What is lacking at the moment is the implementation of a uniform naming scheme. It is not possible at this point to track a detector part from inception to decommissioning. A uniform naming scheme for part-IDs would include one data type for all the databases concerned. At present only a prescription exists for a global format for the identification string (19-character format) [AHB2003], but this is not adhered to by all production groups. The ID- string would allow the encoding of the major sub-component that the part belongs to, and its unique ID inside this component. Versioning information of the part should also be included. A good place to introduce this naming scheme would be at the point where the construction database information is copied into the other databases. A reference copy of this can be kept in the geometry database. A complicating factor here is that for some sensitive components an ID has already been hard-coded into the hardware. These kinds of IDs will have to be incorporated into the naming scheme, or be mapped to it.

The location inside the detector should also have a unique id (the so-called slot-id). From this id the nominal position in the detector should be deducible and vice versa. The slot ids should have a two-level structure. The higher level would point to volumes that are interesting to the off-line software. The second level would point to positions taken by components (chips or boards) that are relevant for configuration settings but too detailed for the reconstruction software.

The matching of detector slots in the geometry database to the volumes in the detector model of the reconstruction software is of major concern. As stated before, the reconstruction software will be an important client for certain conditions data but has an evolving (software based) detector model. The match between the slots in the database and those in the software based model will have to be done at the level of sensitive and support (e.g., yokes) parts which correspond to stable parts of the software model. One possible strategy to do this is by embedding the database slot ids into the software detector description. A complication here is that the software model is based on a compact description [DDL2003] that is expanded in memory to recognize individual detector components. Since the detector model for the simulation and reconstruction software is composed by hand, this will be a manual task. Great care will have to be taken to leave the augmented part of the software model invariant under subsequent optimizations of the software detector model. Another strategy would be to make use of the mechanism used to match the sensitive detector parts in the various versions of the software model. In both approaches, the database geometry will also be fixed, because of this dependency. It is clear that a solution should satisfy the needs of both sides as well as possible.

A similar remark holds for the conditions data. The globally supported matching [LCGCDB] between event data and conditions data on the basis of Interval-of-Validity, Version or Tag is a high level matching that shields all (sub-)detector-dependent conditions data structures. These structures have to be known and agreed upon by both the database and the reconstruction software. In the latter case, this will mean that the conditions data structures will be hard-coded into the software (and thus will be highly resistant to change). This imposes a big dependency on the databases containing the source for these data. Fortunately, in the case of relational database implementations, one can use the view mechanism to shield some of these dependencies.

The detector metaphor can be used to access regions or components in the detector. This is useful for adjacency queries (e.g., give me all temperatures for a given period in the neighborhood of this specific component).

7.3. Status

Detector Geometry Database

A prototype of the Detector geometry was completed in November 2003. This prototype is discussed in CMS Internal note [AL2004]. Its relation to other parts of the CMS detector description has been discussed in [AGL04] and [ACLM04]. The prototype was populated with the relational equivalent of the transient detector model used in the reconstruction and simulation software programs ORCA and OSCAR. This yielded a detector hierarchy in which a lot of volumes are used in the simulation model as an envelope or grouping mechanism for the contained volumes. These volumes have no physical counterpart.

At present, no complete version of the CAD drawings exist in a form that is easily loadable into the database. Such a version may become available, when and if the transition is made to a new CAD package with support for hierarchies, as a replacement for the present Euclid system.

The geometry data has been augmented by alignment data for the individual Muon chambers. These data were generated by the simulation software. Five datasets were generated and then combined to provide an example of versioned, time-dependent data.

Construction Database

At this moment, there is no single Construction database. The construction information is fragmented over a set of disjoint databases that each contain the relevant information about a particular sub-detector in isolation. Each will eventually contain a set of data including, amongst other things, information about the construction process of the sub-detectors and their components, initial calibration information, and workflows associated with the manufacture of the components. Each sub-detector has the responsibility over its Construction database.

Examples of construction databases are the Redacle and Cristal2 database of the ECAL community [Cristal2, Red2003], and the EMU construction database [Bre2001].

The agreement is that after completion of the construction phase of the detector components the construction databases will be frozen and preserved for the duration of the CMS experiment. Relevant portions will be copied into the Equipment Management, Configuration and Conditions database.

Equipment Management Database

The EMDB at present contains a portion of the installation information on the peripheral devices on the two floors in the cavern. This part contains both component and localization information. Rack configuration is in preparation.

Work on the detector construction is already in progress. Some components have already been mounted on the detector support systems. This information is not contained in the central database, but kept by the responsible sub-detector groups.

In this database a first implementation of the geometry database has been integrated.

Configuration Database

A working group was started during the CMS week of June 2003. Configuration databases will be developed on a per sub-detector basis. A prototype using the EMDB and configuration database for electronics setup is planned with the ECAL.

As of December 2003, an initial version of a configuration database for the Silicon Tracker [TFE2004] and the EMU DCS system [Syt2003] exists. For the pre-shower, the other Muon chambers and the ECAL sub-detectors databases will be designed and implemented hopefully on the basis of the Silicon Tracker design.

The configuration database prototypes will be part of the DCS and will be accessed via the XDAQ on-line software framework [ORS2003].

Conditions Database

An initial statement of requirements was given at the CPT Week Calibrations workshop [CPT03].

The first investigations into the generation of interfaces to the RDBs have started to support the programmatic access to the relational version of the conditions data.

7.4. Conclusions and outlook

In this note, a point of view has been taken that the detector database is conceptually one database, regardless of how it is or will be implemented, in order to emphasize the common issues. The main purpose of the database is to act as the complete source of information about the relevant aspects of the detector. These relevant aspects have been defined as the geometry, construction, installation, configuration and conditions views of the detector.

One such issue is the availability of a common reference structure. Such a structure is needed without question when a single database is being designed. However, when the database is being developed in a fragmented way, many solutions are proposed, each one specific to one information island and thus no common accessibility is guaranteed. One option for the reference structure would be a uniform naming schema for the physical detector components. This option is complicated by the fact that for some sub-detectors the sensitive parts already have chip-encoded identification. Moreover, these ids will change when parts get replaced which implies the need for a versioned schema. Another option is to use the detector geometry, which captures the rather constant spatial aspects of the detector. In both cases the mapping to the detector model in the simulation and reconstruction has to be made.

Another issue is the intended use of the detector information. Some informal Use Cases exist for on-line debugging and configuration management. No Use Case exists, for example, in the use of conditions data in the trigger and reconstruction software. These Use Cases need to be formulated. They will determine more accurately the main uses of the detector information and may provide the basis for the implementation design. In view of the complexity and the uniqueness of the information, professional database administration will be needed to provide for proper authorization, access strategies and performance optimizations to support the diverse use of the information and to guard the quality of the data.

In view of the fact that most of the databases, and in particular the larger ones, are still in the conceptual phase now is a good time to design and implement the infrastructure needed to connect them. It will be (very) much harder to have the various databases to communicate with each other once they have emerged from isolation and are filled with GBytes or TBytes of incompatible data.

Acknowledgements

It's a pleasure to thank Frank van Lingen for several useful discussions and comments.

References

[AL2004] A.T.M. Aerts, M. Liendl, R. Gomez-Reino *“Detector Geometry Database”* CMS Internal note 2004/011

[ACLM04] A.T.M. Aerts, M. Case, M. Liendl, Asif Jan Muhamad “*CMS Detector Description: new developments*”, CHEP04, p235, 2004.

[AGL04] A.T.M. Aerts, F. Glege, M. Liendl “*A database perspective on CMS Detector data*”, CHEP04, p225, 2004.

[AHB2003] A.H. Ball, *“CMS numbering and naming scheme”*, http://cmsdoc.cern.ch/~cmstc/naming_and_labelling/

[Bre2001] R. Breedon, M. Case, V. Sytnik, and I. Vorobiev *"Database for Construction and tests of End Cap Muon Chambers"*, http://www-hep.phys.cmu.edu/cms/TALKS/cms_01_sep_ivorob.pdf

[Bri2003] V. Brigljevic et al., “Run Control and Monitoring system for the CMS Experiment”, THGT002, Proceedings of CHEP2003, La Jolla, CA, 2003.

[CDB] S. Paoli, *"Conditions DB Interface Specification"*, CERN-IT Division

[CPT03] CPT Week 3-7 November 2003: Calibrations workshop http://agenda.cern.ch/fullAgenda.php?ida=a035751#s2

[Cris1996] J.-M. Le Goff et al., *“C.R.I.S.T.A.L./ Concurrent Repository and Information System for Tracking Assembly and production Lifecycles”*, CERN CMS Note 1996/003

[Cristal2] Homepage: http://proj-cristal.web.cern.ch/proj-cristal/

[DDL2003] M. Case, M. Liendl, F. van Lingen, “*XML based Detector Description Language”*, CMS Note 2003/000

[EMU] EMU (Endcap MUon) Database page: http://cmsdoc.cern.ch/cms/CSC/CERN/db.html

[Est2000] F. Estrella, *“Objects, Patterns and Descriptions in Data Management”*, Ph.D. Thesis, University of the West of England, Bristol, England, December 2000.

[FL2004] F.J.M. van Lingen, “XML and Graphs for Modelling, Integration and Interoperability: a CMS Perspective”, Ph.D. Thesis, Eindhoven University of Technology, March 2004.

[Gam1995] Gamma Erich, Richard Helm, Ralph Johnson, and John Vlissides. *"Design Patterns:Elements of Reusable Object-Oriented Software".* Addison-Wesley, Reading, MA, 1995

[Geant4] GEANT4 home page： http://geant4.web.cern.ch/geant4/

[JCOP] JCOP Framework Configuration Database Tool, *“Use Cases Document”*, LHC-IT/CO Joint Control Project. Homepage:

http://itcobe.web.cern.ch/itcobe/Projects/Framework/Developments/Configuration/welcome.html

[LCGCDB] The ConditionsDB project home page: http://lcgapp.cern.ch/project/CondDB

[LIS] A. Amorim et al., *"Experience with the Open Source based implementation for ATLAS Conditions Data Management System"*, MOKT003, Proceedings of CHEP2003, La Jolla, CA, 2003.

[ML2003] M. Liendl, *"Design and Implementation of an XML Based Object-oriented Detector Description Database for CMS"*, Ph.D. Thesis, Technische Universität Wien, April, 2003

[ORCA] CMS OO Reconstruction home page: http://cmsdoc.cern.ch/orca/

[ORS2003] L. Orsini "*DAQ Annual Review 2003: Online Software*", AR-2003 16 September 2003

[OSCAR] CMS Simulation Project home page: http://cmsdoc.cern.ch/oscar/

[Per2003] W.S. Peryt et al., *"Detector Construction Database System for ALICE Experiment"*, THKT002, Proceedings of CHEP03, La Jolla, CA, 2003.

[Red2003] L.M. Barone et al., *"REDACLE: A Database for the Workflow Management of the CMS ECAL Construction"*, CMS Note 2003/022

[Syt2003] V. Sytnik, *"Implementation of EMU Configuration Database with OCCI"*, 31 October 2003, http://agenda.cern.ch/fullAgenda.php?ida=a036126)

[TDR2] CMS TDR 6.2. *"The Trigger and Data Acquisition Project, Volume II: Data Acquisition and High-Level Trigger"*, CERN/LHCC 2002-25.

[TFE2004] F. Drouhin et al., "*The CERN CMS Tracker Control System*", CMS Internal Note 6 January 2004.

[UML] http://www.uml.org

[Vor2003] I. Vorobiev, *"Information Pool at CERN"*, Talk 25 February 2003, http://agenda.cern.ch/fullAgenda.php?ida=a03481)

Glossary

Terms and Abbreviations

ADC	Analog to Digital Converter: electronics component
as-built	Description of a manufactured object that conforms to the specification laid down in the as-designed description.
as-designed	Blue-print or specification of an object that may serve as metadata for its manufacture.
ATLAS	A Toroidal LHC ApparatuS is the name of an LHC-detector as well as the collaboration that builds and operates it
CAD	Computer Aided Design; computer tools for making blueprints.
CMS	Compact Muon Solenoid is the name of an LHC-detector as well as of the collaboration that builds and operates it.
CMSIM	CMS Simulation and Reconstruction Package
Compact View	Graph structured description (of the detector) that includes a node for every *kind* of (detector) component. Components with an identical specification are therefore included only once.
CPT	Computing, Physics and Trigger-DAQ are three central CMS detector sub-projects
CVS	Concurrent Versions System
DAQ	Data Acquisition, CMS sub-project
DCS	Detector Control System; is responsible for the traditional control and monitoring of all detector services and other elements, from high and low voltage supplies to temperature sensors and front-end electronics configuration. It is autonomous but will be driven during data-taking by the RCMS.

DDD Detector Description Database: set of XML documents that describe the compact view of the CMS detector geometry, including materials, optimized for CMS simulation and reconstruction, and the sub-detector specific data for the compact and the expanded view needed for simulation and reconstruction.

ECAL Electromagnetic Calorimeter, CMS sub-detector

EMU Endcap MUon system, CMS sub-detector

ETL Extract, Transform and Load; refers to a set of tools for migration of data between heterogeneous databases, such as from a data source into a data warehouse.

Expanded view Tree-structured description (of the detector) that includes a node for every individual (detector) component.

Geant4 **Geant4** (Geometry And Tracking) is a toolkit for the simulation of the passage of particles through matter.

HCAL Hadronic Calorimeter, CMS sub-detector

HLT High Level Trigger

INB Installation Nucleaire de Base

JCOP Joint Control Project

LHC Large Hadron Collider: experimental facility at CERN

Muon Muon subsystem of the CMS-detector

MySQL Open Source Relational Database Management System

on-line database Database that supports on-line processes, such as the control of the detector and the data taking processes.

off-line database Database for the support of analysis and reconstruction processes that are not directly coupled to the measuring processes.

Oracle Commercial Relational and Object-Relational Database Management System

ORCA CMS reconstruction code (Object-oriented Reconstruction for CMS Analysis).

OSCAR CMS simulation code (Object-oriented Simulation for CMS Analysis and Reconstruction).

PLC Programmable Logic Controller: a device used to automate monitoring and control of an industrial installation, such as a detector or accelerator.

PVSS Prozess Visualisierungs- und Steuerungs-System. It is a toolbox for automatic detector monitoring and control, to be used for supervisory control and data acquisition (SCADA). It is the choice of the LHC experiments for DCS.

RCMS Run Control and Monitoring System provides the user interface to CMS DAQ. Its functionality includes the configuration of all elements in the DAQ system, monitoring their performance, display their status, identification of malfunction or underperformance of elements and provision of recovery mechanisms (resetting and restarting after failure).

RDB Relational Database System, such as Oracle, MySQL or PostgreSQL

SCADA Supervisory Control and Data Acquisition Systems are used to monitor and control installation status and provide logging facilities. SCADA systems are highly configurable, and usually interface to the installation via PLC's.

SQL Structured Query Language. Standard query language for relational databases.

TAG A tag is a label for a data set that is used for easy reference. Such a set is typically composed out of data objects with different versions and intervals of validity. Some tags are assigned automatically, such as that for the set containing the most recent versions of the data items, other tags are assigned by hand.

Tracker CMS sub-detector

Transient detector model In-memory, tree-structured model of the detector that is generated from the compact description of the detector.

Trigger	CMS sub-project
UML	Universal Modeling Language
XML	Extensible Markup Language