C.N.R.S
Centre National de la Recherche Scientifique

I.N.F.N.
Instituto Nazionale di Fisica Nucleare

The Hardware Inventory & Integration Database

F.Carbognani, G.Hemming, D.Sentenac

VIR-014A-09

Date : 16th February 2009

Change Record

Version	Date	Sections Affected	Reason / Remarks
VIR-014A-09	16/02/09	All	First version.

Contents

Introduction

1.1 Purpose

This document deals with the use of the hardware inventory concept in the context of a large instrument environment and gives a detailed approach in the case of the Virgo experiment. Far beyond the original needs for setting up a spare hardware database for Virgo, the purpose of the Hardware Inventory and Integration (HWII) database is to become a living image of the detector, providing precious detailed information on its status in every moment of its operation.

1.2 Scope

The scope of this document is to present the HWII database in its key concepts, the achievement made so far, and future developments envisioned.

1.3 Reference Documents

[1] A database perspective on CMS Detector data, A.T.M. Aerts et al., p225, CHEP04, Interlaken CH, Sept. 2004
[2] Detector Construction Database System for ALICE Experiment, W.S. Peryt et al., Computing in High Energy and Nuclear Physics, 24-28 March 2003
[3] User Guide for the Virgo Hardware Inventory and Integration Database, G. Hemming, EGOXXX

1.4 Abbreviations and Acronyms

DAQ	Data Acquisition System
HWII	Hardware Inventory and Integration
IC	Integration Chain

2 General Description

2.1 Problem Statement

A crucial aspect of big instrument environments is the sharing of information among the people involved in the functioning of the experiment, where the information needed covers very large periods, from the construction time up to the final data analysis. Efficiency is considerably improved when the path access to the information is clear. Unfortunately, it is not easy to access information that results from broken up groups of competencies, various places and epochs. In general, a big instrument is divided into well defined subsystems, each taking care of their own information data. But when the instrument is used as a whole, information sharing becomes essential between subsystems. Ideally, the information data structures needed to operate the detector as a whole would need to be present in the database before the data is entered. If this is not the case, after a certain time of maturity of the experiment, an effort is desirable in order to centralize the information access paths and gain in instrument operation and maintenance ease.

2.2 Objectives

The Virgo experiment now has a sufficient degree of maturity. We have taken advantage of the enhancement period leading to Virgo+ to set up a tool aimed at simplifying the various detector information source path access points. In terms of software tools, we use a robust relational database system to centralize the information, the HWII information is made available through a handy web UI and through the use of many language clients by providing a C API.

Although focusing on supporting the current Virgo project upgrade phase, the HWII database aim is to become a tool that is able to support both Virgo and Advanced Virgo in all project phases.

The HWII baseline information for each item is its identification, localization and physical connectivity to other items. Different item characteristics data can be added through a flexible mechanism called part type management. However, the item data is not limited to the proposed list above. Different data sources can be combined together. In particular, configuration/conditions data held by a separate Configuration/Conditions database is linkable to the HWII database.
The main interface client provided is a public web page, accessible from anywhere. Users will find the possibility to fully manage the information related to their part, and find easily the localization of a part by navigating the integration chain. Detailed how-to information can be found in the user guide [3]. A dedicated C API (in read mode only) is also available to be used inside other software, including basic query functions (stored procedures) to more specific query functions.

2.3 Constraints

The HWII must be considered as a living and functional part of the experiment. The information reliability depends strongly on users inserting their information. Maintenance operation must include interaction with the HWII. Operations such as replacing a part, or making a new cable connection, must be registered in the HWII in order that it is kept up to date. A strong policy for user management is crucial in order to control the quality of the information. Each subsystem may have its own HWII Group Administrator, guarantor of the information for their own group. Each new part may be submitted to an HWII Administrator, in order to avoid any errors in the part description that may have an impact on the data quality and visibility later on.

3 Database Description

Historically, the first effort to centralize information concerned the electronics hardware and optical elements inventory. A first database attempt was made to comply with the expressed needs on the parts status, localization and availability. In parallel, a growing need to make the software configuration for the data acquisition system (DAQ) more comprehensible has lead to the formulation of the general requirement to combine different subsystem information, i.e., software and electronics, and also different database information such as configuration and integration data. These considerations lead to the refactoring of the first database, by placing more emphasis on the information data structure content of the relational database, in order to allow more flexibility in information granularity and cross-relation, focusing on the requisite of providing a tool that can be used as a primary access point to a broad range of information.

3.1 Database Principle

The experience gained in other big experiments helped to express the fundamental baseline for such a database: the detector geometry is the skeleton for any descriptive information [1]. In relational databases a self container-contained field concept is used with the general constraint that all children can have at most one parent. In the HWII, each component or part has a *unique* geometric ID relating to its logical localization, as primary information. The logical localization of components constitutes the minimum information required to provide a hardware inventory. From this model, one can envisage detailing the geometric information by attaching a spatial coordinate system to the detector in order to add spatial information to each part. Spatial information is relevant for particle experiments, which need to track the path of particles within the geometric assembly of tracker units. For what concerns Virgo, it may be used for particular items, such as optical benches and vacuum tower viewports, as we will see later. The advantage of the self container-contained logical part table is to allow the reconstruction of the logical position of each component with respect to one another in a tree, also called the Integration Chain (see Fig.1). This leads to a centralized, *global to detailed* information access page to the detector, where items can be easily managed.
In addition to this in-depth tree-based representation, we introduce a connectivity model that allows the interconnection of elements through the use of physical cables. In other words, it adds the possibility to relate elements of the tree, transversally. The cabling is an essential piece of information, physically linking elements together across subsystems and providing the baseline for the configuration mapping of the detector. This is especially true for Virgo+, having control loops tightly relying on a complex optical fiber network topology.

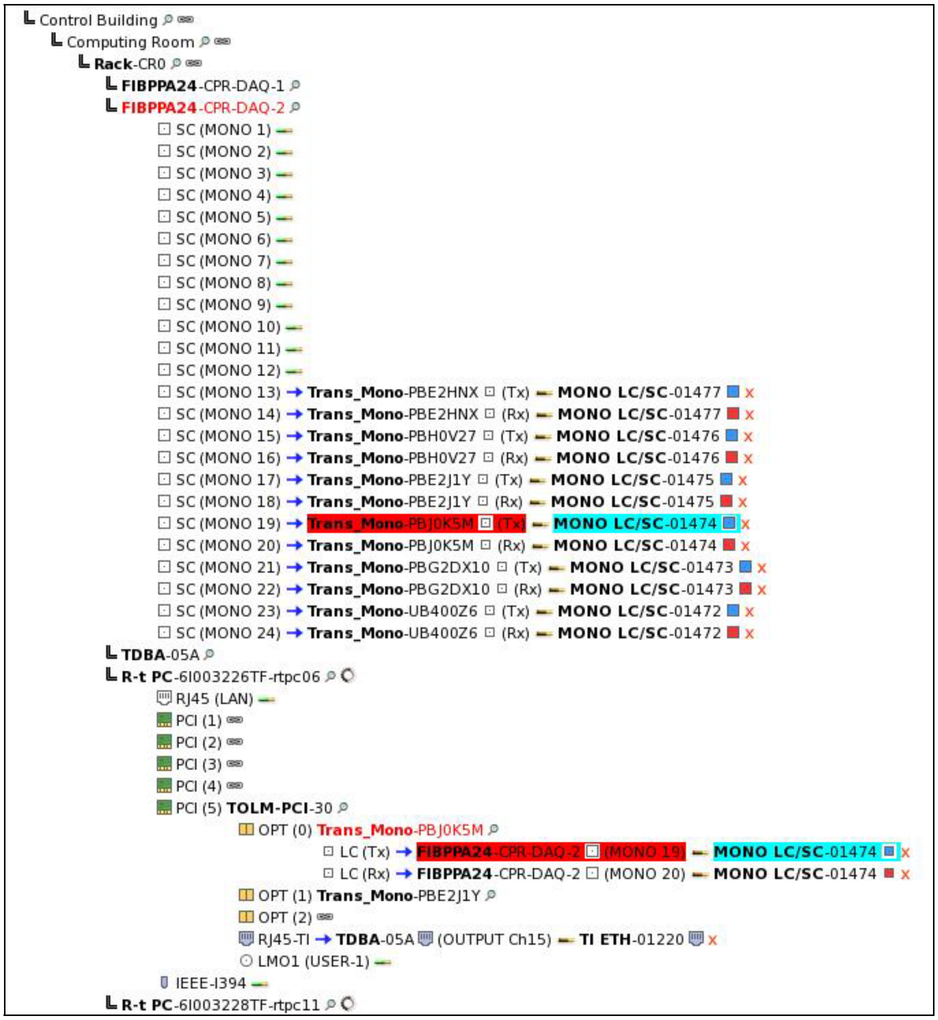


Fig.1: Logical tree representation of part localization using the self container-contained relational concept. In addition, an optical fiber connection logical representation is shown between a patch-panel and a TOLM-PCI. The TOLM_PCI part occupies a PCI slot of the R-t PC, the Trans-Mono part occupies an OPT slot of the TOLM-PCI, and the cable MONO LC/LS occupies an LC connector of the transceiver Trans-Mono.

3.2 Parts Principle

Each part belongs to a predefined category type (the name of a board, controller etc...). A part type is defined by some general properties. It may be done by the Group Administrator and is detailed in the following sub-sections.

3.2.1 Serial Number

When requested, the serial number must be unique for each part instance within its category type. The Serial Number allows the identification of the real part instance, and the accessing of its related information. The Geometric ID has just a functional meaning, and is used internally to locate the part. A serial number may not always exist, which means that it would be impossible in such a case to identify a unique object without the knowledge of its geometric ID. This may apply to collection of optical lenses for instance, where only the number of spares and their localization as a whole and overall characteristics will be of importance for the user.

3.2.2 Cable or Device

A part may be defined either as a *device* component or a *cable*. In the latter case its outputs can only be connectors as explained below. In addition cable localization may be treated differently to device components as they can exist in more than one place at a time, given that it can be connected to more than one part.

3.2.3 Available outputs

The output type is a key concept which tells if the part itself is a leaf or a node in the tree representation. If it is defined as a *connector*, it means it is a leaf, and will be used for cabling purposes. If it is defined as a *slot,* it is a node and another (smaller) component can be attached to it. The part itself, when not defined as a cable stands for a node. The output of type *slot* is used to refine the description of the part, by telling which type of component can be attached to it (compliant with the type). For instance, we can define a Real-Time PC (R-t PC) machine holding 5 *PCI* outputs of type *slot* as well as 1 *RJ45* output and 1 *IEEE-1394* output of type *connector*. Another example is an optical transceiver holding 2 *LC* outputs of type *connector.* As shown in Fig.1, the transceiver outputs allows transversal connection relations through the use of the *LC* connector, while the *PCI* slot only allows in-depth parent-child relation inclusions. Using this capability, the optical fiber network connecting all processing units is described in the HWII database, revealing the control loops topology. An important aspect of the web UI is the flexibility left to the user to define their components. A part can always be defined down to a certain level of details, using more or fewer outputs, and it is always possible to refine this by adding new outputs at a later moment.

3.2.4 Group

Each part type is attributed to a Group, which relates to one of the Virgo subsystems, having one or more Group Administrators. The part type definition setup is done by the Group Administrator. The Integration Chain can be browsed selectively by Group or by viewing part types belonging to all Groups. The Group attribute enhances security by enabling the possibility to restrict the modification of part instance details and the movement of part instances within the Integration Chain specifically to users belonging to the related Group.

3.2.5 Part Management

By default a part type is provided with a default description sheet. Besides its category *type* and *serial number*, a static *additional information* field allows for the addition of more details on the characterization of the part instance. A dynamic *work log* associated to each part instance serves as a container for inserting historical information and document links, where for instance, the part status can be explained in details (see Fig. 2).

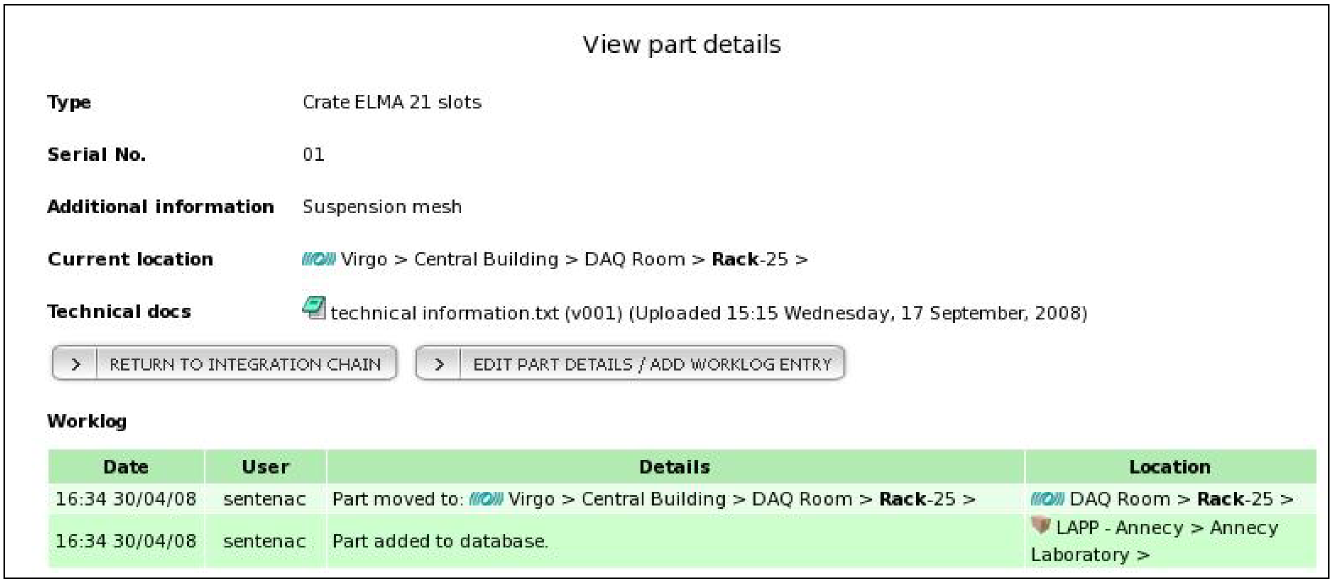


Fig.2: Typical default information sheet containing the part type, serial number, an additional information field, the current location pointer, some attached technical documents, and a work log history.

In addition to the basic generic needs of all part instances, some types can have particular, specific requirements in terms of the recording and visualization of additional fields. The DB table structure used for this feature is generic and a similar idea can be found in other experiments [2]. It preserves the data structure clarity and avoids in particular the proliferation of DB tables in the long term (see Annex 1).
The web UI takes advantage of this structure through the presentation of *Part Management* sheets. These allow Group Administrators to visualize all part instances in a given group in list format and to define additional fields to individual part types.

Optics part management

Beam Dump

Part	Mechanics	Optical Characteristics (supplier/reference)
Beam Block	(Thorlabs/design)	-
Beam Dump	(CVI/design)	-
Beam Trap	(EGO/design)	Schott-Desag Athermal 11, th= 3.2mm, T=10-7, R=0.003-0.01 depending on incident angle, AR coated on both sides (Schott)
Absorbing Glass	(EGO/design)	Schott-Desag Athermal 11, th= 3.2mm, T=10-7, R=0.003-0.01 depending on incident angle, AR coated on both sides (Schott)
Dihedron Beam Trap	(EGO/design)	Schott-Desag Athermal 11, th= 3.2mm, T=10-7, R=0.003-0.01 depending on incident angle, AR coated on both sides (Schott)
Faraday Beam Trap	(EGO/design)	Schott-Desag Athermal 11, th= 3.2mm, T=10-7, R=0.003-0.01 depending on incident angle, AR coated on both sides (Schott)

Diaphragm

Part	Characteristics (supplier/reference)
LB_DP1	(OCA)
LB_DP2	(OCA)
LB_DP3	(OCA)
LB_DP4	(OCA)
LB_DP5	(OCA)
LB_DP6	(OCA)
EIB_DP1	(EGO)
SIB_DP1	Circular, diam=110 mm?, anodized (EGO)
SDB_DP	Rectangular, 40*30 (L*H), anodized (LAPP)

EOM

Part	Mechanics	Optical Characteristics (supplier/reference)
EOM1		14MHz, LiNbO3 (LINOS/documentation)
EOM2		6 and 22 MHz, KTP (Cristal Laser/documentation)
EOM3		8MHz, KTP (Cristal Laser/documentation)
AOM		80MHz

Fig.3: Part Management visualization showing additional fields relating to individual part types.

Fields added to specific part types are automatically rendered visible in the Part Details View/Edit visualization as well and can be edited either within that section or directly within the Part Management list. All modifications of all fields are directly recorded in the part instance Worklog.

The aim of this feature is to provide users that are members of the different groups within HWII to have a quick visualization of all of the parts that are of direct interest to them. The basic philosophy of the tree localization remains the same.

In addition to the possibility to add field values, there is also the possibility to define a specific image to display with a given part type. It is then possible to virtually locate child part instances on this image, thus providing a straightforward interactive interface for the user showing exactly where a part is located in greater detail than is available within the Integration Chain. An example of this capacity is the Tower Wall part type, parent to both viewports and blind flanges – and shortly also to Local Control part instances – and can be seen in Fig. 4 below.

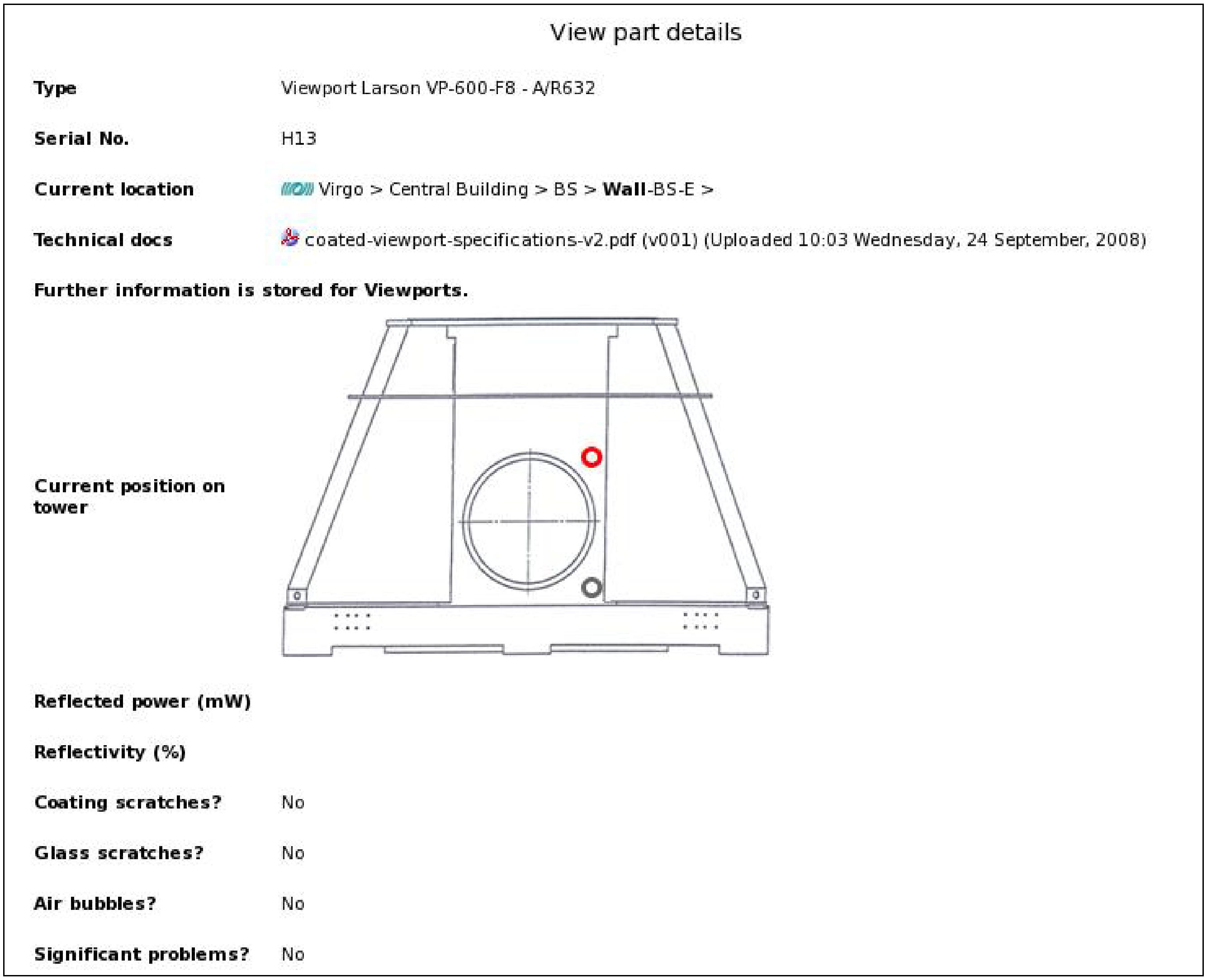


Fig.4: Specific image visualization and Part Management-defined fields, requested by the vacuum team. Besides the default fields, a visualization of the current position on the tower face, and a quality flag list has been added.

This feature works by laying an invisible grid across the image associated with a part type. Child part instances can be located on the individual points of this grid.

We foresee the implementation of this function of HWII across all part types to the extent that it ultimately provides a complete topological representation of Virgo in the web UI.

3.3 Relation with other Software Systems

HWII needs to be connected to other software systems employed in the experiment. The various links are illustrated in Fig.5 and are described in the following sections.

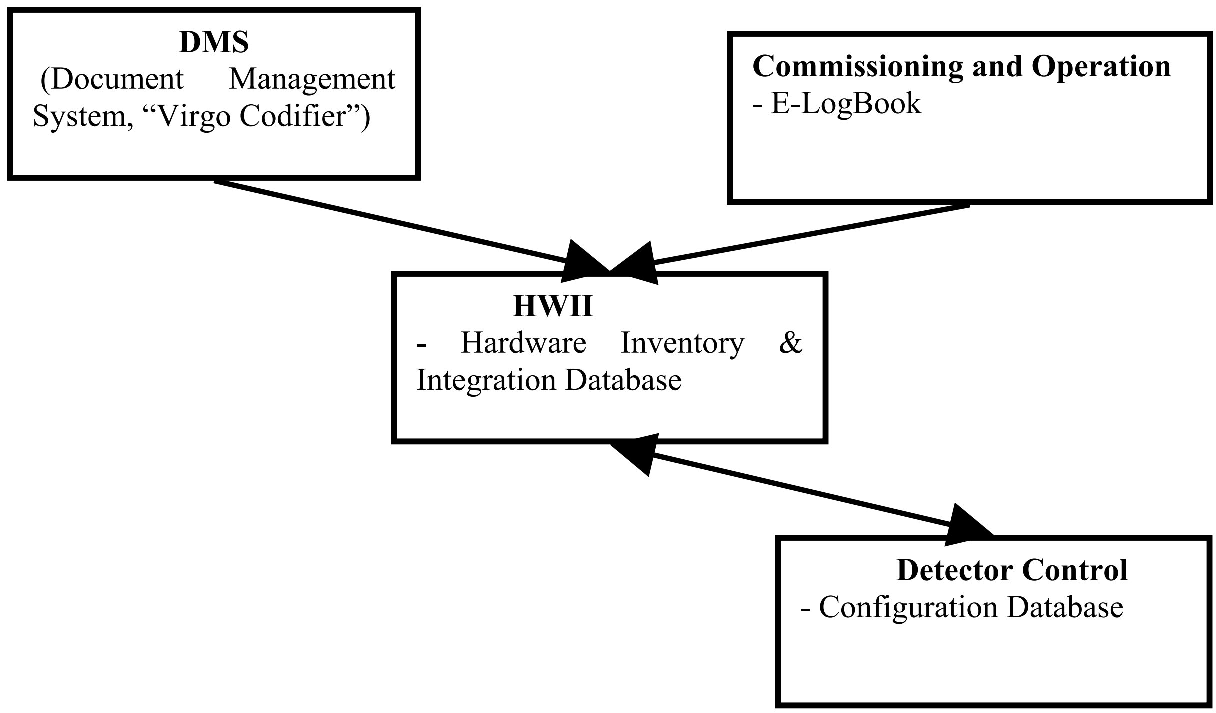


Fig5: Information exchange between HWII and other software in Virgo

3.3.1 Link to DMS (Virgo Codifier)

The Virgo Codifier is a document database dedicated to the storage of information relevant to the conception, construction, and exploitation of the Detector. A specific project has recently been started in order to upgrade the current system so that it incorporates the features necessary to render it a state of the art Document Management System (DMS). This is particularly relevant to the support of the specific needs of the Advanced Virgo project.
HWII is linked to the Virgo Codifier as Administrators may call specific document types within specific codes and attach them to part types as Technical Documents. For a standard user these documents then become available when browsing part details and appear in precisely the same way as a Technical Document that has been specifically uploaded to that part type.

3.3.2 Link to Virgo LogBook

In HWII it is also possible to directly link to an entry of the Virgo Logbook database. Inversely, an HWII Worklog can also be pointed to from a LogBook entry. This feature will also be expanded to include links to the various instances of the Logbook applications now used in different aspects of the experiment and R&D.

3.3.3 Link to Configuration Database

The Configuration Database holds all information required to bring the detector into any running mode and is a basic constituent of the Detector Control System (DCS).
Since HWII contains information relevant for the configuration of some of the servers composing the DCS (in particular the optical fiber network topology) the possibility to automatically update the corresponding information in the Configuration Database is very important in order to avoid information redundancy and mismatch. The best way to perform such an automatic upgrade is currently being investigated.
Moreover the information hosted in the Detector Control Configuration could be very useful when integrated into the HWII. For the moment we have implemented the possibility to display the list of processes running on a given CPU machine, by linking the Configuration database information to the HWII database, using the machine *alias/name* bridge. This information is precious for the DAQ activity. Indeed, when a problem occurs in the DAQ chain, one often needs to identify communication dependencies between processes running on different machines.

3.3.4 Additional Links being considered

As well as serving as a repository of part localization and connection information, HWII also serves as a portal to other information relating to the Virgo experiment, as is detailed in the two above sub-sections. This process is constantly evolving and will be further developed:

- By providing linkages to the Channels DB application, via the *alias/name* bridge and even to allow the online visualization of related DAQ channels.
- By providing linkages to server logs and configuration files.

Currently missing is a connection with Detector Monitoring - the application used in Virgo operation to understand the current condition of the interferometer. This *conditions* database [1], holding flags setting the best conditions of operation of the detector, may be linked to the HWII database once the channel information is made available through the Channel DB. This connection would enable the possibility to view snapshots of the history of condition flags in the web UI.

4 Use cases

In this section we present a few use cases to give an idea of what content can be stored in the HWII database, from the perspective of different subsystems.

4.1 DAQ chain

The DAQ subsystem structure is twofold: it contains a physical architecture composed of pieces of hardware (CPU machine, optical fiber network, ADC, Optical transceiver boards etc…) and pieces of software interconnected in an Optical&Ethernet network architecture. In both aspects, it is essential to have knowledge of the physical connectivity between each item to understand the different data flux in the detector. The optical fiber network in Virgo+ includes routes from the front-end ADC boards to the back-end CPU machine real-time processing, along various interconnected MuxDemux switches. A precise description of the different boards cited is stored in the HWII database. The hardware can be localized easily by browsing the integration chain. Every fiber connection has been registered. The link between two interconnected boards is also represented in the integration chain (see Fig.1). In addition, as presented in the section 3.3.3, we use the *alias/name* paradigm to link software-related information sources to machine network names. The DAQ topology is enriched in the web UI Integration Chain as shown in Fig.6, by adding a machine configuration link next to CPU machines. In this way, one can view which server is running on a given CPU machine together with the physical connection to other pieces of hardware (through the optical network). This information is very helpful in enabling more of an insight into the channel data flux, especially in understanding the difference between data travelling in the ITF control loops, from those going to the data offline storage line.

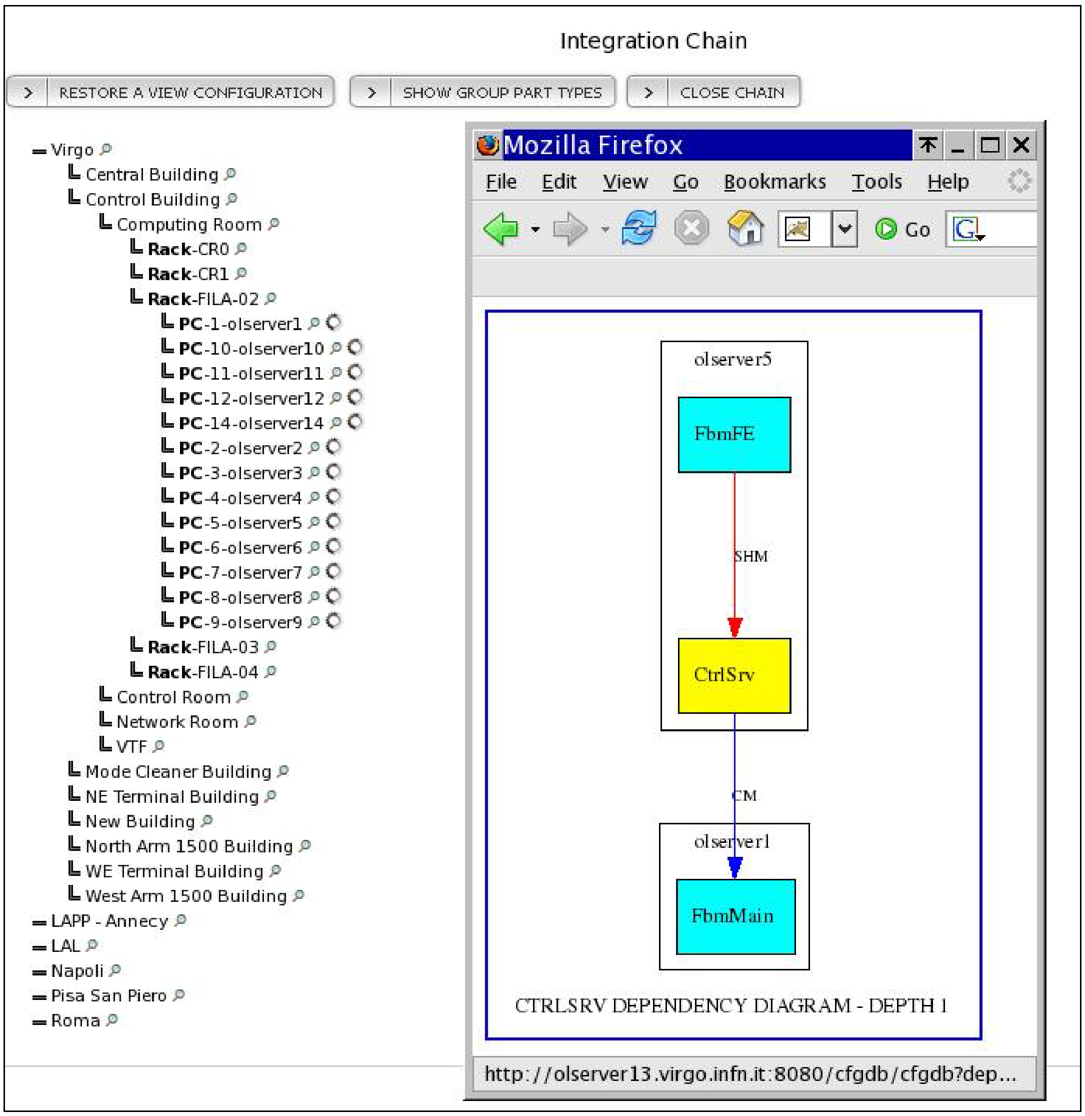


Fig.6: Example showing the HWII database as a single access point to different information sources. On the HWII web client, from the Integration Chain, by clicking on a specific link next to each CPU machine, one can browse the list of servers running, and the DAQ link between servers running on different machines. The latter information comes from the server configuration database.

As a result, from the DAQ point of view, the HWII database covers the needs of both the DAQ experts, providing a high level of detail for each of the DAQ nodes, and the operators, offering a single access point, bringing together essential complementary information needed to solve unexpected and not immediately understandable failures.

4.2 Vacuum

The Vacuum subsystem team requested the possibility to have a visual representation of the walls of each of the towers in order to be able to locate both viewports and blind flanges. Fig 7 shows an example of a tower wall showing the location of part instances of both of these types.

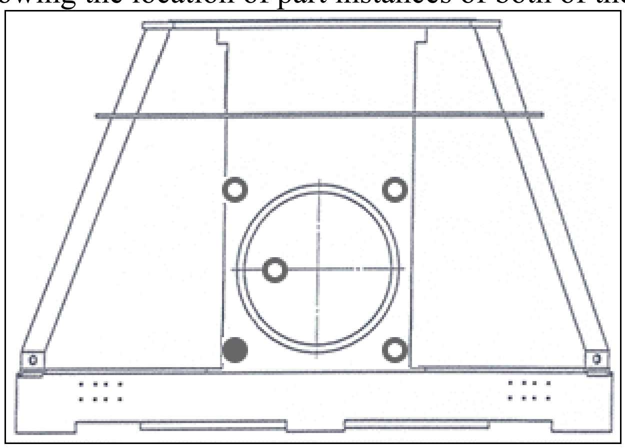


Fig.7: Visual representation of a tower wall showing both viewport and blind flange part instances.

Within HWII it is possible to re-locate these part instances by clicking on an individual part instance on the tower wall and then clicking on the new location. When a part instance is detached and/or attached to a wall image the operation is automatically recorded in the relevant part instance Worklog.

In addition to the specific image connected to the Vacuum part types, additional fields have also been defined as part of the Part Management for that group.

4.3 Optics

The Part Management concept emerged from an understanding of the specific requirements of the Optics subsystem team. They use the Part Management feature to have a quick overview of all part instances relating to their group – see Fig. 3. In addition, specific additional fields are added by the Group Administrator as required.

The Part Management view enables the Optics subsystem team to visualize information on their part instances in the same way as they were doing previously – using a static HTML page – without the need to maintain the code of that HTML page and worry about structural errors and problems viewing the page across different browsers. Instead all they need to concern themselves with now is the maintenance of the content.

4.4 Electronics

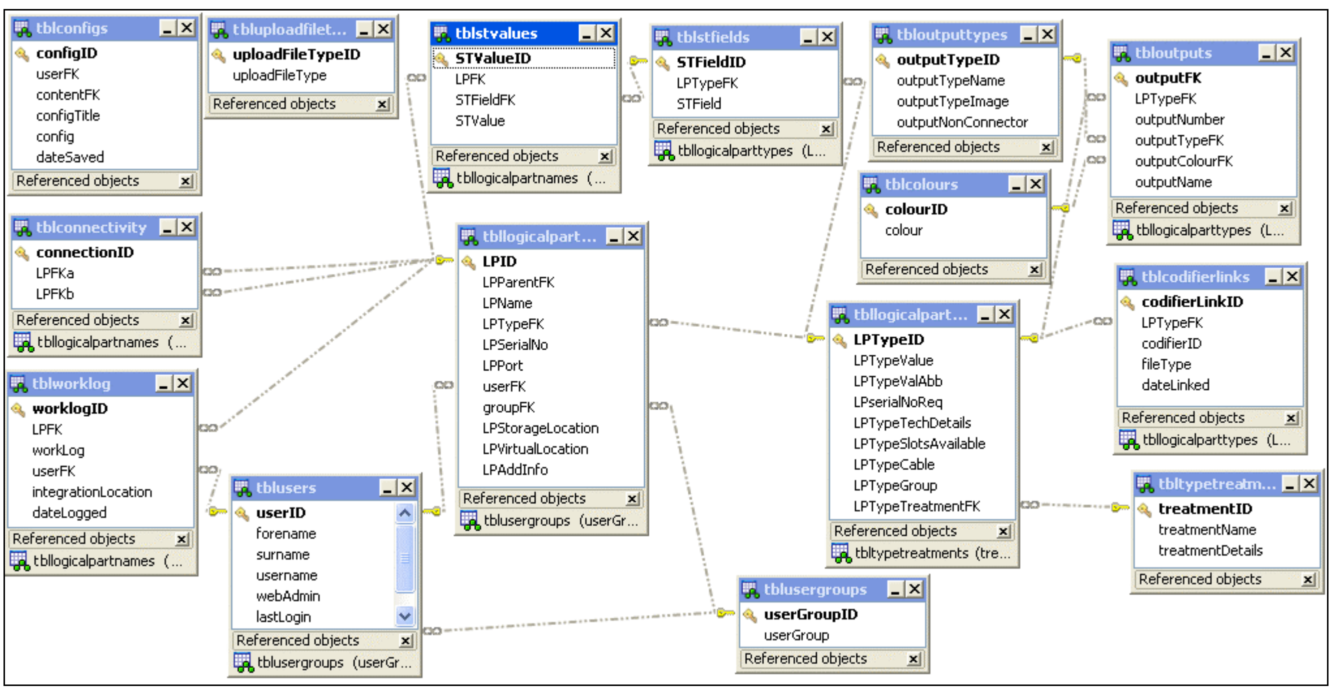
Based essentially on the model implemented in the Optics use case, the Electronics team also required assistance in the incorporation of information entered in pre-existent applications. As such, the development of a modular solution, allowing the definition by Administrators and Group Administrators of different fields and, importantly, field types, was important.

4.5 R&D Monolithic Suspension

Following an initial trial with a specific Special Treatment sheet designed specifically for use by the Monolithic Suspension team, a switch to the general extensible Part Management sheet was decided upon as this more than satisfied the group requirements.

It is interesting to note that compilation of the original Special Treatment sheet was deemed to be too laborious and required the inclusion of redundant information. Instead, the team took the design of the Monolithic Suspension Part Management sheet into their own hands and decided which content they should include as it best served their needs.

5 Annex 1 - HWII database table structure and relationships



___oOo___