# Towards the final ATLAS Pixel Detector Control System


Susanne Kersten [a], Francesco Bellina [a], Jenny Boek [a], Tobias Flick [a], Tobias Henß [a], Dirk Hoffmann [c], Kerstin Lantzsch [a], Peter Mättig [a], Olivier Pisano [c], Joachim Schultes [a], Sebastian Weber [a]

[a] University of Wuppertal, 42097 Wuppertal, Germany; [b] CERN, 1211 Geneva 23, Switzerland; [c] CPPM, 13288 Marseille, France

susanne.kersten@cern.ch



## Abstract

The innermost part of the ATLAS experiment is a pixel detector, built by 1744 individual detector modules. To operate the modules, readout electronics, and other detector components, a complex power supply and detector control system (DCS) is necessary. This includes a large number of crates, which house the different hardware components as well as a PC net where the different control projects are running. To test the final detector after its assembly before it is installed in the ATLAS cavern, a large test system has been set up at CERN, which allows to operate ca. 10 % of the detector in parallel. Since autumn 2006 this system is in permanent operation. As nearly everywhere the final control hardware is used, its reliability could be investigated and the performance of the control software could be studied. After an overview on our DCS hardware, we report on the experience with the control software.


## I. Introduction

The pixel detector is the innermost component of the ATLAS inner detector, which is used for the track finding. One main goal of the pixel detector is the reconstruction of secondary vertices.

To achieve the required track resolution the pixel modules consist of 46080 individual cells, each (50 × 400) µm. In the central region the detector modules are mounted on staves, which are carried by three shells, while in the end caps the modules are installed on disks. The DCS can act on units, which are made up by the detector modules, half staves and disk sectors, each containing six or seven detector modules.

A detector module consists of the sensor itself and 16 front end readout chips which are bump bonded to the sensor cells. The front end chips are supervised by the module control chip, which is mounted on a flexible hybrid circuit. This again is attached to the detector module. The further data transfer is handled by an optical transceiver system. The on-detector part, the opto-boards, are located close to the detector modules while the counterpart, the Back-Of-Crate (BOC) cards, are installed in the counting room ca. 100 m away from the experiment.

In the second half of 2006, a large system was set up at CERN, which allows to operate ca. 10 % of the final detector in parallel. Hardware of the production series as well as services very close to the final ones were used everywhere. Aim of the first period was the operation of the detector under cosmic data taking conditions. During this period we concentrated on verifying the compatibility of the various components with each other, and we collected important information about a safe and reliable operation of the detector via the DCS software. While the detector was assembled on the surface, the connectivity of the whole detector was tested stepwise before its installation in the ATLAS cavern. This put special constraints on the configuration of the DCS system and its communication to the DAQ (Data Acquisition) system.

## II. The Hardware of the DCS

There are some common requirements to all our supply and control hardware. Reliability has two major aspects: first the sensitive front end chips, which are produced in deep sub micron technology, can be destroyed even by short over-voltages and second part of the equipment won't be accessible over long periods. A good compromise was needed between costs and a high granularity, which gives optimal control and tuning for each detector module. All units should be easily integrated into the pixel and ATLAS control system using the same front end controller and common protocols.

Fig. 1 gives a simplified overview on the hardware of the pixel detector control system. Its main components are a complex power supply system, various monitoring units, an interlock system and the DCS PCs [1].

The power supply system consists of five major components. All of them have adjustable outputs, as due to irradiation the needs of the loads will change during the lifetime of the detector. To fit into the ATLAS grounding scheme all power supplies have floating outputs. The detector sensors are depleted by HV supplies built by iseg (Rossendorf, Germany). To operate the front end chips two low voltages are required, which are also delivered by a commercial power supply: PL512 from W.IE.NER (Burscheid, Germany). Multi channel current measurement units (LV-PP4) split one LV channel onto all modules of one readout unit and provide in parallel information about the power consumption per detector module. The low voltages are regulated closely to the detector by the remotely programmable regulator stations, which protect the front end chips against transients. The needs of the opto-boards are covered by the Supply and Control system for the Opto Link (SC-OLink), a custom made system, which is especially adapted to the requirements of the opto-boards.

Monitoring of the humidity and temperatures is performed by the Building Block Monitoring (BBM) and Building Block Interlock and Monitoring (BBIM) crates. While the first provides only monitoring values, the second one generates digital warning signals in parallel, which are sent to the interlock system.

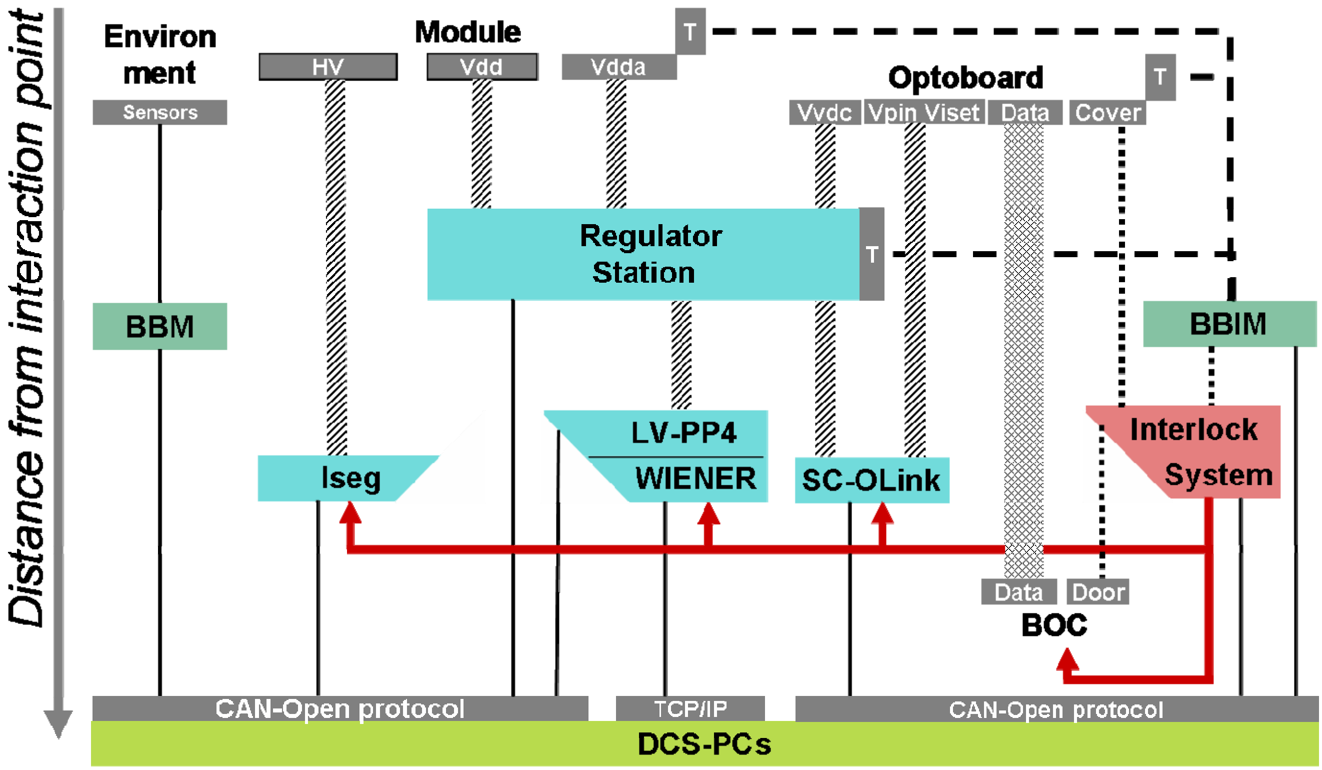


Fig.1: Overview of the DCS hardware

All power supplies and the off-detector part of the optical link, the BOC cards, are connected to the interlock system. This system protects the sensitive detectors and equipment against heat ups and handles the laser safety. To achieve a high granularity more than 1000 interlock signals are distributed. This allows to keep the number of detector modules switched off as low as possible. The interlock system is a completely independent hardware based system, which is just monitored and tested via DCS.

While the commercial HV and LV power supplies are coming with a CAN (Controller Area Network) interface, respectively a TCP/IP connection, all other systems are using the ELMB (Embedded Local Monitor Board) as the front end monitoring and control unit [2]. The CAN Controller of the ELMB is used for communication and provides the interface into the DCS PCs. The CAN protocol was chosen by the ATLAS community for its robustness and reliability.

As the 10% system is using production units and no failures of the hardware were reported during the operation of eight months, we have a lot of confidence in the reliability of our commercial and custom made hardware.

## III. The Software of the DCS

### A. *Overview*

All hardware is controlled by four PCs, which collect data from the various sub systems via CAN or TCP-IP and the related OPC (OLE for Process Control) servers.

Following the LHC wide standard, most of the DCS software is built using PVSS (Process Visualization and Steering Software), a commercial product from ETM (Eisenstadt, Austria). Its concept is based on data-points and managers. Data-points and data-point elements, defined by the user according to the needs of the experiment, provide all tools to store and handle control data. The managers are the processes which supervise data taking, data storing, or reaction to special events. One dedicated manager provides the graphical interface to the user. PVSS offers the possibility to split the main projects into several components, which can easily be prepared by a group of developers in parallel.

The DCS software (see Fig. 2) can be grouped in different layers. The bottom layer is built by the front end integration tools (FIT), which establish the communication to the hardware. The related OPC clients are part of these projects. The system integration tool (SIT) provides the mapping between the hardware oriented organisation of the PVSS data-points and a geographically oriented view of the detector. On top of it the finite state machine (FSM) provides the standard user interface to operate the detector. Furthermore connections to the databases and to the data acquisition (DAQ) system are required.

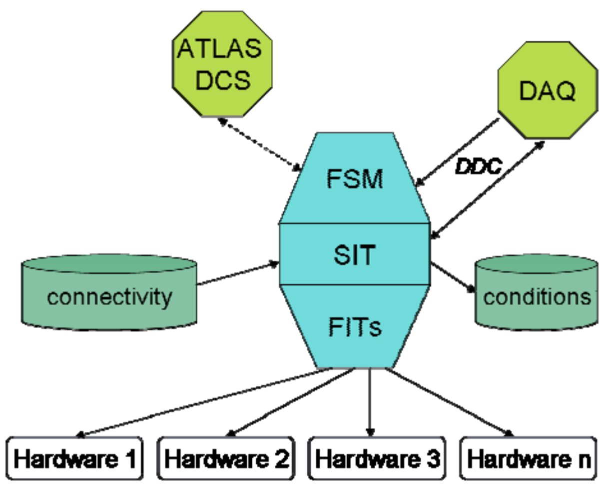


Figure 1: Overview of the DCS software

To handle the different versions a CVS based solution is used. This enables the distribution of the development versions for debugging and test purposes. A stable release is then distributed to the production systems by creating a package and using the functionality of the fwInstallation [3].

## *B. The Front End Integration Tool*

To integrate the devices of the hardware components into the control software, the Front End Integration Tools were developed. They provide a functional structuring of the various devices according to the specific properties of e.g. a power supply or a temperature measuring unit. As the FITs are independent of the detector structure, it is possible to develop them independently for each device type. There are the FITs for ELMB based devices (like BBIM, Regulator Station), for the W.IE.NER and for the iseg power supplies.

Each FIT consists of two parts. To define and manage the various units, the integration part is used. It handles the information such as communication parameters or the type of device. All these information are used to establish the integration into the control software, such as creating the internal data structure, defining the communication details and the conversion of the raw data. In addition, part of the external configuration of the system is handled by creating the necessary configuration files (e.g. for the OPC servers).

A specific Watchdog service monitors all integrated devices in a separate process for safety reasons. It is responsible for the notification in the case of a communication failures and summarizes the information for the higher level Watchdog.

A FIT also provides the necessary graphical user interfaces for the operation of the integrated devices inside the control part. It is mainly used by the DCS expert for debugging purposes. Therefore these panels provide all available information of the devices and enable their operation and detailed tuning.

There are no severe problems with the FITs to report. With typical refresh rates of 5 to 6.5 s also the performance of the OPC server and clients was satisfactory.

## *C. The System Integration Tool*

The complexity of the system and the high number of channels used for operating the setup creates the necessity of organizing, grouping and displaying these values in a way that the operator can handle.

In order to achieve this, the System Integration Tool (SIT) adds “aliases” to the data-point elements used by the various FITs. These aliases act as a second name or synonym for those channels, reflecting the channels task and position within the detector control system.

After those aliases have been extracted from the connectivity database (see section III.E) and applied to the FIT’s channels, they are used in order to provide an easily understandable graphical user interface for the shift crew.

Instead of displaying sets of unrelated high or low voltage channels, these graphical user interfaces provide data monitoring and control facilities following the detector’s “geographical” layout.

In a tree-like, hierarchical structure, the user can navigate down from detector partitions to readout units and even detector modules, each being represented by panels that combine all necessary information for this instance.

In contrast to the former way of handling geographically organized data, by copying it into a separate data-point structure that follows the detector layout, the presented solution, using aliases, has proven to be very effective. Amongst the most notable advantages are the minimal CPU load imposed by the SIT and the elimination of unnecessary data transfer operations between the different PVSS systems.

Specially during the connectivity tests, when the cabling was changed frequently, the SIT was used to reconfigure conveniently the whole DCS. As the construction of the finite state machine and the functionality of the communication packages (see section III.F) rely on the structure provided by the SIT, also these projects could follow changing connectivity comfortably.

## *D. The Finite State Machine*

To allow the operation of the pixel detector, a clearly structured overview of the status of its individual components and services is needed. There must be the possibility to easily change the state of the detector or parts of it. Following the ATLAS wide decision, a Finite State Machine is used, which evaluates and summarizes the status of the detector. Objects, states and commands are the three ingredients to build a FSM, the pixel detector specific ones are described below. For the creation of the FSM the framework component "controls hierarchy" is used [4].

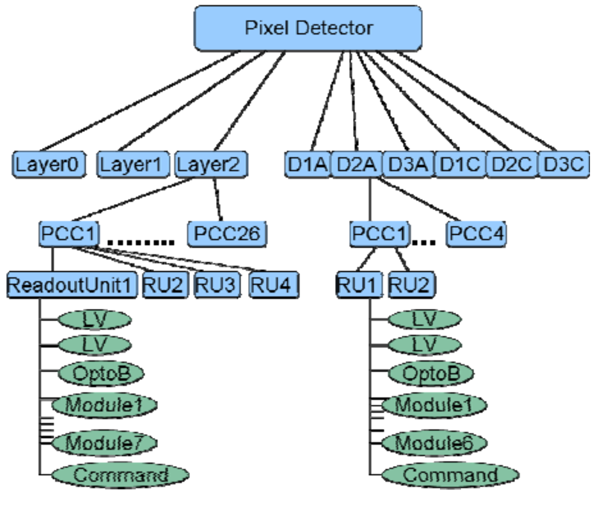


Fig. 2: Overview of the DCS software

The FSM, which was used during the 10% test, was built by two kinds of objects, Device Units and Control Units, see Fig. 3. The lowest level of the FSM consists of three types of Device Units: LV, Opto-Board and Module. While Opto-Board summarizes the voltages and currents of the opto receiver components, Module depends on the temperature, the high and low voltages and their currents of each individual detector module. LV represents the main LV power supply channels, which provide their outputs per readout group. The lowest level Control Unit is the Readout Group, which is the smallest unit regarding optical data transfer and consists of two LV, one Opto-Board and 6/7 Modules. The next level is the Parallel Cooling Circuit (PCC), which is the smallest unit regarding the cooling. The uppermost Control Units (below the detector level) are the 'layers', which consist of the three disks per end cap and the three shells of the barrel part.

Inside the ATLAS FSM state and status describe the objects. The state of an object informs about its operational mode and the status of an object gives more details about its performance. While the status is fixed by ATLAS DCS to be only ok, warning, alarm and fatal, the states are defined by the sub detector needs.

The state and status of the device units are depending directly on one or more PVSS data-point elements, i.e. actual measured values. They are calculated by a timed function (interval 5 s) to take into account that not all values are updated simultaneously and therefore a direct reaction on change of one out of many parameters is unwanted. Modules and Opto-Boards can be in the states: running, ready, not-ready, off, unknown and disabled, while LV is either on or off. The state and status of the Modules are mainly given by the values of the currents of the different channels. Not all states can be reached by actions of the DCS, the running state can only be achieved by a DAQ-action. Unknown was used for any combination of values that would not fit into the defined states.

The states of the Control Units do not depend on data-point element values but are calculated by the State and Status of their children according to user-defined 'when-lists' of the FSM.

To trigger the transition from one state to another a set of commands is available. They are propagated down the hierarchy, while information about the change of the state is propagated and combined upwards up to the top level. Besides the switch-on, switch-off, recover and reset commands, there is also the disable command. Like the standard partition mechanism of the FSM, the state of a disabled object is not propagated to the higher levels, but different to the standard behaviour it guarantees that a disabled object stays switched off. An additional Device Unit 'Command' is used to synchronize the order in which the commands for a Readout Group are executed.

The detector oriented organisation of the Finite State Machine turned out to be very useful, as it is intuitive for the operators. The granularity down to the module level delivered easily understandable Readout Group states and was very helpful during the connectivity tests.

With ca. 15 % average load on a standard PC, the performance of the 10 % detector FSM was sufficient. The speed of command execution depends on delays which are used to guarantee the correct order of commands. As up to 192 regulator channels are controlled by one ELMB, the possibility to parallelize commands is the limiting factor here.

We learnt that a controlled power sequence avoids undefined states of the Readout Group. The possibilities of the Command Device Unit should be expanded to guarantee a proper and safe power up sequence of the detector.

The unknown state should in the future be used for the more intuitive 'loss of communication', a state undefined will be introduced to replace the old unknown state.

### *E. Database Connections*

Handling of the comprehensive volume of input and output data of the Pixel DCS is performed by means of databases. During the 10% test two distinct data streams were used for the following functions:

- Setup of the hardware connectivity map (input at start-up)
- Logging of condition data (continuous output)

All data streams are connected to the ATLAS Online-Server, which is running Oracle RAC [5] software. Like the whole ATLAS online infrastructure, it is connected to internet through a restrictive firewall. We give details on the two use cases in the remainder of this section.

Connectivity data are used jointly for the setup of DAQ and DCS for the sake of coherency and integrity of a given setup at a given time. A program written in PHP (multi-platform hypertext pre-processor) translates data from native SQL (Structured Query Language) tables into an XML (Extensible Markup Language) format that is understood by the DCS and contains the declaration of more than 4000 data-points. The use of PHP allows us to retrieve the XML file by means of a universal resource locator (URL), which can indifferently point to a file or a web server. The use of XML as interface layer eases debugging and quick modification of minor points in the setup for tests and commissioning. Any XML file extracted from the database will be cached locally for future reference and avoids dependency from local IP availability.

Some 13k measurements of environmental and detector conditions are produced permanently by PVSS. They are compressed by an ETM proprietary algorithm and logged into an Oracle instance, thus producing several GB per day. Recording has been performed over several months without significant problems.

A general-purpose application for condition data display has been developed in the context of this project. This application selects and extracts logged data from the database, generates graphical displays and analyses and exports the results into various formats. An effort has been made to keep it as generic as possible, thus allowing access to any server and even for data of all ATLAS sub-detectors, which will become interesting once the experiment will have been completely assembled in one place and data need to be correlated.

### *F. Communication with the DAQ System*

The detector control and the data acquisition system are two independent entities. However the operation of the detector requires coordinated procedures of DAQ and DCS. Further calibration and tuning algorithms – in particular for the optical link - need simultaneous access to the hardware controlled by DCS and the data collected by DAQ.

To fulfil these needs the DAQ DCS Communication (DDC) package [6] provided by the TDAQ group of ATLAS was adapted to our requirements and intensively used. It provides tools for exchange of data and commands between both systems, see Fig. 4. The data consist of the FSM state of the detector or its components and monitoring values like currents and temperatures for tuning purposes. On the other side, certain control parameters of the BOC cards, which are accessible only by DAQ hardware, must be transmitted to DCS. Commands, which trigger FSM actions or just change operation parameters of a single channel, should only be sent from DAQ to DCS.

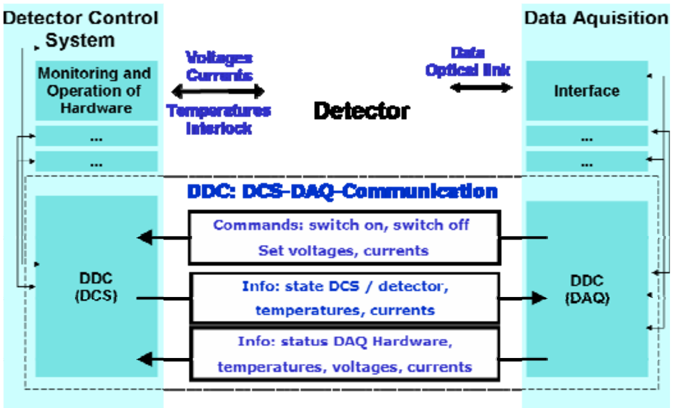


Fig. 4: Overview of the DCS software

The mechanism of DDC is based on the Distributed Information Management (DIM) [7], a general server client protocol for data and command transfer between different platforms and applications. DIM servers publish the value of application variables using a unique name as identifier. Clients can retrieve the value once or subscribe to it to be provided with new values in a time based way or on change. In addition there are published commands, where variables are writeable by clients. A command with a feedback, a published Remote Procedure Call completes the facilities of DIM.

DIM can directly connect to DCS, respectively to the PVSS internal database consisting of data-points. A script generates a list of data-points whose content should be transmitted or received as well as data-point structures for commands, all with a unique name. For command handling a script is connected to the associated data-point structures as a callback function. Received commands are parsed and a feedback of the (un)successful execution is returned.

The command transfer was intensively used for sending commands from the DAQ side to DCS and its FSM. This ensured the synchronized operation of DAQ and DCS and it worked reliably.

Specially in the second period of the 10% test, when the connectivity was checked, the command transfer was also used for a dedicated tuning of the optical link, e.g. the parameters of the transmitting laser diodes of the opto board were varied via DCS while the receiving components on the BOC card were monitored and bit error rates were recorded. The command transfer worked, however besides a planned parallelization of commands a general improvement of the speed of this method on the DAQ side is investigated at the moment. Command execution on the DCS required just 15 ms, which should be fast enough for operating the detector inside ATLAS.

The data transfer mechanism was applied for sending data from DCS to DAQ. However also the other direction, where monitoring data of the BOC cards, like diode currents and temperatures, were sent to the DCS, was tested. As performance tests have shown on the DCS side DIM for PVSS is the bottleneck for data transfer. About 4000 values can be transmitted per second. Assuming a peak changing interval of five seconds per value, this would give a maximum of 20000 values which could be handled correctly. This should be sufficient for the pixel detector's needs.

## IV. SUMMARY

A test setup, which allowed operation of up to 10 % of the final pixel detector in parallel, has been built and intensively used for more than eight months. While the pixel detector went through the final tests before installation, we could in parallel gather important experience with the control system.

Components of the final production were used for the hardware everywhere, which have not shown any severe errors. The software is split into several projects, with the front end integration tools and the system integration tool providing the base. As the performance of these software components was satisfactory, we do not expect large changes when moving to the final, 100 % control system. Especially the flexibility provided by the SIT turned out to be very useful as during the connectivity tests a frequent re-loading of the mapping was required.

The detector oriented organisation of the Finite State Machine is intuitive and found wide acceptance in the operator community. Therefore the same concept will be followed for the final FSM, however more objects, which e.g. describe the infra-structure and environment, need to be included. The possibility to execute commands in a well defined order avoids risky conditions for the detector as well as a confusing behaviour of the FSM for the operator and will be expanded in the future.

The connectivity database and its interfaces are very close to the final version, the handling of the conditions database was satisfactory. The PVSS dataviewer turned out to be a very useful tool to analyse the contents of the conditions database.

Essential for the operation of the pixel detector is a synchronized operation of DCS and DAQ, which is provided by the DDC package. Besides that DDC has an important role for the tuning of the optical link.

All collected experience will find its way into the preparation of the final DCS software, when we move now from a 10 % system to the 100 % detector.

## V. REFERENCES

[1] T. Henß et al.; The Hardware of the ATLAS Pixel Detector Control System, 2007 JINST 2 P05006
[2] http://elmb.web.cern.ch/ELMB/ELMBhome.html
[3] http://itcobe.web.cern.ch/itcobe/Projects/Framework/Download/welcome.html
[4] A. Barriuso Poy, S. Schlenker https://edms.cern.ch/file/685114//FSM_INTEGRATION_GUIDELINE.pdf
[5] B. Lundhild, P. Sechser; Oracle Real Application Clusters 10g, Oracle Technical White Paper (May 2005)
[6] http://atlas.web.cern.ch/Atlas/GROUPS/DAQTRIG/DCS/DDC/ddc.html
[7] http://dim.web.cern.ch/dim/