THE DIAMON PROJECT – MONITORING AND DIAGNOSTICS FOR THE CERN CONTROLS INFRASTRUCTURE

Pierre Charrue, Mark Buttner, Joel Lauener, Katarina Sigerud, Maciej Sobczak, Niall Stapley,
CERN, Geneva, Switzerland

Abstract

The CERN accelerators' controls infrastructure spans over large geographical distances and accesses a big diversity of equipment. In order to ensure smooth beam operation, efficient monitoring and diagnostic tools are required by the operators, presenting the state of the infrastructure and offering guidance for the first line support. The DIAMON project intends to deploy software monitoring agents in the controls infrastructure, each agent running predefined local tests and sending its result to a central service.

A highly configurable graphical interface will exploit these results and present the current state of the controls infrastructure. Diagnostic facilities to get further details on a problem and first aid to repair it will also be provided. This paper will describe the DIAMON project's scope and objectives as well as the user requirements. Also presented will be the system architecture and the first operational version

MOTIVATION

The controls infrastructure of the LHC and its whole injector chain spans over large geographical distances and is deployed using a big diversity of equipment. In order to allow for the best availability of the controls infrastructure, all these controls parts of the chain need to be constantly monitored. And in the event of a problem detected it has to be notified to the Control Center and means to repair it has to be proposed.

The purpose of the DIAMON project is to propose to the operators and equipment groups tools to monitor the AB Controls infrastructure with easy to use first line diagnostics and tools to solve problems or help to decide about responsibilities for first line of intervention.

The scope of the project covers around 3'000 agents, piece of code that monitors a part of the infrastructure, ranging from the hardware of the consoles, the back-ends, the front-ends, the communication packages, the field-buses, the application software, etc.

The DIAMON project is designed in two main parts, the monitoring part that constantly checks all the items of the controls infrastructure and reports about problems and the diagnostic part that displays the overall status of the controls infrastructure and proposes support for repair.

REQUIREMENTS

The DIAMON project has issued a first version of the requirement document. The main demands were :

- the monitoring part should not hamper the normal operation of the tested item. Its footprint should be minimal and its resource consumption as low as possible
- the diagnostic display GUI should offer different ways of viewing the monitored items
- the LASER [1] project should be used for the upwards communication
- the downwards communication should be based on lightweight and short-lived connection system
- The GUI should propose 'hints' to solve a problem or give links to documentation or display the contact person responsible for that specific problem
- Provide standard diagnostic tools

THE DIAMON ARCHITECTURE

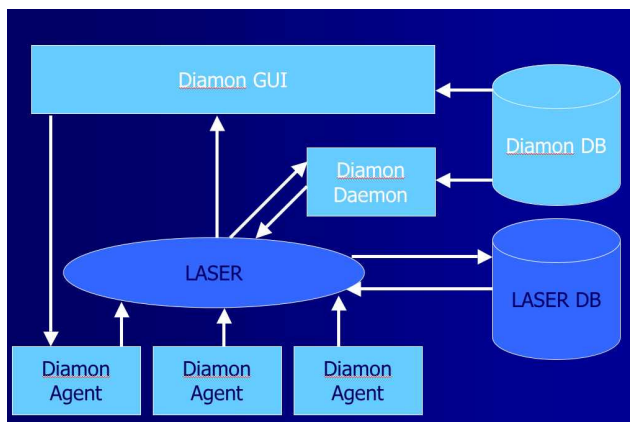

Figure 1: DIAMON architecture.

THE AGENT

The agent (see fig.1) is a component designed for:

- Testing system parameters
- Sending regular updates to DIAMON
- Informing DIAMON about errors
- Processing commands from the GUI

The agent also provides (non-persistent) response to "get detail" command and keeps a short term history of test results.

Clic agent

The CLIC agent monitors the operating system parameters (such as free memory, CPU load, disk space, system resources) as well as specific items such as user process or specific hardware I/Os.

WFIP agent

The WFIP agent monitors the WorldFIP field bus that is heavily used for connecting to LHC devices.

Timing agent

The Timing agent verifies constantly the UTC clock distribution and reports any deviation. It also monitors the correct distribution of the accelerator machine timing.

CMW agent

The CMW agent monitors the usage and the good functioning of the Common Middleware infrastructure.

Future agents

Other important parts of the controls infrastructure will be monitored by agents in the coming months, such as the JAVA software applications, the FrontEnd VME hardware (Power supply, fan speed, temperature control, …), etc.

THE COMMUNICATIONS

The agents are considered as LASER sources and send their messages using LASER as the communication subsystem. The decision to rely on LASER allowed to easily reuse the important functionalities like archiving, statistics or configuration. We can also benefit from the fact that LASER is already supported on the wide range of platforms and it is considered to be reliable and well tested. The inherent limitation of LASER is that it naturally supports only one-way communication, which in our case allows to send monitoring reports from agents to central broker and other consumers.

As a complementary solution for downward communication, which is necessary to support the diagnostics abilities of the system and the basic repair commands, a lightweight communication library is used that supports RPC-like point-to-point message exchange. An important feature of this lightweight protocol is that it does not rely on any other communication components (like brokers), which makes it particularly viable for diagnostics and troubleshooting.

THE DIAMON CONSOLE GUI

The first goal of the DIAMON console GUI is to provide a general view of the control system's state (monitor) and guide the user in the resolution of problems (diagnostics).

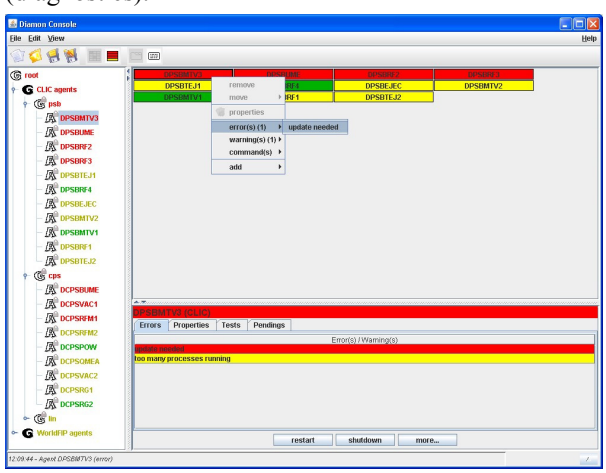

Figure 2: DIAMON console GUI.

As each user might want a different view of the control infrastructure, the DIAMON console GUI is highly flexible. It offers the following:

- Ability of creating custom or standard filtering, grouping and sorting of the agents.
- Users can create their own configuration or simply use the default one.
- Generic views are available but agent provider can easily write their own specific view to be used to display their monitoring data.
- Standard diagnostic support are available but here again specific diagnostics tools can be incorporated by the agent provider.
- The layout of the views can be re-organized.

THE DIAMON DAEMON

The DIAMON daemon is a software component running as a background task listening to all messages coming from the agents. It can therefore automate any action related to the monitored information, in particular:

- Statistics: the daemon will at anytime be aware of the state of the agents connected to DIAMON. This makes it possible to provide statistics about the "up" (status OK) and "down" (status ERROR) times of the monitored equipments.
- Routing: Whenever other systems (for instance the alarms system) are interested in receiving information initially collected by DIAMON, the daemon can produce messages to these systems. This functionality will use a routing table located in an Oracle database. The main interest of this feature is to avoid to information providers to send the same information to various systems. The idea is to send it to one (DIAMON), which informs others if necessary.
- Logging: the daemon will communicate DIAMON internals to the Logging system. This feature does not cover logging needs from our agent providers, who should implement their logging interface directly.
- Automatic repair actions: For given problems, it is possible to trigger actions without human intervention (a process missing on system ? ... restart it !). The DIAMON daemon will give this possibility even if such behaviour is not suitable in many cases. An automatic repair has the side effect that it transforms a failing system into one without apparent error, and inducing the risk that nobody ever tries to fix it.

The DIAMON daemon will also act as an agent, i.e. monitor the behaviour of DIAMON as a whole. To ensure the proper working of DIAMON, the daemon must be active. It will communicate its state to the GUI and respond to a set of diagnostics commands.

IMPLEMENTATION DETAILS

The DIAMON library allows writing agents, which are software components capable of sending monitoring notifications as well as receiving diagnostic commands.

The implementation details as well as some strategic choices are hidden behind a very simple interface, which allows users to integrate their programs with DIAMON with as little work as possible. The current version of the library exposes its interface for use from C++ programs, which fits our usual deployment model of using C++ for server-side and front-end development, where most diagnostics agents are supposed to be implemented. On the other hand, we foresee extending the technology coverage for Java as well to allow easy embedding of DIAMON agents into software components running in the middle-tier of our controls complex.

The DIAMON library is also used by the set of generic tools that the DIAMON Core Team provides for general use. The generic tools were designed to interface with any program that communicates via standard I/O according to simple conventions. It is expected that most of the monitoring activities related to system-level assets (like the list of processes, memory and CPU usage, power supply's working conditions, etc.) can be implemented in terms of shell scripts that are triggered by these generic tools and that only users with specific needs or having specific constraints will have to use the library directly, either by writing separate components on top of the library, or by embedding it in their programs, where the diagnostics and monitoring activities can be integrated with the already existing main event loops. In addition, the generic tools are implemented on top of the DIAMON library and as such allow us to quickly validate its design and implementation without involving final users of the whole system.

The DIAMON library provides both monitoring and diagnostic functionality.

FUTURE PLANS

A first operational version is available today collecting data from clic, timing, WFIP and CMW. The main features of this version are:

- Monitoring of equipment based on the above information providers.
- GUI with sorting, filtering and grouping of monitored equipment
- Support for basic diagnostics commands
- RBAC based security scheme implemented on the GUI side. Menus reflect the access rights of the user.
- Error routing to the LASER alarms system

A second version with more agent covering the whole injector chain will be made available for the PS and SPS startup in March 2008, including also:.

- Links to relevant documentation and responsible persons
- Support of more diagnostics commands
- Support of “expert” plugins in the GUI
- Test configuration using a central database

And the final version will be deployed for the LHC beams in the course of 2008.

REFERENCES

[1] http://service-alarm.web.cern.ch/

[2] http://wikis.cern.ch/display/DIAMON/