The DIAMON Project
Monitoring and Diagnostics for the CERN Controls Infrastructure

Pierre Charrue, Mark Buttner, Joel Lauener, Katarina Sigerud, Maciej Sobczak, Niall Stapley
Accelerator and Beams Department - Controls Group
CERN - European Organization for Nuclear Research - Geneva, Switzerland

2007 International Conference on Accelerator and Large Experimental Physics Control Systems
ICALEPCS '07

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

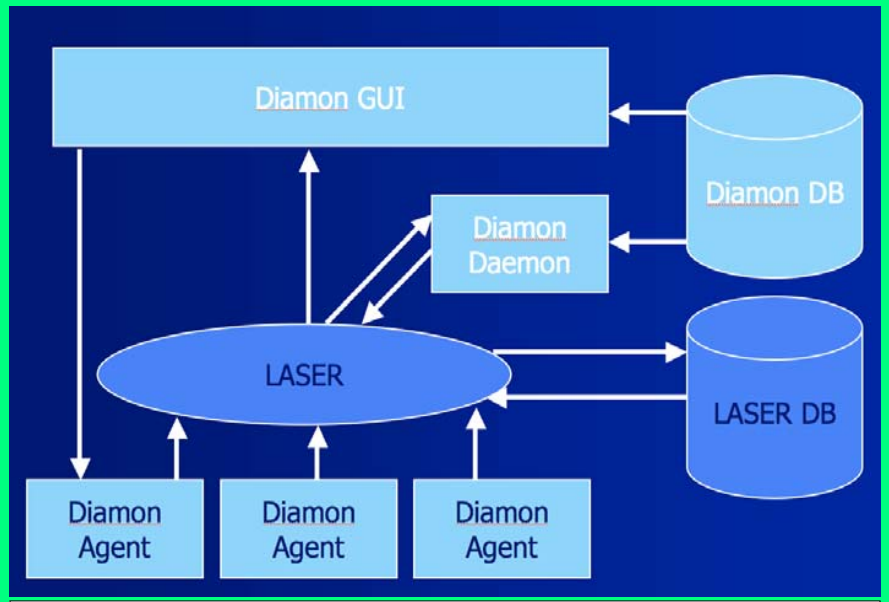
The **agent** is a component designed for:

- 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.

```
#include "Diamon.h"
#include <iostream>
#include <string>

using namespace std;

/***
 *      Main
 */
int main(int argc, char *argv[]) {

// Set Ids and standard variables for the communication with Diamon
        const string agentType = "replace with your agent type";
        const string agentInstance = "replace with your agent id";
        Diamon::SystemState state;
        Diamon::DetailedReport details;

// Take a parameter from the command line and set the Diamon status accordingly
        if (argc < 2) {
                cerr << "Usage: ./HelloDiamon 1|-1 for OK|ERROR\n";
                return -1;
        }
        int iState = atoi(argv[1]);
        if (iState == 0) {
                cerr << "The state parameter is invalid (can only be 1 or -1)\n";
                return -1;
        }
        if (iState == 1) state = Diamon::OK; else state = Diamon::ERROR;

// Initialize the Diamon environment (done once for all)
        Diamon::monitoringInit(agentType, agentInstance, false);

// add some additional data to the "details" and send an update to Diamon
// this step could be repeated as needed
        try {
                details.push_back("Error forced by agent ...");
        Diamon::publish(state, details);
        } catch (const Diamon::DiamonError &e) {
                cerr << "Error while contacting DIAMON: " << e.what() << "\n";
        }
        return 0;
}
```

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.

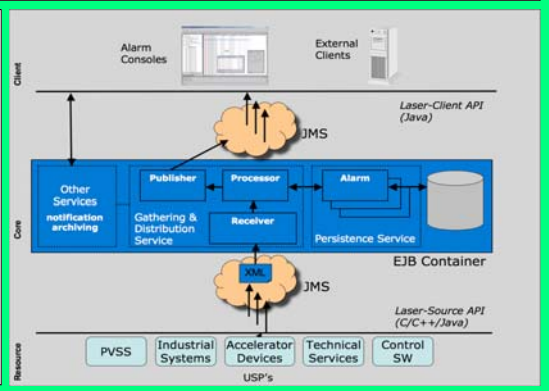


The DIAMON daemon is a software component running as a background task listening to all messages coming from the agents. It can therefore automate any action related to the monitored information, in particular:

- Statistics
- Generation of LASER alarms for specific DIAMON errors.
- Logging of all DIAMON events (using log4j)
- Automatic repair actions for specific DIAMON errors

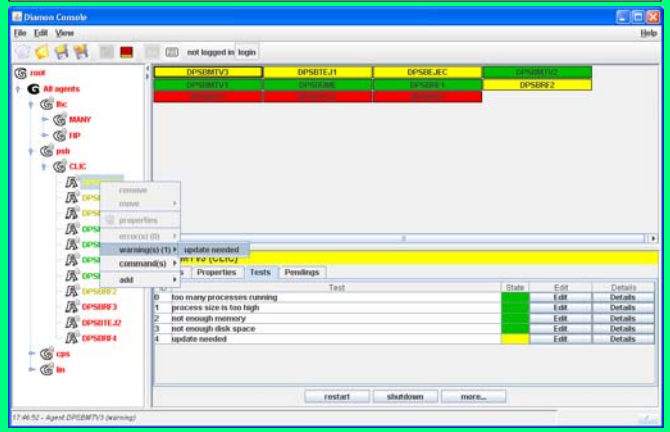
LASER (LHC Alarm SERvice) is an alarm system that accepts alarm events from all around CERN, processes them and presents the results to operators and other software. It supports several services that DIAMON would have had to do itself which instead is delegated to LASER such as :

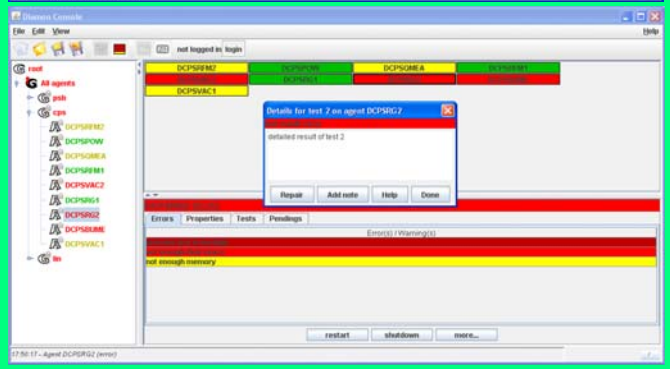
- Communication (using asynchronous messaging)
- Archiving and historic retrieval of events
- Surveillance of the input processes, with failure notification
- Flood prevention: protection from sudden massive input
- Data definition for monitoring points



The **DIAMON GUI** is an easy to use and highly configurable graphical interface.
It displays ALL agents selected and allows for :

- getting additional details
- performing standard diagnostic commands (repair, reboot, …)





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.