Getting all required information for a grid site is complicated.

Monitoring information is not clearly arranged, there are:
- Many sources of valuable information
- Different information displays provided by different technologies

Totality of all monitoring systems is uncomfortable to use, you have to:
- Manage many browser tabs / windows
- Change the settings of the web interfaces (time range, site, ...)
- Long waiting time until page opens up, often more than 30 seconds

Consequences:
- Unnecessary increase of administration effort for a grid site
- Difficult to identify correlations
- Nearly impossible to get a quick overview on a site’s status for non-experts, especially if several services at different sites are involved

Idea to ease administration: Meta-monitoring

Such a framework should:
- Collect and process all important monitoring information
- Present the current status of a grid site and its services
- Display simple rating / warning system (smiley faces, arrows, ...)

Design properties:
- Framework has a modular layout: There is a static core that provides the basic functionality for the dedicated tests. The individual tests can be plugged in.
- Decoupling of collecting the information and the actual visualisation
- All information is accessible via a single website, including a history
- Visualisation should provide a smart and quick overview on the monitored service which also allows to identify correlations

The HappyFace Project provides such a smart summary of existing information

User centric real-time grid job monitoring for the WLCG

To deal with failures of user jobs on the LHC Computing Grid (LCG), the University of Wuppertal has developed the Job Execution Monitor (JEM), providing new possibilities to find problems in a largely distributed computing grid and to allow for analysing these problems in nearly real-time. JEM was developed primarily to aid physicists using the software framework Athena to simulate and analyse collision events of the ATLAS detector, one of the four big experiments at the Large Hadron Collider (LHC) at CERN, Geneva.

JEM monitoring functionality

Data gathered by the Job Execution Monitor includes:
- Detailed execution flow in shell and python scripts, as well as in compiled binary modules by means of a novel technology named CTrace.
- Periodic measurement of worker node system metrics like CPU- and RAM-Usage, network traffic, disk usage, etc.
- Monitoring of job progress: Job state, end- and next physics event set coordinates.
- Real-time peeking in the job’s (stdout-) output

Using JEM, the user is able to trace problems in Athena environments, helping functions and in the physics analysis itself during the job’s execution. For the case of use, JEM has recently been transparently integrated into ATLAS grid job submissions and management-tool Ganglia.

Architecture of the monitoring software

The Job Execution Monitor is a distributed application, implemented in python, consisting of two main parts. One part is run on user jobs on the grid E machine (the machine used to submit user jobs to the grid) and the other part is submitted alongside user jobs to grid computing elements (CEs).

To ease the user in the analysis of monitoring data, we developed a log viewer application providing useful functions like scrolling the data, sorting and filtering, reviewing of user processes, shell and application graphs, etc.

All monitoring data gathered during the job run is transmitted nearly in real-time to the user machine to allow for direct analysis, as well as for post-mortem job failure analysis. Even if all of the jobs’ output was discarded by the grid middleware as a result of the failure.

Using JEM, valuable grid resource usage can be optimized:
- By aborting, fixing and re-submitting faulty jobs as soon as an error is discovered, as opposed to after the job finished execution (possibly after hours of wasted CPU time)
- By finding the reason for jobs hanging / never completing
- By discovering the cause of job crashes.