

Meta Monitoring

The HappyFace Project – Summary & Outlook

Volker Büge, Viktor Mauch, Günter Quast,
Armin Scheurer, Artem Trunov

Institut für Experimentelle Kernphysik, Steinbuch Centre for Computing



Forschungszentrum Karlsruhe
in der Helmholtz-Gemeinschaft

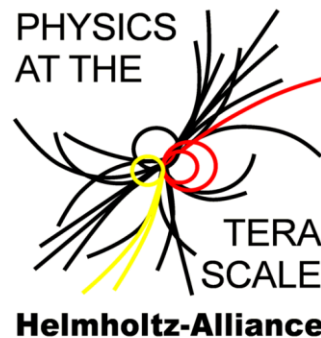


Universität Karlsruhe (TH)
Research University • founded 1825

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



■ Introduction

- Overview
 - Central WLCG Monitoring
 - Experiment Specific Monitoring
 - Local Site Monitoring
- Why do we need a meta monitoring system
- Ideas for Meta Monitoring

■ The HappyFace Project

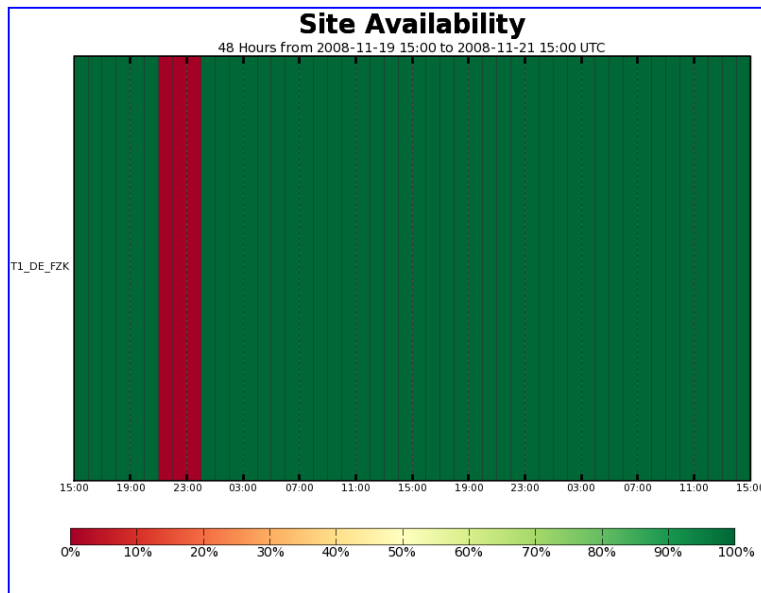
- Already used by several sites
- Current development
- Workflow
- Object-oriented structure

■ Summary

- **Distributed Grid resources are complex systems.**
- **To provide a stable and reliable service, monitoring of the different services and resources is indispensable.**
- **Use monitoring to discover failures before the actual services are affected.**
- **Different types of monitoring:**
 - **Central monitoring of all Tier centres**
 - Status of the high level grid services and their interconnection
 - Availability of the physical resources of the Tier centres.
 - **Central monitoring of experiment specific components**
 - **Local monitoring at the different computing sites**
 - **Local monitoring of experiment specific components**

Many different monitoring instances for different purposes!

Central WLCG Monitoring



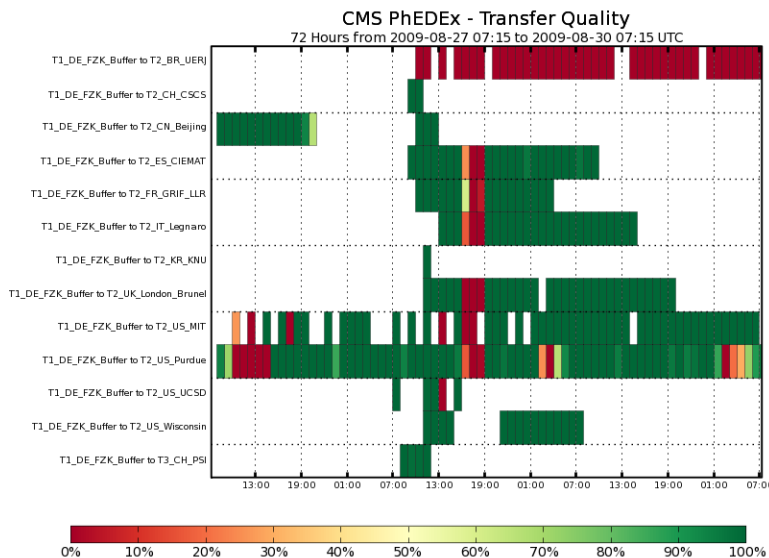
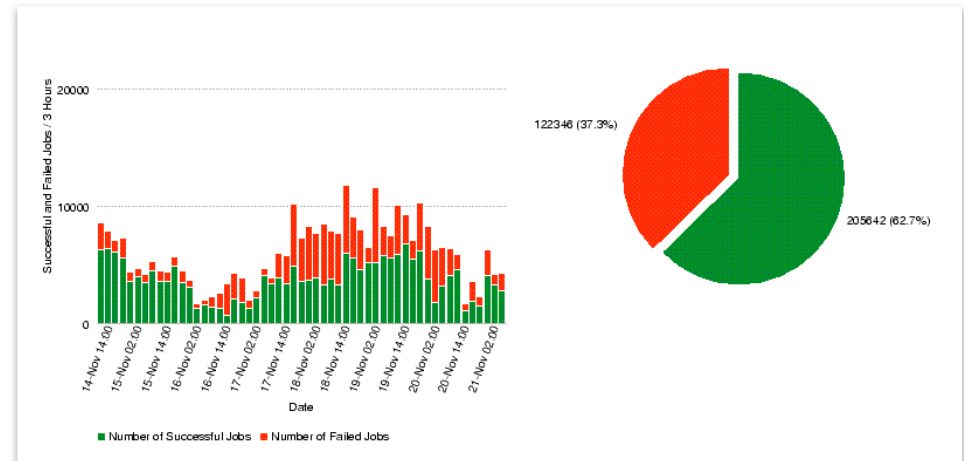
- **Service Availability Monitoring (SAM)**
 - Standardised Grid jobs, testing the basic grid functionality of a Tier centre.
 - Includes all Grid services of a site.
 - Centrally sent to the Tiers (once per hour)
 - Output available on a web page with history

- **Once a critical SAM test has failed on a site, no user jobs will be sent to it anymore.**

Sitename	Service Type	Service Name	mc	js	swinst	squid	analysis	basic	frontier	jsprod	lcg-cp	get-pfn-from-tfc
T1_DE_FZK	CE	ce-1-fzk.gridka.de	ok	ok	ok	ok	warn	ok	ok	ok		
		ce-2-fzk.gridka.de	ok	ok	ok	ok	warn	ok	ok	ok		
		ce-3-fzk.gridka.de	ok	ok	ok	ok	warn	ok	ok	ok		
		ce-4-fzk.gridka.de	ok	ok	ok	ok	warn	ok	ok	ok		
		ce-5-fzk.gridka.de	ok	ok	ok	ok	warn	ok	ok	ok		
	SRMv2	gridka-dCache.fzk.de									ok	ok

Experiment Specific Monitoring

- Besides the basic grid functionality, experiment specific components and functionalities are tested.
- Access point to results for LHC experiments is the ARDA Dashboard, e.g. for CMS:
<http://arda-dashboard.cern.ch/cms/>



SAM:

- Also tests of dedicated functionalities of the Tier centres requested by individual experiments

Job Summary:

- Type of the job and its exit code.
- History per centre and activity.

PhEDEx data transfers:

- Transfer quality for CMS datasets.
- Collects error messages in case of failures.
- Gives overview on the transferred data samples available at the different sites

- Each computing centre has its own monitoring infrastructure:
 - Nagios
 - Ganglia
 - ...
 - Offers monitoring for hardware, services and network connections.
 - Adapted for the local particularities.
 - Automated notification in case of failures by email, instant messages, SMS, ...
-
- Experiment specific local monitoring:
 - Is the local software installation accessible?
 - Are all required services properly configured?
 - ...

Why Do We Need A Meta Monitoring System

■ Monitoring information is not clearly arranged

- Too many information systems
- Different information displays
- Different or even no support concerning history functionality
- Difficult to find correlations

■ Uncomfortable to use

- Have to manage many browser tabs / windows
- Have to change the settings of the web interface (timerange, ...)
- Have to wait for the download, often more than 30 seconds
- Have to update monitoring websites manually

a meta monitoring system would improve this situation

A customizable software framework with special properties:

■ Only one website

- Display simple warning system, own happiness
- Collect all relevant information (+ their history)
- Show all important external plots (cached for faster access)

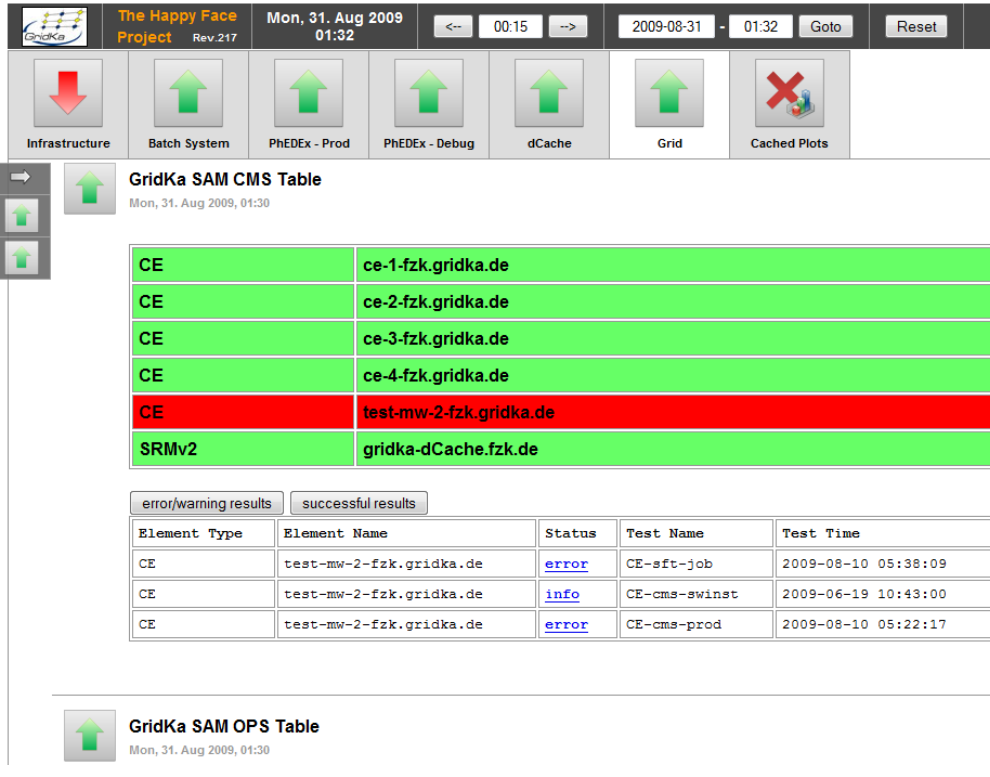
■ Back-end for implementation of own tests

- Design tests, define rules for judging service status and triggering warning alerts
- Correlate data from different sources

■ Fast, simple and up to date

- The complete monitoring information should be renewed every ~ 15 min
- The access / loading time should be very short → simple structure, no complex database
- Every test result should be accessible in less then 3 mouse clicks
- Provides access to the raw information from the data source

The HappyFace Project



The screenshot shows the HappyFace Project web interface. At the top, there's a header with the project name, version (Rev. 217), and a date/time bar (Mon, 31. Aug 2009 01:32). Below this is a navigation bar with icons for Infrastructure, Batch System, PhEDEx - Prod, PhEDEx - Debug, dCache, Grid, and Cached Plots. The main content area displays the 'GridKa SAM CMS Table' for Mon, 31. Aug 2009, 01:30. This table lists various elements (CE, SRMv2) and their corresponding URLs. Below this, there's a section for 'error/warning results' and 'successful results'. The 'error/warning results' table shows three entries for 'test-mw-2-fzk.gridka.de' with status 'error' and 'info'. At the bottom, there's a 'GridKa SAM OPS Table' for Mon, 31. Aug 2009, 01:30.

Element Type	Element Name	Status	Test Name	Test Time
CE	ce-1-fzk.gridka.de			
CE	ce-2-fzk.gridka.de			
CE	ce-3-fzk.gridka.de			
CE	ce-4-fzk.gridka.de			
CE	test-mw-2-fzk.gridka.de	error	CE-sft-job	2009-08-10 05:38:09
CE	test-mw-2-fzk.gridka.de	info	CE-cms-swinst	2009-06-19 10:43:00
CE	test-mw-2-fzk.gridka.de	error	CE-cms-prod	2009-08-10 05:22:17

category navigation & time bar

- each category is a collection of multiple modules
- the status of a category corresponds to the worst error code of its modules

each module has its own configuration and provides a HTML fragment for the final output

the current modules process data from:

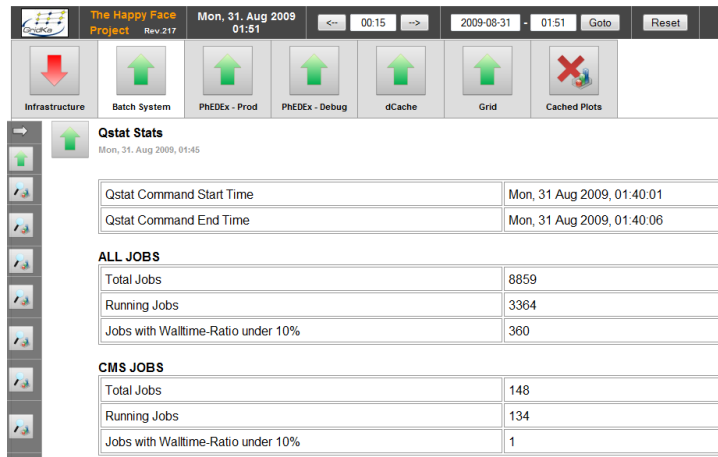
- ARDA Dashbaord
- PhEDEx system (CMS)
- SAM Visualization
- PBS qstat output (FZK)
- dCache
- site specific monitoring systems
- ...

- use available monitoring sources and provide a monitoring information summary for one grid site and a defined set of its services
- provide a simple rating system
- store all external plots in a cache for a very fast access
- every information should be available in less than ~ 3 mouse clicks

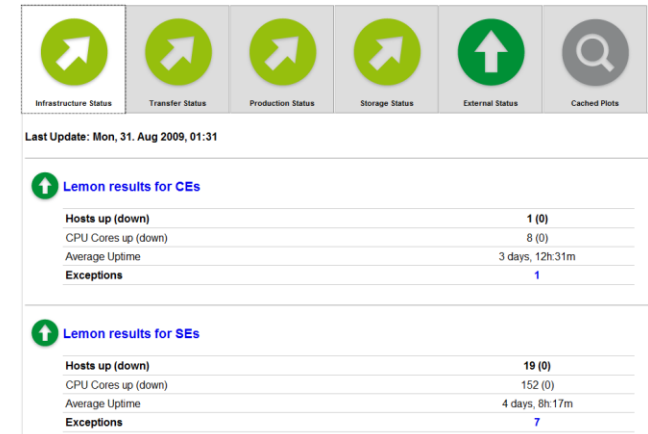
no new monitoring

just a summary of existing ones

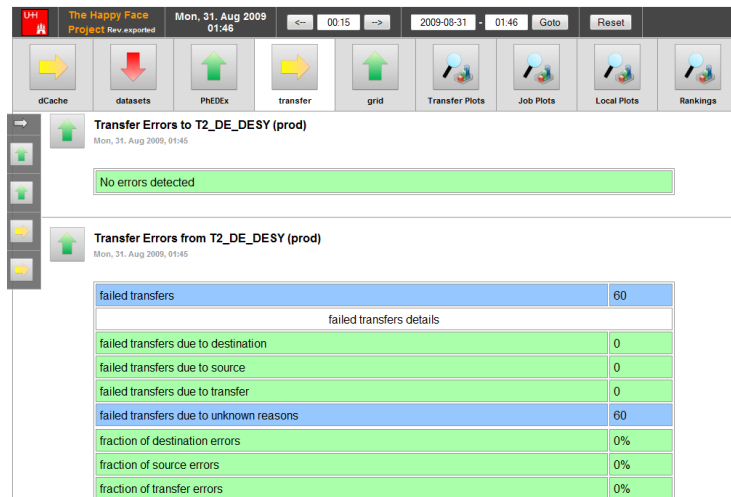
German Sites Using HappyFace



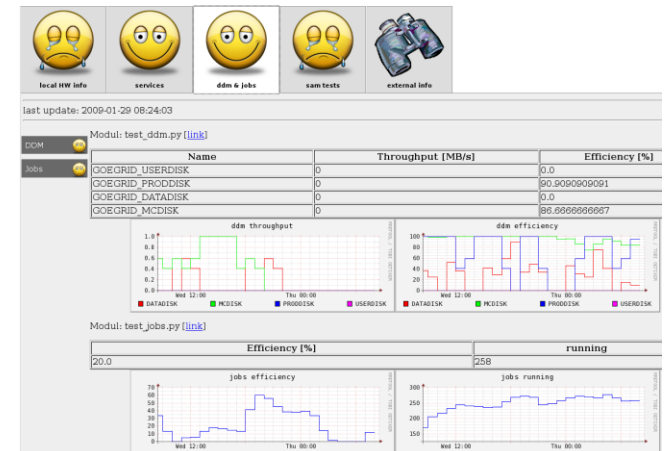
Karlsruhe



Aachen, Philip Sauerland



Hamburg / Desy, Friederike Nowak



Göttingen, Stefan Birkholz

■ Back-end

- **Complete framework in Python:**
 - Good support for object oriented programming
 - Good readable code
- **Good error handling and stability**
 - execution of the modules at the same time in a multi-threaded environment
 - implementation of timeouts for the module execution (e.g. cURL / WGET commands)
 - failed XML parsing or unavailable / inconsistent sources do not crash the framework
- **Decoupling of collecting information and visualization with a SQLite DataBase**
 - timestamps, results, local file locations are stored
 - module specific HTML code is produced dynamically

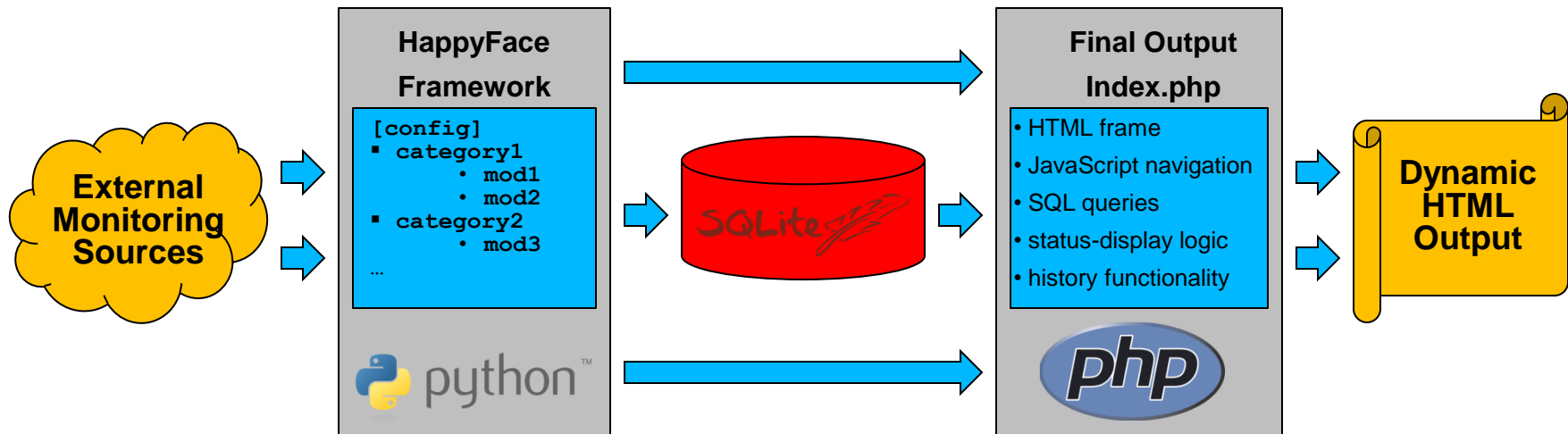


■ Front-end

- Keep same output format: multiple categories, divided in multiple modules
- Keep rating system: **-1** (no info / error), float status value **0.0..1.0** (critical..fine)
- supports style-sheets for the different modules and categories to guarantee equal look & feel
- **Ouput is a PHP/HTML file including SQL queries**
 - “give me the status for a specific time/data”
 - “give me a summary for a specific time scale”



Workflow



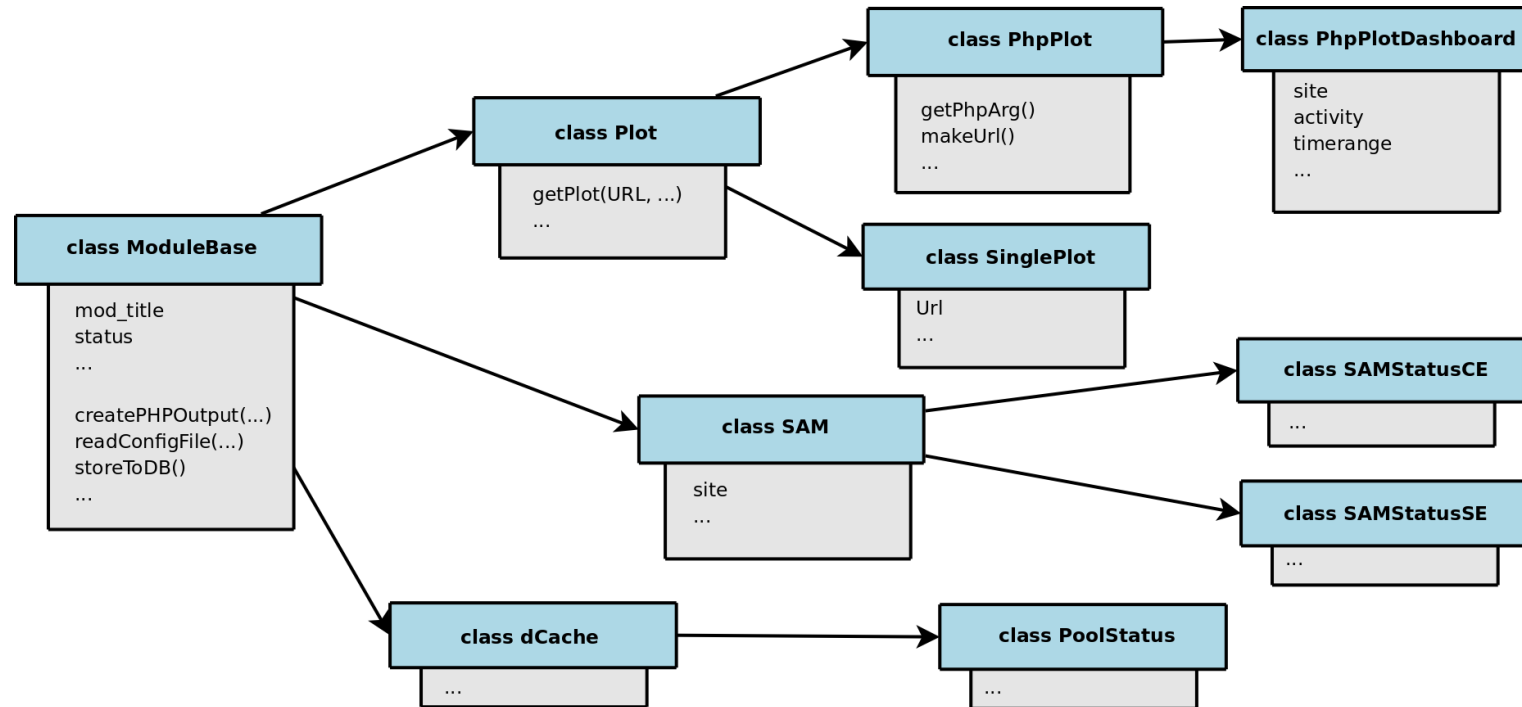
the framework config file sets:

- categories
- the active modules
- the order of the modules in the final output

each module:

- has its own configuration file
- processes external monitoring data
- defines information which are stored to the DataBase
- provides a PHP/HTML code fragment for the final output

Object-oriented structure



- object-oriented structure prevents same code fragments
- ease maintenance and error search
- idea of one repository with all available classes for all sites:
=> a kind of plug-in system: check-out, activate and run

■ conclusion

- HappyFace collects and displays existing monitoring information
- Provides history information even for monitoring sources which do not provide it on their own
- Data is neatly arranged and therefore easily accessible
- Fast access through caching
- Successfully used during large scale computing tests, e.g. CSA08 or STEP09
- Tool already used by many German Sites (Atlas & CMS)

■ Outlook

- Modular structure allows to easily share tests between different sites
- Module development is distributed between all active maintainers
- Comprehensive and fast overview of multiple grid centres through combining their HappyFace instances

GridKa HappyFace: <http://www-ekp.physik.uni-karlsruhe.de/~happyface/gridka>
SVN-Repository: <https://ekptrac.physik.uni-karlsruhe.de/trac/HappyFace>

Kontakt:
Volker Büge (volker.buege@cern.ch)
Viktor Mauch (mauch@ekp.uni-karlsruhe.de)

