

Metrics Correlation and Analysis Service (MCAS)

Project Detailed Requirements Document

Authors:

MCAS Team

Table of Contents

Change Log.....	3
1 System definition.....	3
2 Requirements.....	3
2.1 Data Requirements.....	3
2.2 Data integration.....	4
2.3 Content Hosting.....	4
2.4 Content Aggregation.....	4
2.5 The “Drill down”.....	5
2.6 Data Warehouse.....	6
2.7 Operational Requirements.....	6
2.7.1 Security.....	6
2.7.2 Response time.....	6
2.7.3 Maintainability.....	6
2.7.4 Scalability.....	7
3 Sample Portal Display.....	7

Change Log

Version	Date	Change Description	Prepared By
V1.0	07/21/09	First Version	Andrew Baranovski

1 System definition

The mission of the MCAS (Metric Correlation and Analysis Service) project is to deliver a software solution to help with adaptation, management, and display of the data describing state of applications on the grid. These data are used for monitoring and troubleshooting purposes and are available from either application directly or supplied by the existing data collection and display middle-ware software.

A core business idea of this project is to factor out presentation and business analysis logic from available monitoring solutions into a standalone model, supporting common standards in data manipulation and presentation.

For information on project use cases refer to http://cd-docdb.fnal.gov/cgi-bin/RetrieveFile?docid=3267&version=3&filename=MCAS_requirements_lite.odt

2 Requirements

- a) The product must offer a toolkit and a framework to design and deploy dynamic reports based on data produced or permanently stored by other applications. These reports should address use cases of visual data summarization, content aggregation and data drill down.

The product will contain following applications and tools:

- b) **web frames(widgets)** – widgets will be part of the report composition. Widgets should be designed with the idea that their composition must reflect a report desired by user. The key requirement is that widgets must be re-usable in different reporting application contents. The re usability of the widget implies existence of abstract API specific to that widget or data format that widget should automatically accept for rendering.
- c) **web hosting environment** – application to host, co-display and configure collection of web frames (widgets)
- d) **data integration framework** – user friendly software tool kit to define rules to aggregate,summarize and select data from disparate data sources (including other monitoring systems). These transformations will be adopting data to format and semantics acceptable by the web frame elements.
- e) **data warehouse** – light weight service to temporarily store data used to build dynamic reports.

2.1 Data Requirements

- a) Adopted data format should impose least amount of user level coding for the purpose of storage, retrieval, processing, and adaptation of the new data. For example: an XML data table format can be used.
- b) Define the “data source” and describe structure of data participating in format transformation and display activities.

2.2 Data integration

- a) Support data polling(and configuration of thereof) to sample from configured data source.
- b) Support business rules engine to generate data presentation acceptable to project's web frames. Initially, business rules engine may only work with data stored in the warehouse.
- c) Provide intuitive and easy to use tools to setup data transformation work-flows that produce data required by the web frame elements.
- d) Notify portal administrators of errors occurring during execution of the rules engine.

2.3 Content Hosting

- a) Content of the dynamic report must be available via the Web.
- b) A web page containing the report must be protected by access rights defined roles of a particular user(an entity with proven credentials).

2.4 Content Aggregation

- a) Content of the dynamic report web page must be composed of one or more re-usable frames each of which has been independently initialized with data describing some unique perspective of a particular system aspect.
- b) A user must be able to add or remove frames to/from the dynamic report window.
- c) A user must be able to orchestrate rendering of collection of frames using common set of parameters. In addition, the page itself (it's entire content) must be able to respond to parameters set in the URL of the request to that page.
- d) A user must be able to edit common set of parameters in a separate frame or special portal admin window.
- e) A user must be able to open and edit preferences of the web frame.

Proposed list of common frame types:

- f) Generic Table view frame

The purpose of the table view display is to summarize value-pair set array data using features of the spreadsheet. In its basic form, Table view should be parametrized by

1. maximum number of rows to display.
 2. sorting column
- g) Generic graph rendering/image display frame
1. User must be able to setup Graph display frame. Graph display will reference list of image locations to display image data in the table like form. The number of columns in that table should be configurable.
 2. Be able to embed links in the plots.(see drill downs) For example, when clicking on a ganglia plot, we want to bring up the whole ganglia page (with the other links). Another way of seeing this is having groups of plots: when I click on one plot of a group, I want to maximize all the group.
 3. Be able to easily rearrange the layout of the plots. Ideally, we'd like to say where the plot goes in the grid. Most important (supported) feature is add / delete plots to a page.
- h) Generic time series display frame

We'd like to be able to render value of a function applied over sequence of documents. The resulting time series plot should display how the content of the document(or data element) is changing over time. This frame serves need of application users who wish to define their own views on metrics mined from the warehouse service. The warehouse service may be internal to the project, or it can be external deployment suitably adapted to conform to project's data format.

- i) Health frame display displaying a sequence of health indicators.

These indicators are meant to visualize health, viability, performance or other relative to a threshold characteristic. The indicators may color codes to distinguish between the states/characteristics. However, the design of the indicator should also be color blind friendly. Health indicators should be click-able if detailed status about that indicator is available in HTML. Indicators may be sorted in ascending order using each indicator's weight.

2.5 The “Drill down”.

- a) Each frame element must export list of click-able tags, which provide enough context to initialize a template of the standalone window, designed to reflect more detailed information about clicked tag.
- b) Each frame must therefore also support a template-like URL string which will be used in conjunctions with values of tags clicked by the user. The purpose of using template URLs is to decouple drill down design from implementation of the frame. That is, the design of the frame should allow insertion of that frame into potentially many independent drill down hierarchies.

- c) The drill down page itself should be addressable using a URL sensible to parameters specific to tags clicked on the parent page. This URL may be managed by one of the portal windows or it can be hosted by other applications.

2.6 Data Warehouse.

- a) Using common data format, establish support for acceptance and transient storage of data which can be used to generate content of individual web frames.
- b) Provide mechanism for querying the warehouse.
- c) Support limits on size , quantity and time of validity for the data which can be stored by the warehouse.

2.7 Operational Requirements

2.7.1 Security

- a) A web page must offer sign-in and recognize sequence of requests as belonging to a single user. We do not require support for GSI authentication at this time.
- b) All services must be deployed in a “secure” web container environment. The web container environment should prohibit remote administration interfaces which can potentially enable upload or modification of project's codes. This requirement relates to both data transformation and frame management systems.
- c) The data sensitive to domain should not be available outside of that domain. Note to implementer: this can be accomplished by setting up HTTP traversal rules specific to address of the portal and/or password protecting the portal web page.

2.7.2 Response time

- a) We expect the system to render a “reasonably” complicated portal web page within 1 or 2 seconds. Consecutively , no data source should require more then 2 seconds to respond with valid result or provide standard error message indicating the timeout or other error condition.
- b) The MCAS deployment will be operating as service shared between multiple users/experiments. The MCAS deployment must also able to manage several independent contexts at a time. The system should be comfortable supporting several portal pages that may be owned by different user groups.
- c) The MCAS service is not required to be a critical system.

2.7.3 *Maintainability*

Routine manual maintenance should be limited to:

- a) selective backups of data (if requested by data users)
- b) software and hardware upgrades

System should automatically

- c) Prune data that is does not fit service agreement with data warehouse
- d) Rotate log files
- e) Produce and send alarms in response to internal to MCAS system error conditions.

In addition, the system should provide:

- f) Software documentation as per - <http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2684>
- g) Process of testing and deployment of the software should comply with : <http://cd-docdb.fnal.gov/cgi-bin/ShowDocument?docid=2684>

2.7.4 *Scalability*

- a) All objects used in communicating between components of the infrastructure must carry sufficient information to be independently processable. That is, the MCAS system should not rely on shared context to execute or render *independent* user requests.

3 Sample Portal Display

The screenshot shows a web portal with a grid of monitoring charts. On the left, there are vertical bar graphs for 'total', 'flushPools', 'LFS...', 'Resili...', 'stage...', 'T1DiskEvalPools', 'ReadOnlyPools', and 'LPCPools'. A zoomed-in chart titled 'Sum Restricted DCAP sessions' shows a bar graph with a y-axis labeled 'Count' and an x-axis with time markers. Below the charts is a table titled 'CellInfoTableView' with columns for 'cell', 'domain', 'ping', 'time', 'version', and 'offline'. The table contains two rows of data.

cell	domain	ping	time	version	offline
w-cmsstor74-3	w-cmsstor74-3Domain	9/28 34581	0506:00:22:08	1-9-2-4(10734)	false
w-cmsstor75-1	w-cmsstor75-1Domain				true

Annotations in blue callouts provide the following descriptions:

- Generic Graphs/Image View frame
- Zoomed image from external data sources
- Click on the Image to Zoom it
- Color coded bar graphs to display the health of different components in the system
- Generic Table View frame
- Configurable, 3 Column display makes it easy to visually correlate information from different sources