TOOLS AND PROCEDURES FOR HIGH QUALITY TECHNICAL INFRASTRUCTURE MONITORING REFERENCE DATA AT CERN

R. Martini, M. Braeger, J. Salmon, A. Suwalska CERN, Geneva, Switzerland

Abstract

The monitoring of the technical infrastructure at CERN relies on the quality of the definition of numerous and heterogeneous data sources. In 2006, we introduced the MoDESTI procedure for the Technical Infrastructure Monitoring (TIM) system to promote data quality. The first step in the data integration process is the standardisation of the declaration of the various data points whether these are alarms, equipment statuses or analogue measurement values. Users declare their data points and can follow their requests, monitoring personnel ensure the infrastructure is adapted to the new data, and control room operators check that the data points are defined in a consistent and intelligible way. Furthermore, rigorous validations are carried out on input data to ensure correctness as well as optimal integration with other computer systems at CERN (maintenance management, geographical viewing tools etc.). We are now redesigning the MoDESTI procedure in order to provide an intuitive and streamlined Web based tool for managing data definition, as well as reducing the time between data point integration requests and implementation. Additionally, we are introducing a Class-Device-Property data definition model, a standard in the CERN accelerator sector, for a more flexible use of the TIM data points.

INTRODUCTION

Computerised monitoring of the technical infrastructure at CERN goes back to the early 1980s when a DEC DPD11 was used to display around 2000 alarms onto 2 screens in a dedicated control room. When this system was migrated to a distributed solution based on Unix PCs the decision was taken to hold monitoring definition data on an external database from which the monitoring system would be configured.

This strategy was of great benefit as the monitoring system evolved first to the Technical Data Server – (TDS) in the mid 1990s and finally to the Technical Infrastructure Monitoring (TIM) system [1]. Throughout this time the number of data points grew to reach around 100K data points today, and there is a constant demand to incorporate new monitoring from different services installing new equipment and upgrading existing installations.

For all these monitoring systems the issue of ensuring that data were correctly defined was paramount in determining satisfactory performance. The specification of the monitoring data is the responsibility of the different services whose equipment is being monitored, but the data must be validated by the operators who monitor the alarm screens as well as those responsible for the monitoring system itself.

To ensure that the declared data are correct, complete and follow established standards, the Monitoring Data Entry System for Technical Instructure (MoDESTI) procedure [2] was devised and deployed shortly after the implementation of TIM in 2005.

MoDESTI has evolved over the years to handle the many different scenarios relating to the maintenance of the data that defines TIM; however it is cumbersome to use and difficult to maintain when changes in the monitoring system need to be covered. For this reason a complete reworking of the tool was proposed in 2014 and is now in development.

MODESTI PRINCIPLES

Templates Monitoring data includes descriptive information such as location, equipment concerned, person responsible, information in dealing with an alarm (priority, causes, consequences, actions to take), applications that use the data point (synoptic views, logging, external systems), and configuration data that indicates which acquisition units handle the data points as well as how they are communicated to and from TIM. In this way the reference database contains an overview of all that is monitored by TIM and can be consulted by any authorised user.

Changes will occur during the lifetime of a data point, either through corrections to inaccuracies or due to changes in the way the data point is monitored. Furthermore data points are not monitored indefinitely, when an installation is dismantled, the related data points must be removed from the monitoring system to avoid clutter and possible confusion. These events are also managed by MoDESTI so that the monitoring systems always keep up with the latest operational changes.

Since the aim of MoDESTI was to promote data quality in monitoring, the tool was adapted for the launch of the CERN Safety Alarm Monitoring (CSAM) system. This is a separate system from TIM but CSAM is configured with data defined through the MoDESTI procedure held on the reference database. More recently, the capacity of MoDESTI in enforcing standard definitions and rules relating to alarms was harnessed by CERN's WinCC application for alarms sent to the CCC.

In order to declare data points for monitoring by TIM users must create a data integration request and submit their data in a standard way. Depending on the nature of the data points being declared, specific actions must be taken by the different agents involved in the monitoring process. Alarms must be validated by CCC operators to verify that they are correct, complete and comprehensible

ISBN 978-3-95450-148-9

when they arrive on screen. Some data points may need cabling to PLCs, those who carry out this work will be notified by the MoDESTI procedure when this is required and will be able to inform the system when this has been carried out. Those responsible for the CSAM system are notified when changes to their alarms are implemented so that they can synchronise their data. Finally, data points are configured onto TIM by the system administrators.

The current implementation of MoDESTI uses CERN's Engineering Data Management System (EDMS) to create data integration requests and then manage the workflow. Data is entered on pre-formatted Excel data sheets where simple checks are carried out by Visual Basic modules before these files are linked to the EDMS request. Data is then loaded onto the reference database where fuller validation checks can be carried out prior to configuration in TIM.

ENTEPRISE LOGIC

Looking back at the last ten years of experience with the TIM monitoring service we can confirm that investing in a reference – configuration database was the correct approach. The complete review of the data model, data validations, integration and configuration processes in 2005 has guaranteed a stable and reliable configuration data source for the clients of our main systems TIM and CSAM.

Data Validation Principles

The reference data checks start with the light validations as mentioned above. More complex and thorough business logic validations with detailed cross checking of the declared data against existing monitoring data are then carried out. Great care is taken to verify that the integrated data do not lead to non-unique point declarations, incomplete, non-standard, or ambiguous definitions. Alarm points must pass additional validations to ensure that once operational the CCC team have access to all details to react efficiently once the alarm is activated.

TIM is tightly coupled with other CERN technical databases: INFOR EAM for equipment maintenance, GEOSIP for locations, LASER for alarm definitions, TIMBER for long term logging, and FOUNDATION for personnel. Verification procedures must ensure that any TIM data referencing the external systems or any TIM processes injecting data in these systems (e.g.: data logging) are validated against the parent source. This is achieved by implementing off-line synchronizations of the lookup data, thus guaranteeing that our operational environment remains independent at all times from the operational constraints of the third-party databases. The same principle is applied to systems which declare data in TIM. The PL/SQL modules that carry out the validations for MoDESTI are also called by external systems in order to protect the TIM configuration data against any unverified modifications. The MoDESTI interface and the workflow are decoupled from the data business logic and share only the data validation APIs. These APIs are exposed as externally accessible PL/SQL procedures. They call the same validations that are used by the standard MoDESTI process thereby guaranteeing that the data injected to TIM are of the same quality as those that follow the standard declaration path.

Having a well-structured and organized database is essential in providing the correct configuration for the monitoring systems. Another, equally important role of the reference database is to serve TIM users with complementary information related to their work. For this reason we provide applications such as Help Alarm and SMILE (Static Monitoring Information Lookup Engine). Since alarm and data tag signals are transmitted with minimal operational information, these web based applications allow an efficient interpretation of alarms as well a full understanding data tag meanings. Authorised CCC operators can also register further operational instructions or update a tag's detailed information contributing to the overall "operational" quality of the TIM data.

Class/Device/Property

The class/device/property data paradigm is very popular in the CERN control environment and in particular in the accelerator sector. For this reason, in 2014, the TIM data model was adapted to handle the class-device-property representation for TIM data [3]. This new data abstraction layer extends the definition of data tags allowing a mapping to the properties defined within devices. Organizing data in the class-device-property model enhances the further usability of the data sets which share common definitions and are logically classified within devices. This allows us to focus on the particular type of device being monitored, and "hides" the underlying tags to a large extent. On the client side, both when creating visualisation symbols or monitoring views, and writing client applications, the user can work exclusively on the device level without needing to know about the underlying tags used interpret the device. Having opened this new data dimension in TIM we foresee a rapidly growing interest in TIM in the coming years [4].

TOOLS AND TECHNOLOGIES

The current implementation of MoDESTI uses CERN's Engineering Data Management System (EDMS) to create data integration requests and then manage the workflow, and pre-formatted Excel data sheets to capture the data

ISBN 978-3-95450-148-9

which is then loaded onto the reference database prior to configuration in TIM.

Though this approach was quick to implement, it has several serious drawbacks. First, the batch processing nature of the database validations leads to a slow data integration process as users must wait until the cycle is triggered to receive the validation result, and must wait until the next cycle to reattempt their validation. Second, managing different versions of Excel work sheets when changes to TIM or CSAM require the specification of new data point parameters is also time consuming and requires cooperation with users to migrate to the new Excel template. Third, the workflow cannot be easily altered to adapt to new requirements, moreover the workflow cannot be made dependant on the data of a request. Lastly, the current implementation could not take advantage of the improvements in TIM which allowed for the dynamic reconfiguration of its data acquisition modules.

MoDESTI 2

Having reviewed the drawbacks of the existing collection of tools, a new design to realise the MoDESTI principles and requirements was proposed in March 2015. MoDESTI 2 is a web-based collaborative configuration management tool which incorporates the entire workflow of the data integration process from initial data definition through to data retirement in a single web application.

Plugin System

MoDESTI 2 incorporates extensibility as a first-class principle from the beginning. It has a simple yet flexible plugin-based architecture, which allows a new monitoring domain to be easily integrated with the system. Each plugin provides two major components: a mapping like representation of its domain and constraints called a schema, and a specification of the workflow requirements of the domain called a workflow process. The following sections describe these two components.

The MoDESTI Schema

Each domain provides its own schema, which defines how data points of that particular domain must look. The schema specifies the properties that the points of a domain can have. A schema also provides flexible ways of defining constraints and relationships between properties of data points. Schemas are written in the simple, flexible and lightweight JSON format.

The MoDESTI Workflow Process

Workflow processes in MoDESTI are implemented as BPMN (Business Process Model and Notation) files [5]. This is an XML file that specifies the stages the workflow is composed of and the conditions that govern the transitions between those states. The file also includes a graphical representation of the workflow for ease of understanding by non-experts. The Activiti workflow engine [6] is used to provide reliable, robust, transactional workflows for MoDESTI requests. The default built-in workflow begins with initial data input and flows through validation, approval, configuration and testing stages. However, the default workflow is not prescriptive and individual plugins may supply their own custom workflows as required by the domain in question.

Architecture

The architecture of the new MoDESTI system consists of two distinct components: the backend and the frontend. The backend is a standalone application server which exposes a REST API for creating, updating, reading and deleting “requests”, and persists them to a database. The frontend is a web-based user interface which allows a user to graphically interact with the backend to craft a MoDESTI request. The following diagram illustrates the high-level structure of the system.

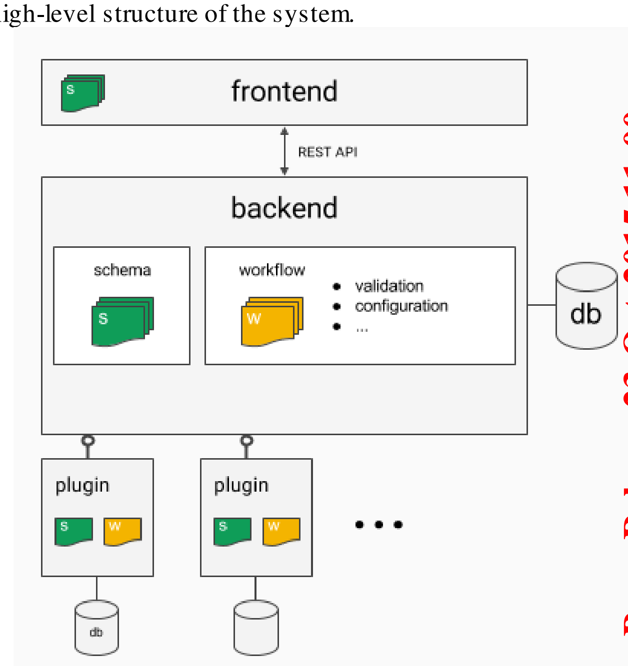

Figure 1: MoDESTI 2 architecture.

The diagram illustrates the plugin-based nature of the system, whereby each plugin provides a schema and a workflow. The backend collects these schemas and workflows, and provides them to the frontend. the frontend shows dynamically generated domain-specific data manipulation controls based on the schema in use.

This design was chosen for a number of reasons, including separation of concerns, maintainability, loose coupling, testability, and the differing skill sets required by frontend and backend developers. Consequently, the frontend contains little to no business logic; it simply provides graphical manipulation capabilities.

Backend

The backend is a Java application which makes use of several libraries for ease of implementation. Most notably, the Spring framework is used for core dependency injection and bootstrapping functionality, as

ISBN 978-3-95450-148-9

well as several useful abstractions for implementing database persistence functionality.

Frontend

The frontend is written in JavaScript and makes extensive use of the AngularJS framework for structural organisation and coherent functional separation. AngularJS lends itself well to the REST API paradigm, and contains built-in mechanisms for asynchronous backend communication via JSON.

To ease the migration process for existing MoDESTI users, the need for an Excel-like spreadsheet-based interface component for defining a set of data points was chosen, providing a familiar user experience. The "handsontable" library was chosen as the base for this component. The spreadsheet is dynamically generated based on the domain-specific schema.

Advantages over Existing System

MoDESTI 2 is a significant improvement on the existing system for a number of reasons. First, it provides a centralised platform with which users can define new data and search for existing data, without having to switch between several different programs and web pages, and without having to manually manage different versions of files stored locally on the user's computer. The entire user experience is integrated into a single web application; the interface provides instantaneous validation feedback to the user, vastly reducing the time required to identify data entry errors from potentially days to a matter of seconds, and the target systems (e.g. TIM or CSAM) can be configured instantly with new data.

The schema-based approach to modelling domains and their constraints is a powerful one that enables flexibility to changes. Changes to the parameters of a domain can be quickly and easily incorporated by modifying the schema.

Last, as target domains in MoDESTI 2 are integrated as plugins, the core system contains no domain-specific logic or functionality. Each target system specifies the structure of its data points (via schemas), its workflow, and its domain-specific validation and configuration logic. This means that new monitoring systems (or other systems that may benefit from the MoDESTI data integration/configuration management principles) can be quickly and easily integrated into a MoDESTI plugin.

Summary

The frontend world of JavaScript frameworks is fast-changing in contrast to a more slowly evolving Java-based backend world. Having a solid, stable backend positions us well for the future in the eventuality that the frontend might need adapting to a new framework. Separating the two is a good general approach, providing flexibility to change in the long term.

CONCLUSION

The use of a reference database for holding monitoring parameters has been a crucial factor in ensuring a smooth transition between major system upgrades. It facilitates the validation of newly declared data points as well as changes that are often made to monitoring parameters, and lastly provides a view to non specialists of what is monitored by the system. Users can have confidence in TIM insofar as whatever is defined in the system abides by established rules and standards, and has been validated by the agents involved in the monitoring process.

The implementation of the new interface is an important step in making the system more responsive and easier to use, as well as freeing support staff from tedious maintenance tasks. The extensible, plugin-based nature of the new system enables new monitoring systems to be integrated with the well-defined principles that MoDESTI enforces to ensure quality and validity of monitoring data.

REFERENCES

[1] M. Bräger, M. Brightwell, A. Lang, A. Suwalska, "A customizable platform for high-availability monitoring, control and data distribution at CERN", ICALEPCS'11, 2011

[2] MoDESTI home page https://edms.cern.ch/document/707081/1

[3] Devices: a new data structure in C2MON by Mark Brightwell (2013) https://edms.cern.ch/document/1308340/1

[4] Integration, processing, analysis methodologies and tools for ensuring high data quality and rapid data access in the TIM monitoring system (ICALLEPCS 2013) https://edms.cern.ch/document/1311918/2

[5] BPMN 2.0 Specification, Object Management Group (2011) http://www.omg.org/spec/BPMN/2.0/

[6] Activiti BPM Platform http://activiti.org/

ISBN 978-3-95450-148-9