ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA036

APPLYING SERVICE-ORIENTED ARCHITECTURE TO ARCHIVING DATA IN CONTROL AND MONITORING SYSTEMS*

J. M. Nogiec#, K. Trombly-Freytag, FNAL, Batavia, IL 60510, USA

Abstract

Current trends in the architecture of software systems focus our attention on building systems using a set of loosely coupled components, each providing a specific functionality known as service. It is not much different in control and monitoring systems, where a functionally distinct sub-system can be identified and independently designed, implemented, deployed and maintained. One functionality that renders itself perfectly to becoming a service is archiving the history of the system state. The design of such a service and our experience of using it are the topic of this article. The service is built with responsibility segregation in mind, therefore, it provides for reducing data processing on the data viewer side and separation of data access and modification operations. The service architecture and the details concerning its data store design are discussed. An implementation of a service client capable of archiving EPICS process variables (PV) and LabVIEW shared variables is presented. Data access tools, including a browser-based data viewer and a mobile viewer, are also presented.

INTRODUCTION

As part of building Fermilab's Solenoid Test Facility (STF) [1][2], a unique magnet test facility designed for testing large aperture superconducting solenoid magnets, the authors designed and implemented a dedicated data archival service based on a microservice.

Monolithic software architectures have proven to be very difficult to develop and maintain efficiently. Service Oriented Architecture (SOA) allows for a modular approach to application design based on functional boundaries. Microservices offer a solution inside of SOA that focuses on the independence of the service as an entity, rather than as part of an application. Consequently, a service implemented as a microservice is developed and deployed independently from the applications that use it.

This design approach has been applied to the data management solution for the STF test facility. The developed management solution belongs to a class of systems known as a data historians.

Data historians are data management systems dedicated to recording and retrieving data organized by time, known as time-series data. Time-series data allow the user to view the data as a trend plot or as tabular data over a time range. Data historian applications are found in a large set of domains, including industrial process control and monitoring computing resources in data centres. Collected data can be analysed for performance monitoring, optimization, material consumption, plant availability, quality control, etc.

In control and monitoring systems, data historians (sometimes referred to as process historians) are used in systems that support management of the plant. Data from various sources, such as a PLC, embedded computers, instruments, and intelligent sensors, are collected in a set of uniquely named tags and, thus, provide a complete view of the controlled plant.

DATA ARCHIVING IN A TEST FACILITY

A typical process historian can be characterized by:

- High capacity storage and use of data compression.
- High insertion rates (thousands of points per second).
- Noise filtering and use of data interpolation for reducing data size.
- Interoperability with many heterogeneous data sources.
- Included analytic capabilities aiding in answering plant management relevant questions such as availability and utilization of plant equipment, production rates, and max/min values of gathered signals.
- Presentation of gathered data in the form of trends or calculated metrics.

Although data historians used in an R&D setting still need to save time-series data of many points, this niche application domain presents itself with a different and specific set of quality of service demands.

The data archival system for the STF R&D test facility can be characterized by:

- Relatively limited number of tags (measured in hundreds) and rather low insertion rate.
- Data organized as tests (uniquely identifiable time-boxed operations of the system).
- Storing all signal values, rather than only those values which have changed.
- Testing of different subjects (accelerator components such as superconducting magnets) leading to different sets of data to be archived for different tests.

The off-the-shelf data historians offer an abundance of features, but only a subset of them can be utilized in the R&D testing domain. They excel at handling time-series data, but are not designed to handle relational data. The limited ability to impose a structure on top of the time-series data, such as including metadata and relating the

* Work supported by the U.S. Department of Energy under contract no. DE-AC02-07CH11359
#nogiec@fnal.gov

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA036

time-series data to other events and data collected by other systems, is a weak point of standardized solutions.

In a test-oriented R&D facility with several independent systems gathering data and controlling subsystems, flexibility, extensibility and introspection play vital roles. It is also very important to keep the data in context, which means related to other data and information.

The difference between a typical control system and a R&D test system is that an industrial plant saves a continuous stream of data reflecting the state of the system, whereas a test facility saves data in different sets of tags (per subject) and their corresponding discrete sets of data in the form of runs. A run is a set of data taken over a period of time that fulfils some part of a test plan. In a typical industrial plant, a set of tags does not change frequently and gaps in data correspond to system or sub-system outages. In contrast, the set of tags for each subject tested in an R&D facility will be different, and gaps in data indicate time between runs. (see Fig. 1).

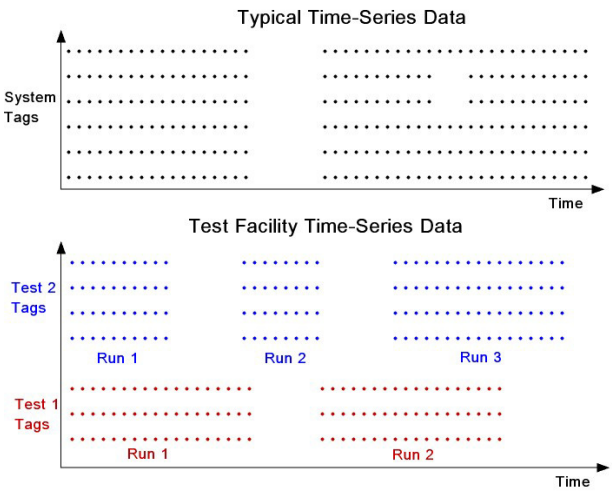

Figure 1: Time-Series data: typical vs. test facility.

ARCHITECTURE

The architecture of the data archival system follows the SOA approach and is designed as a microservice. Microservices are used to cut monolithic applications vertically, thus encapsulating an entire function in a maintainable, functionally discrete package with a specified API. The use of a microservice in archiving in our specific realm of monitoring and control data allows for independence and ensures separation of concerns. As such it can be used both for process data archiving and generic data archiving.

The overall architectural design comprises several layers (see Fig. 2):

- the presentation layer, with data retrieval, and data visualization as trends,
- the data storage layer,
- the data archival layer, and
- the data acquisition and direct control layer.

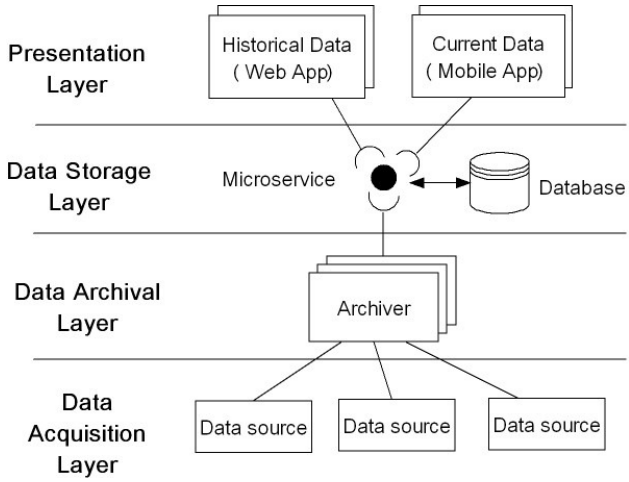

Figure 2: Data archiving architecture.

DATA STORAGE LAYER

The data storage layer comprises a microservice with an XML-based API and a storage mechanism based on a relational database management system (RDBMS).

Microservice

A microservice is a separate, independent component offering a specific service, in our case a data archival service. All external dependencies are on the service API, which constitutes a de facto contract on the service functionality. As long as the contract is upheld, the service maintainers are free to modify its implementation, including its design, technology and tools (e.g. programming language, application server, software development toolkit).

The discussed data storage service for time-series data fulfils the typical characteristics of the microservice, including:

- Built as an independently replaceable and upgradeable component, thus suitable for the evolutionary design approach.
- Organized around a business capability – the archival service spans user-interface, persistent storage, and external integration, thereby forming a complete solution providing a distinctly different functionality.
- Focused on being a product, a separately maintainable and upgradable system.
- Decoupled and cohesive with decentralized data management.

The microservice is built with responsibility segregation in mind: essential functionality is supplemented with calls that provide data in a format easily consumable by data visualization clients. This solution reduces the code complexity on the client side dealing with data aggregation and transformation for visualization.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA036

Data Persistence Solutions

There exist several approaches to data management when dealing with time-series data. These include:

- Specialized solution dedicated to storing time-series data.
- Solutions based on general-purpose RDBMS.
- Solutions based on general NOSQL systems.
- Hybrid solutions: RDBMS for aggregate and metadata, combined with dedicated storage for time-series data.

The simplicity and high availability of popular RDBMS, coupled with their sufficient performance, flexibility and simplicity of using an existing, dependable storage engine, led the authors to construct their data storage solution based on a relational database. The focus of this solution is not on the insertion rates, but rather on the simplicity and flexibility when searching and querying interrelated data.

The RDBMS system used is the open source MariaDB.

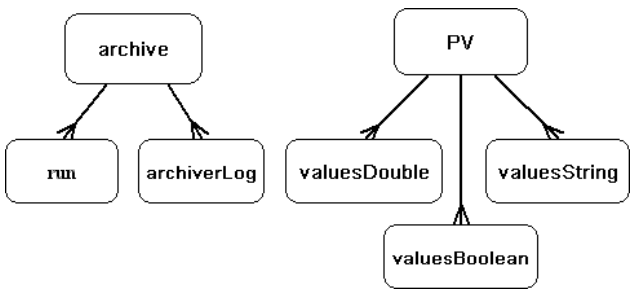

Figure 3: Archive data model.

Data Model

The data is stored in archives. Each archive contains a set of named time-series data (PVs), where names are unique tags with a *system:tag* format (see Fig. 3).

Each archive uses a separate set of tables to store data of different types, with each set containing one or more runs. A run groups the data in a specified time range, corresponding to an actual test or a separate activity at the facility (see Fig. 3).

DATA ARCHIVAL LAYER

The data archival layer, situated above the data acquisition layer, may contain several archivers concurrently storing data to different archives.

Archiver

The archiver has a modular design consisting of a data connector component that periodically acquires data from the data sources, a processor aggregating the data and preparing it for saving, and multiple data persistence modules. Minimal configuration is needed as the archiver will automatically detect changes in the tag name space and update the archive meta-data.

Currently, the standard persistence module is based on the microservice described above, used for standard recording of the history of the system and targeted at easy inspection and visualization of data. Alternate data saving is provided by two other file-based persistence engines.

The first file-based persistence option is based on the TDMS format allowing for fast recording of self-described data. This solution is geared toward recording high insertion rate data for a relatively short period of time during specialized tests. The data saved in files can be uploaded off-line into the appropriate database archive.

The second file-based solution allows for recording data in the ASCII CSV format for a short period of time, and is geared toward immediate, ad-hoc analysis with standard tools (such as MS Excel) during activities such as test stand commissioning and troubleshooting.

Each archival option can be independently enabled and disabled (see Fig. 4). The data acquisition and storage modules operate with independently settable acquisition and insertion intervals.

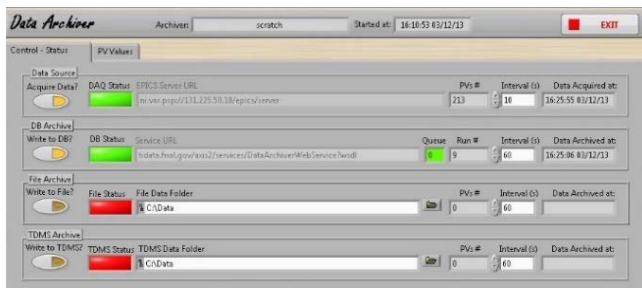

Figure 4: Archiver control panel.

Data Sources

The archiver used in the system supports two data sources: process variables coming via the EPICS channel access interface and LabVIEW shared variables. Using the IFIX gateway [3] developed as part of the complete test facility system solution, the archiver can also gather data from the iFIX SCADA system from GE Digital.

PRESENTATION LAYER

Currently the presentation layer includes two applications for data visualization:

- a Web based application for historical trend analysis, and
- a mobile application for showing real-time trends of selected time-series data.

Web-based Visualization Application

System data visualization is supported by a charting and data extraction Web application (see Fig. 5). The application allows for choosing the data series to be visualized by selecting the appropriate archive and tags of interest within a specified time-horizon. It accesses stored historical data via the microservice.

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA036

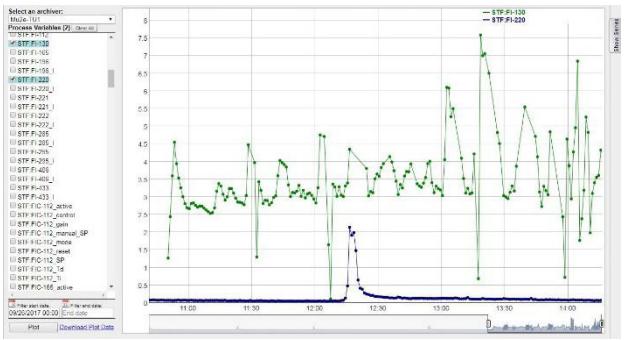

Figure 5: Web-based charting application.

The visualized time-series can be selectively switched on/off. A waveform showing the silhouette of the signal trend at the bottom of the screen can be used to select a sub-range of the analysed trends. The selected time-series can be downloaded for off-line analysis from the same tool.

Mobile Plotter Application

The mobile trend visualizer (see Fig. 6) works on Android-based devices (tablets, phones) and is built to allow spot checking the system's current behaviour and the status of crucial system elements, regardless of the current location of the user. It allows for selecting an hour, day or week time-range, nicely complementing the rather stationary long-term historical analysis tool.

Figure 6: Mobile trend plotting app.

SUMMARY

The described data management solution is part of the system developed for Fermilab's Solenoid Test Facility. This solution has already been used to test a coupling coil for the Muon Ionization Cooling Experiment (MICE) [4], and its future use includes testing solenoid magnets for the Fermilab Mu2e experiment [5].

The system follows the tenants of the Service Oriented Architecture - decoupling, isolation, composition, integration, discrete and autonomous service - with an archival microservice at its core. The main function of the service is to persist time-series data and allow for retrieval of historical time-series data.

As a true microservice it is maintainable and configurable separately from the rest of the SolTF systems and is developed in a software technology distinctly different from other components of the system.

The organization of the data repository reflects our R&D testing environment; it requires a specific organization of the data that is dictated by our testing procedures and needs.

The microservice and its three companion applications - the archiver, the Web-based data browser and display, and the mobile data plotter - form a complete solution for data archival and retrieval.

The system is currently fully functional, however, there are ways it could be extended, for example, by adding an alternative time-series storage engine, or adding yet another interactive data visualization tool.

REFERENCES

[1] J. Nogiec, R. Carcagno, S. Kotelnikov, K. Trombly-Freytag, "Software System for Monitoring and Control at the Solenoid Test facility," *ICALEPCS 2013*, San Francisco, USA, October 2013, paper THPPC065.

[2] R. Rabehl, R. Carcagno, J. Nogiec, D. Orris, W. Soyars, and C. Sylvester, "A Cryogenic Test Stand for Large Superconducting Solenoid Magnets," *CEC/ICMC 2013*, Anchorage, USA, June 2013.

[3] J. Nogiec *et al.*, "A Gateway to Intellution's FIX System", in *Proc. PCaPAC'02*, Frascati, Italy, October 2002.

[4] R. Carcagno *et al.*, "Prototype Conduction Cooled Capture Solenoid Test Design and Plans," *MT-23*, Boston, USA, July 2013, paper TU5PFP056.

[5] M. Lopes *et al.*, "Mu2e Transport Solenoid Prototype Tests Results," *MT-24*, Seoul, Korea, October 2015, paper IPoBC_01.