ITERATIVE DEVELOPMENT OF THE GENERIC CONTINUOUS SCANS IN SARDANA

Z. Reszela, G. Cuni, C. M. Falcón Torres, D. Fernandez-Carreiras, C. Pascual-Izarra, M. Rosanes Siscart, ALBA-CELLS Synchrotron, Cerdanyola del Vallès, Spain

Abstract

Sardana [1] is a software suite for Supervision, Control and Data Acquisition in scientific installations. It aims to reduce cost and time of design, development and support of the control and data acquisition systems. Sardana is used in several synchrotrons where continuous scans are the desired way of executing experiments [2]. Most experiments require an extensive and coordinated control of many aspects like positioning, data acquisition, synchronization and storage. Many successful ad-hoc solutions have already been developed, however they lack generalization and are hard to maintain or reuse. Sardana, thanks to the Taurus [3] based applications, allows the users to configure and control the scan experiments. The MacroServer, a flexible Python [4] based sequencer, provides parametrizable turn-key scan procedures. Thanks to the Device Pool controllers interfaces, heterogeneous hardware can be easily plug into Sardana and their elements used during scans and data acquisitions. Development of the continuous scans is an ongoing iterative process and its current status is described in this paper.

INTRODUCTION

Traditionally, a continuous scan needed an ad-hoc controls software being in charge of the configuration, process control, data acquisition and storage of the whole measurement. Its use was restricted to particular axes, hardware involved to count encoder or motor pulses and generate external triggers or specific experimental channels and detectors. The generalization of this type of scans is strategic in all synchrotrons and potentially in any other laboratory. A generic continuous scan framework provides the flexibility of step scans in a continuous and synchronized data acquisition, allowing time-resolved experiments. The complexity resides in using any arbitrary combination of movable elements, experimental channels, and detectors, generating the required triggers and evidently, allowing the coexistence of slow interpolated software-triggered channels and "virtual" elements like pseudocounters and pseudomotors.

EXPERIMENT CONTROL WITH SARDANA

Sardana has been designed for suiting large installations like a particle accelerator, or smaller such as experimental stations, up to small labs. Different characteristics and necessities of these type of facilities require from Sardana high flexibility and scalability.

In order to speed up the learning curve many Sardana features were inspired on SPEC [5], a complete and powerful software tool, very popular in building control systems for X-ray and neutron experimental stations. But Sardana goes beyond its functionalities. Its architecture is based on the client-server model with Tango [6] as the middleware. This allows to balance the workload in a distributed manner between multiple clients and multiple servers on both Linux and Windows platform PCs.

Sardana had proven to be a solution for synchrotrons, with its successful use in all beamlines, the accelerators and the peripheral laboratories of the Alba synchrotron [7]. It is also used in other similar installations like DESY in Germany, MaxIV in Sweden Solaris in Poland.

Taurus Based User Interfaces

Users can choose between Taurus based command line (CLI) and graphical interfaces (GUI) for interacting with the Sardana system.

From one hand exist turn-key applications. Spock, a single point of access CLI, provides the total control over the system. It allows to run the control procedures and interact with the laboratory elements. Its syntax tries to mimic as much as possible the SPEC commands, what makes the user transition between the two systems easier. A collection of Sardana specific, Taurus based, widgets (many of them as well available as standalone applications) includes: the experiment configuration tool, the scan plots, the user procedure editor, executors and sequencer as well as the elements' control widgets e.g. a generic motor widget.

From the other hand, users may create their own complete Sardana GUIs without the need to program a single line of code. The TaurusGUI framework [3] offers a wizard-based GUI creation process, in which the user just needs to specify which Sardana system the application should interface to and which graphical synoptic should it use. The application gets automatically populated with the previously listed widgets and the instruments' control panels bidirectionally connected with the graphical synoptic. At the application runtime, user can easily navigate in the laboratory by selecting instruments' panels via clicks on the synoptic.

Standard and Custom Experiment Procedures

Experiment and control procedures in Sardana are programmed as Python scripts and are called macros. The MacroServer manages available to the user macros and executes them either sequentially or simultaneously on the user request. Macro development and execution

ISBN 978-3-95450-148-9

provides many attractive features like the macro input and output parameters, user interaction at runtime, hook extension points, data exchange or plotting.

Sardana provides a miscellaneous set of standard macros including generic, n-dimensional scans available in various modes: step, hybrid and continuous, where distinct actions like motion, data acquisition and data storage are synchronized and optimized. Sardana allows the scan data to be stored in many different formats. This process is handled by one or more optional *recorders*. They receive the scan data and the experiment metadata from the scanning macro and transfer them to the destination e.g. a file, console output or to a data post-processing program.

Unified Access to the Hardware

Sardana interfaces all the equipments via the Device Pool (Pool) server and its plug-in controller classes. Instances of the Sardana controller classes may group many elements of the same type (see Table 1 for the list of the Sardana element types) analogously to what the hardware may do e.g. a single motor controller may control many moveable axes. Each of the Pool elements has a corresponding Tango device what facilitates its control to the Taurus clients.

Table 1: Sardana Element Types and its Examples

Element Type	Example of application
Motor	stepper, servo or piezo actuator
PseudoMotor	energy, HKL of a diffractometer, slit's gap or offset
CounterTimer	event counter, position measurement
PseudoCounter	vertical beam position in the X-ray beam position monitor (XBPM)
0DExpChannel	analog to digital converter (ADC), low current electrometer
1DExpChannel	position sensitive detector (PSD), multichannel analyzer (MCA)
2DExpChannel	CCD camera, 2D X-ray detector

The Pool optimizes common control actions. The grouped acquisitions are handled by the MeasurementGroup (MG) elements, which configure and synchronize the experimental channels previously selected by the user. Grouped movements are also synchronously started and controlled by the Pool's motion action. The API of the controllers and the actions' algorithms allow access the hardware efficiently.

Sardana system may consists of many distributed Pools. This architecture is successfully used at Alba to control the beamline elements in one Pool and the insertion device (ID) elements in another Pool, within the same Sardana system.

DEVELOPMENT APPROACH

Both Sardana and Taurus were conceived as Alba's internal projects. While gaining interest of other facilities, its development and decision-taking was opened to a community of users and developers. Discussions about the critical improvements and modifications in Sardana are organized and formalized around public process called Sardana Enhancement Proposal (SEP).

Design and development of the generic continuous scan framework is organized as SEP6. Alba drives this enhancement in a close collaboration with Desy. An internal to Alba *Scrum* team of 4 developers is in charge of this and other projects [8]. The complexity of the problem requires exploration and learning from experience. The iterative and incremental development allows to periodically deliver an improved version of the continuous scans to the beamlines. The evolving design was adapted to the user feedback and the complete model of the framework is presented in continuation.

The following limitations were assumed at the beginning of the project in order to reduce its scope and be able to deliver a working and quasi-backwards compatible version of Sardana to the laboratories as soon as possible:

- All the elements, but slave motors, participating in the continuous scans must be defined in the same Pool.
- Physical motors maintain uniform velocities while scanning – no trajectory control.
- All the experimental channels present in the MG share the same integration time across the scan point.

GENERIC CONTINUOUS SCAN MODEL

Three main actors participate in the continuous scan measurement and are controlled by their corresponding actions. The moveable objects vary the actuators' set points – motion action. The experimental channels acquire the measurement signals – acquisition action. The trigger or gate synchronizers (either software or hardware) are in charge of controlling the acquisitions based on the moveables' position updates – synchronization action.

Transparency

Users can easily switch between the scanning modes, by simply changing the macro name. Continuous scan names result from adding the *c* suffix to the step scan names e.g. ascan**c**. The order and meaning of the input parameters are preserved unchanged and the scan data outputs and files have identical formats. The same MG definition should be equally valid for both step and continuous scans (if the hardware allows that).

Generic Scan Framework (GSF)

Sardana implements a set of classes, called GSF, responsible for: parameters calculation, configuration,

ISBN 978-3-95450-148-9

execution and data storage of the scans. Both standard and custom scan macros extensively use the GSF what limits the experiment logic necessary to be programmed in the macro code.

N-dimensional equidistant continuous scans receive as parameters: the motor names, the scan ranges, the number of scan intervals and the integration time per scan point. These parameters are not sufficient to configure the involved elements. In order to calculate the scan parameters the GSF interrogate the involved physical motors for their maximum velocities, acceleration and deceleration times. The active MG and its experimental channels are interrogated for the *wait time* and the *re-arming* times respectively.

The scan is executed and controlled by the GSF in the following order, allowing users to interrupt it at any time:

- Move physical motors to the *pre-start* positions at the nominal speed.
- Configure all elements with the scan parameters.
- Start the experimental channels and later the synchronizers.
- Move physical motors to the *post-end* positions with the adequate parameters.
- When the scan finishes, move the physical motors to the end positions at the nominal speed.

During the scan, GSF progressively receives data updates reported by the experimental channels. They get classified into scan records and passed to the corresponding recorders. In some cases data interpolation may be applied in this process, always preserving information about the original data.

Motion

Any combination of Sardana motors and pseudomotors could be used in continuous scans. The following attributes: acceleration time, velocity and deceleration time are configured in a way that all the motors reach and leave the constant velocity region at the same time. While the acceleration and deceleration times are common to all the motors (based on the slowest motor), the velocities may differ between moveables (adapted to the necessary displacement).

The motion action controls whether the movement has finished and interrogate the motors for the position updates. The frequency of these queries should be configurable per motor due to its impact on the accuracy of the software synchronization.

Measurement Group and Configuration

MG aggregates experimental channels and either software (SW) or hardware (HW) synchronizers. Its role is to manage the configuration and supervise the acquisition and the synchronization actions. Coexistence of both types of synchronizers, e.g. as in the Fig. 1, is highly desired in many experiments.

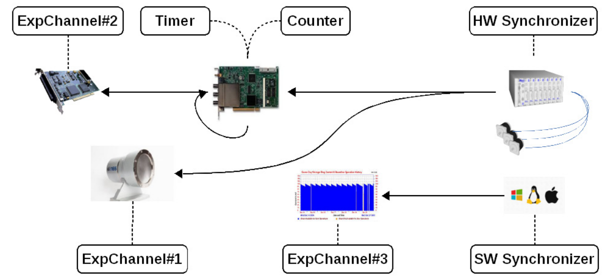

Figure 1: Example of the MG elements involved in the continuous scan.

Configuration of the MG, apart of other parameters, contains all the necessary information to describe the relation between the synchronizers and the experimental channels during the measurement. It is organized in a table with 1-to-1 relation between these elements e.g. Table 2. Single acquisition may be controlled by either *trigger* or *gate* signals. In case of the first one, the experimental channel is notified by the synchronizer when to start and decides on its own (based on the internal integration time control) when to stop acquiring. In the latter case, the experimental channel is also notified by the synchronizer when to stop.

Table 2: MG Configuration Corresponding to the Example from Figure 1.

Experimental channel	**Control**	**Synchronizer**
Timer	Trigger	HW Synchronizer
ExpChannel#1	Trigger	HW Synchronizer
Counter	Gate	Timer
ExpChannel#2	Gate	Timer
ExpChannel#3	Trigger	SW Synchronizer

MG contains other parameters used during the continuous scans: wait time, moveable reference source and synchronization. The first two smartly apply when the software synchronizers are in use. The first one expresses the desired wait time between the two acquisitions. It may help to avoid acquisitions being skipped in case the previous one is still in progress, but have a side effect and slows down the scan. The second one indicates the reference motor which position updates are used by the SW synchronizer to determine whether to emit start and stop signals.

Synchronization

A new Sardana element type TriggerGate (TG) was defined. It represents devices with trigger and/or gate generation capabilities. Their main role is to synchronize acquisition of the experimental channels. The synchronization characteristics could be described in either of two configuration domains: time or position. In the time domain, elements are configured in time units (seconds) and generation of the synchronization signals is based on passing time. The concept of position domain is

ISBN 978-3-95450-148-9

based on the relation between the TG and the moveable element. In the position domain, elements are configured in distance units of the moveable element configured/connected as the reference source (this could be: mm, mrad, degrees, etc.).

The synchronization data structure is prepared by the GSF and passed via MG to all the involved TG controllers. It is composed from the groups of equidistant acquisitions described by: the initial point and initial delay, total and active intervals and the number of repetitions. Due to the high flexibility of the TG hardware controllers, the synchronization description contains redundant information e.g. initial point and initial delay, always expressed in both: time and position domains. This way the controllers may choose which parameters and in which domain to use.

Acquisition

For the continuous scan needs the experimental channel controllers define the re-arming time (minimum time between the two consecutive HW controlled acquisitions) and whether the data readouts while acquiring are supported.

The acquisition action configures and controls the measurement process. In its start phase the following steps are executed:

- controllers are informed whether SW trigger or gate, or HW trigger or gate will control the acquisition (determined from the MG configuration)
- controllers are configured with the number of acquisitions to be executed
- integration times are loaded to all the channels controlled with the trigger signals
- start sequence is called on all the channels

The loop phase of the acquisition actions proceeds with the following steps for the channels still acquiring:

- call the state sequence in order to determine if the acquisition has finished
- call the read sequence to obtain the new data (if the controllers allow that)

Finally channels which do not support data readouts while acquiring, are interrogated for all the data.

Data Merging

Every value acquired during the continuous scan is stamped with an absolute time and the acquisition index. Data coming from the experimental channels synchronized by hardware provide the core part of the records. The software synchronized channels do not guarantee to provide data for each record. The GSF assigns data into records based on the index. The zero order hold [9] (constant interpolation) method is applied in order to fill gaps left by the skipped acquisitions. The real data must be easily distinguishable from the interpolated one, so each recorder could store/visualize them in its own way.

CURRENT STATUS AND FUTURE PERSPECTIVES

While the design of the generic continuous scan model is complete, the implementation is still ongoing. Its increments are gradually deployed in three Alba's beamlines.

The material science beamline (BL04) executes high resolution powder diffraction experiments (Fig. 2) in the continuous scanning mode. The measurement group involves fifteen hardware triggered counters and an arbitrary number of software triggered 0D experimental channels (mainly used for the sample environment monitoring). Mixing hardware and software synchronization gives a precise information about the sample experiment conditions (e.g. temperature) during the scan. Data post-processing programs calculate max, min and average temperature and exclude scan points based on temperature outliers from a given window.

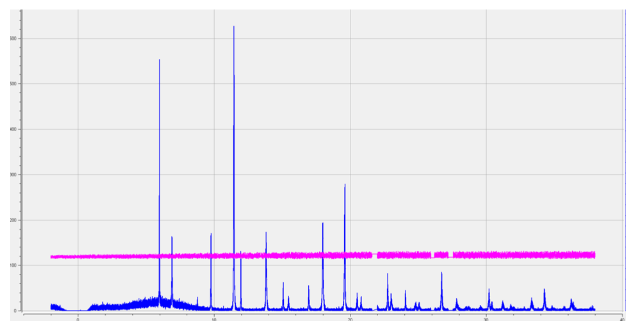

Figure 2: Diffraction patter (blue curve, left ordinate) and temperature (pink curve, right ordinate) measured during continuous scan of the diffractometer outer circle (abscissa) at BL04 – Alba.

The presented model covers the needs of equidistant continuous and time scans. Exchange of the parameters calculation layer in the GSF and use of the multiple groups in the synchronization description will allow non-equidistant scans. It is planned to allow trajectory control by varying physical motors' velocities while scanning.

ACKNOWLEDGEMENT

We would like to thank the Alba beamline scientists – especially François Fauth, Laura Simonelli and Manuel Valvidares for their valuable feedback during the commissioning. We would also like to thank the Controls and Electronics Sections of Alba for their active work in this project: Sergi Rubio, Fulvio Becheri, Roberto Homs, Daniel Roldan, Jordi Andreu and Xavier Serra. We also appreciate the Sardana Community feedback: Teresa Nunez, Thorsten Kracht and Jan Kotanski from Desy, Alejandro Homs, Tiago Coutinho and Jens Meyer from ESRF.

ISBN 978-3-95450-148-9

REFERENCES

[1] T. Coutinho et al. "Sardana, the Software for Building SCADAs in Scientific Environments", WEAAUST01, ICALEPCS2011, Grenoble, France, (2011)
[2] D. Fernandez-Carreiras et al. "Synchronization of Motion and Detectors and Cont. Scans as the Standard Data Acquisition Technique", WECOAAB03, ICALEPCS2013, San Francisco, USA, (2013)
[3] C. Pascual-Izarra et al. "Effortless Creation of Control & Data Acquisition Graphical User Interfaces with Taurus", THHC3O03, ICALEPCS2015, Melbourne, Australia, (2015)
[4] Python website: http://www.python.org
[5] SPEC website: http://www.certif.com/spec.html
[6] Tango website: http://www.tango-controls.org
[7] Alba website: http://www.albasynchrotron.es
[8] G. Cuni el al. "Introducing the SCRUM Framework as Part of the Product Development Strategy for the ALBA Control System", MOD3O04, ICALEPCS 2015, Melbourne, Australia, (2015)
[9] Wikipedia article "Zero-order hold": https://en.wikipedia.org/wiki/Zero-order_hold

ISBN 978-3-95450-148-9