ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG012

USE CASES FOR CONSISTENT ROBUST PROCESSING OF DATA MODELS

P. Schnizer*, G, Rehm, W. Sulaiman Khail
Helmholtz-Zentrum Berlin, BESSY, Berlin, Germany

Abstract

Many control algorithms or optimisation procedures profit from a consistent set of data which is available with a high frequency: e.g. machine learning or automated commissioning. Modern distributed control systems allow combining and presenting data based on data models, which are then transported consistently over the network: e.g. EPICS7 introduced these data models as normative types or their combination.

The authors present use cases that can profit from a consistent robust combination of data sub-models of many devices to a higher order model. Finally common patterns are presented which could be reasonable to implement independently.

INTRODUCTION

Distributed control systems are used for operating complex machines e.g. accelerator complex or large scale experiments. These machines have hundreds to thousands of actuators (e.g. the number of magnets of a large 4th generation light source) or hundreds of dedicated diagnostic instruments (e.g. beam position monitors for modern light sources [1–5]).

All these devices work together for a common target: to provide e.g. stable beam with low emittance for a fourth generation light source. Thus one not only needs to control the individual device or collect data from an individual instrument but rather one wants to operate them as a whole or at least a subset of it: e.g. to get an update of the current orbit or to apply a new consistent settings to the steerers to drive the orbit closer back to its desired positions. Here one can see that these instruments work as an ensemble or need to be orchestrated.

These instruments exhibit their state over typically an extensive set of process-variables where only a certain subset of them is typically needed for studying and describing the machine: these are typically provided in a flat name space, which is sufficient when addressing a single device or instrument.

For real machines, however, hundreds of devices are combined e.g. beam position monitors to an orbit object. One could see it as a line camera producing data over time. Combining this data is rather trivial as long as communication issues can be neglected. Typical instruments of today provide their measurement together with a time stamp which is synchronised over all of them. Furthermore, modern devices deliver data in a data model.

* pierre.schnizer@helmholtz-berlin.de

Data from many devices in a family can be combined, especially if all data packages that are part of a measurement contain the same tag. Handling it in a simple and efficient manner is not so trivial any more, in particular in case soft real time latency is required. Furthermore, such combiners should be able to deal with non-responsive devices or devices sending obviously faulty data or with delay.

Modern distributed control systems support communication using data published in models. EPICS 7 calls these ”normative types” [6–9]. Thus an orbit model can be built as an aggregate of the beam position monitor sub-models. Using these data-models changes of the sub-model could then propagate through the whole on-line or even off-line analysis chain.

Similarly, data with time tags could be combined, filtered, analysed and delivered to the data-sink. These data analysis pipelines are typically based on streams. Although these could handle delays and jitters, their end users can depend on a swift delivery: Control loops require a timely flow of data to achieve their optimisation tasks within a required frequency band. Training reinforcement learning agents typically needs many steps (1 000 – 10 000 steps), thus fast swift data combination can make tasks feasible, which can be out of reach if data combination takes too long.

This article is based on the authors' experience combining data from their BPM's into an orbit object using an in-house developed application. We have the impression that it could be worthwhile to consider that certain aspects of this task are common tasks which would be best provided within the distributed control system environment.

HANDLING DATA MODEL DATA: AN OVERVIEW

Distributed control systems of today allow publishing data in a structured manner: e.g. as normative types for EPICS. Then these structured data or data models are published as a consistent set by a single node.

We think that it would then be helpful if we could work on this data in a straightforward manner to

- **A**ggregate single data models to a consistent larger one
- **T**ransform contents of the data models
- **M**odify the data model e.g. enriching the data model with metadata or remove sub parts of the model that could be confusing further down the processing line.

USE CASE: ORBIT OBJECT

We see creating an orbit object as an example of aggregation. We build a combined data model from single sub-models ensuring that these are consistent.

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG012

The Ideal World

In the ideal world:

- All beam position monitors dispatch their measured beam position with appropriate time stamp for the aggregator.
- The aggregator combines all packages. As soon as all individual packages have arrived it creates a new instance of orbit object.
- The orbit object is received by a controller which produces a new set of target steerer values.
- The steerers receive their new value and deliver it timely and consistently to the beam.

The Not So Ideal World

In the real world aggregations are more involved. Here only the aspects of data combination and data delivery are addressed.

1. A beam position monitor can have stopped working.
2. A beam position monitor is sending data with wrong time stamps or identifiers.
3. A beam position monitor has been brought back to service.
4. A steerer is not reacting to the delivered data.

USE CASE: HANDLING MAGNET CROSSTALK

Modern machines, e.g. 4th generation synchrotron light sources, install magnets so close to each other that the magnetic flux from one magnet links to neighbouring magnets [10]. The flux change affects the magnetic field of the neighbouring magnet to such extent that it can not be ignored. This problem is addressed today implementing a correction loop in the control system, which then corrects the artifact.

Analysis shows that this task requires

1. aggregating the sub-models to a consistent data model
2. transform the inputs to take the cross talk into account
3. modify the resulting sub-model so that only the information required is delivered to the power converters.

This split up then would mean that steps 1 and 3 could be used from a standard tool set provided by the control system processing units. Only the step 2 is then left to the user.

We will discuss further down the different issues in real world why we think these two are not so easy to process.

USE CASE: HANDLING INSERTION DEVICES

The groups designing and building Insertion Devices (undulators or wigglers) take a lot of effort to build them in such a manner that any unwanted effect of the beam is minimised. Dedicated devices (corrector magnets) are then used to compensate the remaining deterioration caused by the undulators. These artifacts are partly mitigated by adjusting the set-point of some accelerator magnets. The control system takes care that these values are provided to the different parts of the system.

Here typically the insertion device publishes a consistent set of values. These need then to be further processed and provided at the locations where needed.

Analysis of performance could make use of aggregators again. If the different devices provide data with consistent time stamps, aggregation could ensure that a consistent set of applied corrections can be provided to an analysis algorithm or self-adaptive system.

SO WHAT'S NEEDED?

We described above orbit correction, where we have collected ample experience. We will explain what we think what is required focusing on this use case in particular.

Orbit feedback, in particular its fast version, is often realised at high speed in dedicated control and hardware, we see it as a good example for explaining basic aspects that we see applicable to other systems.

Aggregating Data-Models in a Fault Tolerant Manner

Aggregating the data is straight forward as long as everything works smoothly (see also Fig. 1). We think that such an aggregator should be based on the following assumptions:

- The aggregation should be based on its sub-models. Changes of the sub-models should be passed automatically through. One can consider that adaption only happens on e.g. external request.
- Aggregation should focus first on integrity then on timely delivery.
- Missing items shall be clear to down-stream consumers.
- Aggregators should communicate clearly what they are processing, what they are waiting for, or which devices they consider as non responsive.
- Aggregators should be able to combine packages by arrival time. These should contain a manager that estimates average arrival time and its distribution. The maximum deviation threshold from the average arrival time should be configurable.
  This mode could be seen e.g. as a fall back mode if the controller realises that sorting by data-identifier (or internal timestamp) is not reliable.

Making Devices Ready for Aggregation Individual devices (here the electronic boxes reading the beam position monitor system) can fail and be brought to service. Typically devices of an ensemble emit their data in parallel as these are commonly triggered. Here functionality should be foreseen which allows getting one member "part of the choir" again: It needs to be informed which stamp and perhaps increment to use starting from the next trigger.

This functionality could be (partly) implemented inside the aggregator. It would then monitor if the device is back in good service and then include its data into its ensemble. Furthermore it could provide information what it expects as

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG012

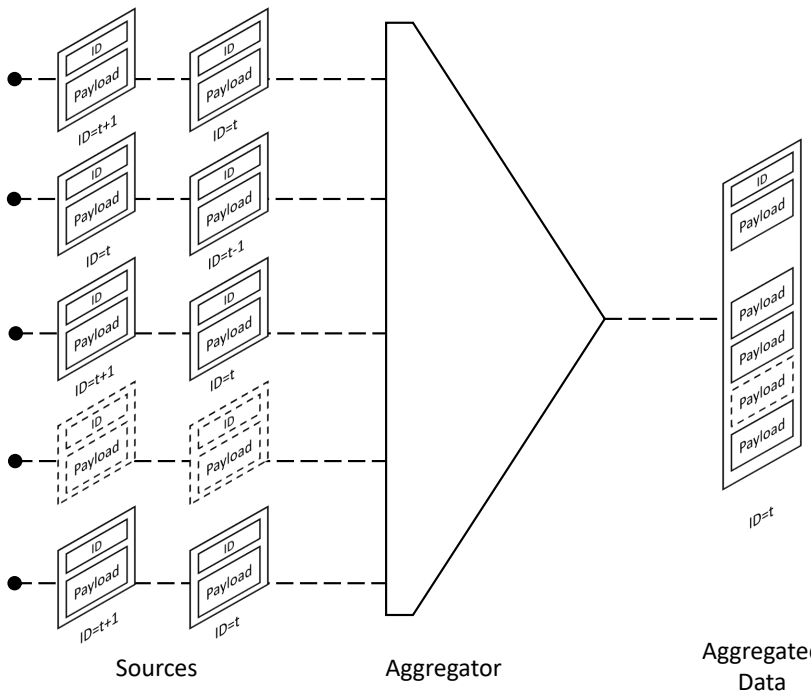

Figure 1: Aggregation: different data sources are publishing their data model. For aggregation one can consider that each document consists of an id and a payload. These have to be aggregated by "id" which is identical for data model which are member of the same set. Please note that some source may not be delivering data (indicated by the document in a dashed line or being delayed for some reason (as for the second sender).

next update based on the majority vote of the participating devices (e.g. using linear interpolation).

Modifying Data-Models

Data models can need to be modified: e.g. with some meta-data. This e.g. would be nearly possible on the EPICS side using the Q:Info fields. What is missing that it is just added to an existing model. Changes of the down stream model should be automatically propagated through.

In similar manner one could see recalculating different parts of its values. Here again one would only affect certain aspects of the data model, but be able to leave the rest intact.

Furthermore calculations would need to process different fields or submodels. Publishing of the model should then happen in a consistent fashion again.

Filtering Data-Models

Down-stream consumer can perhaps be selective on the data-model content passed to them:

- The consumer could not accept data models with field entries it does not know.
- They could expect a different content in some field.
- The data should not be made available to the down stream user as it e.g. could overwrite configuration data, which one could consider as a better use.

All these requirements make it necessary to

- remove fields from the model or
- only select a subfield or some subfield components
- or rearrange the received data to a different data model.

CONCLUSION

Modern distributed control systems allow transmitting data consistently within a structure or data-model. These data are provided by different sources: e.g. beam-position monitors and aggregated to higher order models. Furthermore, these models are transformed or modified further down the processing line.

We consider aggregation and modification of data models steps that are happening frequently within data processing pipelines realised within control systems. These are rather straightforward to implement if one does not consider failure scenarios. Handling these failure scenarios then makes them more complex, thus we believe that it is worthwhile to provide the aggregation and modification step within the control system.

ACKNOWLEDGEMENT

We would like to thank Thomas Birke, Malte Götz, Timo Korhonen, Ralph Lange, Roland Müller, Luca Porzio, Markus Ries, and Andreas Schälicke for fruitful discussions.

ISBN: 978-3-95450-255-4 ISSN: 2226-0358 doi: 10.18429/JACoW-ICALEPCS2025-WEMG012

REFERENCES

[1] K. Ha, L. R. Dalesio, J. H. D. Long, J. Mead, Y. Tian, and K. Vetter, "NSLS-II Beam Position Monitor Embedded Processor and Control System", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, pp. 932–934. https://jacow.org/icalepcs2011/papers/WEPMN024.pdf

[2] K. Vetter *et al.*, "NSLS-II RF Beam Position Monitor Update", in *Proc. BIW'12*, Newport News, VA, USA, Apr. 2012, pp. 238–241. https://jacow.org/%20BIW2012/papers/WECP02.pdf

[3] F. Aureo, V. Hardion, M. Lindberg, R. Lindvall, R. Svärd, and C. Takahashi., "Control System Suite for Beam Position Monitors at MAX IV", in *Proc. IBIC'22*, Kraków, Poland, pp. 496–499, 2022. doi:10.18429/JACoW-IBIC2022-WEP38

[4] W. Cheng *et al.*, "Beam position monitoring system and beam commissioning at APS-U storage ring", in *Proc. IPAC'24*, Nashville, TN, USA, pp. 2170–2172, 2024. doi:10.18429/JACoW-IPAC2024-WEPG01

[5] M. Abbott, *Concentrator for beam position monitors at Diamond Light Source*, private communication, 2023.

[6] L. R. Dalesio *et al.*, "EPICS V4 expands support to physics application, data acquisition, and data analysis", in *Proc. ICALEPCS'11*, Grenoble, France, pp. 1338–1340, 2011. https://jacow.org/icalepcs2011/papers/FRBHMULT06.pdf

[7] J. Dalesio, "EPICS version 4 - implementing complex data types", EPIC Consulting, Tech. Rep., 2012. doi:10.2172/1055720

[8] M. Kraimer, "Brief history of EPICSV4", presented at EPICS Collaboration meeting Fall 2016 ONRL, 2016. https://conference.sns.gov/event/11/contributions/36/

[9] R. Lange *et al.*, "Five years of EPICS 7 – status update and roadmap", in *Proc. ICALEPCS'19*, Cape Town, South Africa, pp. 1087–1092, 2023. doi:10.18429/JACoW-ICALEPCS2023-TH1BCO01

[10] G. Le Bec, J. Chavanne, S. Liuzzo, and S. White, "Cross talks between storage ring magnets at the extremely brilliant source at the european synchrotron radiation facility", *Phys. Rev. Accel. Beams*, vol. 24, no. 7, p. 072 401, 2021. doi:10.1103/PhysRevAccelBeams.24.072401