ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA084

DECOUPLING CERN ACCELERATORS

A. Dworak*, JC. Bau, CERN, Geneva, Switzerland

Abstract

The accelerator complex at CERN is a living system. Accelerators are being dismantled, upgraded or change their purpose. New accelerators are built. The changes do not happen overnight, but when they happen they may require profound changes across the handling systems. Central Timings (CT), responsible for sequencing and synchronization of accelerators, are good examples of such systems.

This paper shows how over the past twenty years the changes and new requirements influenced the evolution of the CTs. It describes experience gained from using the Central Beam and Cycle Manager (CBCM) CT model, for strongly coupled accelerators, and how it led to a design of a new Dynamic Beam Negotiation (DBN) model for the AD and ELENA accelerators, which reduces the coupling, increasing accelerator independence. The paper ends with an idea how to merge strong points of both models in order to create a single generic system able to efficiently handle all CERN accelerators and provide more beam time to experiments and LHC.

TIMING AT CERN

Raison d'être

The General Machine Timing (GMT) or just Timing is one of the core components of the CERN control system. It has two main functions: (1) it is responsible for the precise synchronization of the equipment guiding beams in the accelerators, and (2) beam scheduling, or sequencing, i.e. deciding which particle beam to produce in a given accelerator at a given time.

How Does it Work? Cycles, Beams and Sequences

The synchronization is achieved through services called central timings (CT). For each accelerator they produce events which are distributed via a timing network. The events are received by dedicated hardware which is able to produce electrical pulses or to trigger real-time (RT) software tasks, both used to control the equipment around the accelerator with the required precision. In parallel to the events, the CT also sends telegrams. While events concern a point in time (what happens at that very moment), the telegram describes what is happening during a period of time. As such it gives context information to events occurring during its validity. For example, telegrams are used to distribute information about particle types in an accelerator, or to which accelerators/experiments the particles are going to be sent next. Telegrams are distributed in regular intervals called basic periods (BP), which hence constitute a limit to their granularity. For most of the accelerators a BP length is 1.2 seconds.

Although a very interesting subject, this paper will not delve deeper into the synchronization. Instead, it will take a closer look at the sequencing. To understand it better, we need to introduce a few key concepts used when scheduling particle beams. The first one is called *cycle*. A cycle precisely describes the behaviour of an accelerator during a period of time. Typically this includes preparation of the accelerator to receive particles, particle injection, acceleration and extraction towards the next machine. The description is provided through a list of timing events. Each event, when received by a timing client, may be interpreted as a request to perform a specific action. For instance, it may trigger ramping of magnets, or request preparation for a coming injection.

At CERN, cycles of the small accelerators (Booster, PS, LEIR) last between one and three seconds while SPS cycles take around ten seconds, and AD cycles around two minutes. LHC cycles, although the same from the physical point of view, often take over ten hours to execute. Because of that they are handled by the timing system in a different way, and are not going to be the subject of this article.

It is clear that a cycle describes behaviour of one accelerator in separation. To describe how particles should be accelerated from the very beginning in Linacs, up till the very end when they are collided in experiments, a structure called *beam* must be prepared. The structure may be represented as an organised list of cycles of accelerators through which particles are going to be passed. The involved cycles, their number and structure are selected in a way to guarantee the characteristics of the beam as required by the experiments.

The big and flexible CERN accelerator complex [1] makes it possible to run a number of experiments in the same period of time. Depending on the agreed physics program, the operation team has many options regarding which beams to execute and in what order. Sequencing is the very process of selecting and ordering the beams. The following chapters present the two main sequencing models supported at CERN. The emphasis is put on each model's functionality, and strong and weak points. The results of the analysis are used later on to propose a unification of the two systems, with the main goals of providing more beam time to the experiments and simplifying the Timing software stack.

For the overview of the presented concepts have a look on the Figure 1.

THE CBCM MODEL

The CBCM model [2], is used by accelerators in the so called LHC Injector Chain (LIC) group: Linac2, PS Booster, Linac3, LEIR, PS and SPS. These machines usually work closely together to produce the beam, notably for the LHC [3]. The CT that implements the logic has been in op-

* andrzej.dworak@cern.ch

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA084

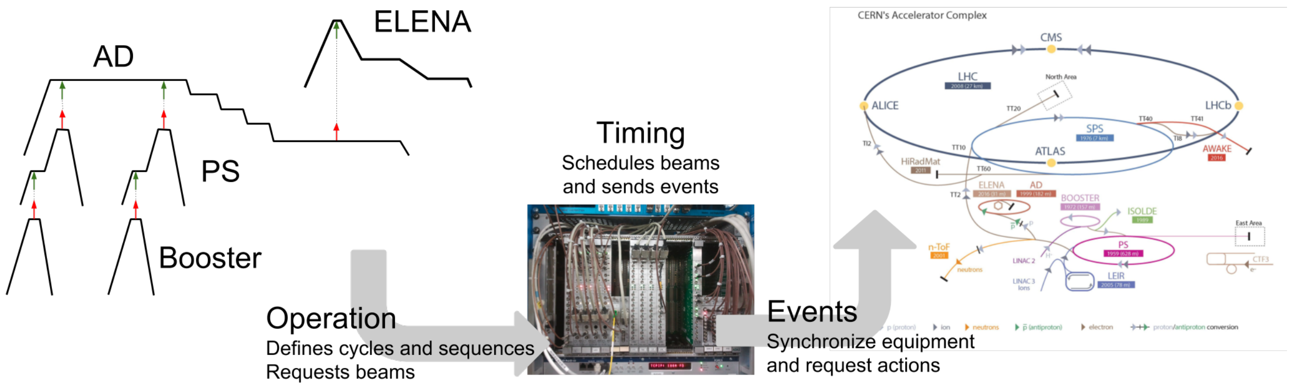

Figure 1: The operation teams program the central timings, which calculate and publish timing events. The events are received by equipment which drive accordingly the CERN accelerator complex.

eration for almost 20 years, is well-known by the accelerator operators, and has many interdependent applications. These factors do not directly influence the logic of the model, but must be taken into account when the proposed upgrades are going to be discussed.

Functionality

Key components of the CBCM model are GUI applications provided to the operation, which allow definitions of cycles, beams, and sequence diagrams. When defining a cycle an operator is constrained to define the cycle length in terms of basic periods (the same which are used to distribute the telegram information). There is no such constraint for the events which are set anywhere in or around the cycle as required by the operation and concerned equipment. The created cycles are merged into beams, which in turn are grouped into the so-called sequence diagrams.

In the CBCM model a request to execute a beam is made by sending a sequence diagram to the CT. The CT executes all the beams as defined in the sequence, and when finished it starts anew from the beginning. The CBCM provides also other ways to execute a sequence, but for our discussion this simple example is enough.

Limitations

To understand better how the CBCM functions, and what are the potential limitations let us look at some concrete examples.

To fill LHC with particles [4] the accelerator operators define the parameters of the requested beam using the LHC Sequencer application. The request is passed to the LHC CT, which requests particles from the SPS. A nominal filling pattern consists of 12 injections from the SPS to the LHC per LHC ring. Each injection from the SPS comes from a single SPS beam. This beam has 2, 3 or 4 batches injected from the PS, which corresponds to 2, 3 or 4 PS cycles. For the nominal beam the LHC filling schema, as presented in the Figure 2, in terms of number of batches takes a sequence: 2, 3, 4, 3, 3, 4, 3, 3, 4, 3, 3, 4. Because of a relatively low responsiveness of the system, it is more efficient to reuse the

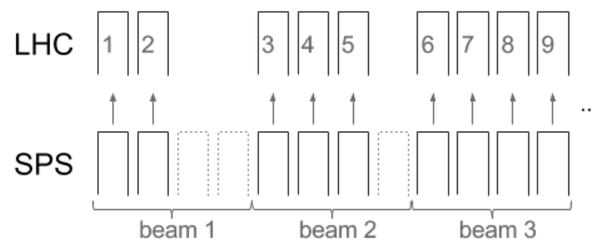

Figure 2: LHC nominal beam consists of 39 batches injected from SPS. The figure presents the first 9 batches and their mapping between SPS and LHC. Because of a small responsiveness of the CBCM model, it is faster to execute the same 4-batch cycle 3 times.

same beam with 4 PS batches, see Figure 3, for any of the 2, 3 and 4 batch PS injections into the SPS, and eliminate the unwanted batch by suppressing particles when not needed, than it would be to define each time the sequence exactly as needed.

The execution of this schema takes 24 SPS cycles of around 23 seconds (19 * 1.2 seconds), which sums up to around 10 minutes. If the system would allow to efficiently request what is really required, a total of 17 PS cycles each of 3 BP length, and 34 Booster cycles each of 1 BP length would be freed each time. That would be a gain of 1 minute for PS, and 40 seconds for Booster every time the beam is executed. Also, the SPS cycle could be shortened accordingly to the actual number of the PS batches.

The saved minutes do not seem a lot in comparison with a 10 hour LHC cycle, but they constitue almost 10% of the LIC time to prepare the beam. The yearly CERN run contains other beams where potential savings are of the similar magnitude. The integrated machine time to gain becomes considerable.

Another limitation is that the CBCM model sometimes blocks a time slot for a beam, but for some reason is not able to execute it. For example, sometimes the LHC beam is allocated and played in the LIC completely without particles. This happens especially around the times of the LHC fill, when the LIC machines are ready to deliver the beam to

16th Int. Conf. on Accelerator and Large Experimental Control Systems ICALEPCS2017, Barcelona, Spain JACoW Publishing
ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA084

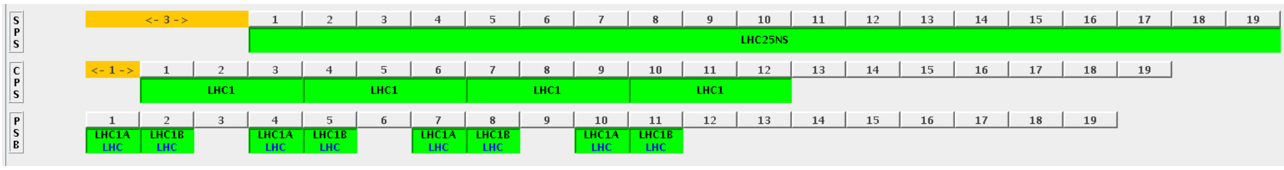

Figure 3: LHC injection beam structure programmed in the CBCM model.

the LHC, but wait for the injection request. Executing an empty beam may also happen in the case of interlocks or external conditions. Such situations are in most cases a waste of machine time, as there are experiments that could profit from the additional beam time.

A third example of a shortcoming of the CBCM model comes from the fact that cycles and sequence diagrams are built as multiples of the BP. This makes it impossible to optimize their length to a finer degree. The time lost per cycle is not big, but it grows linearly with the passing time.

THE DBN MODEL

Functionality

As discussed in [5] the AD accelerator [6] requires additional functionality on top of what the CBCM model is able to provide out of the box. Also, it operates with much longer cycles than those of the LIC machines. At the time when AD was a part of the LIC certain hacks were applied to "squeeze in" the AD cycles and to provide required functionality. They did the job, but when plans to build ELENA [6], a new accelerator connected to AD, were accepted, it became clear that the solution would not scale.

The introduction of the ELENA accelerator was an opportunity to propose a new CT model called *Dynamic Beam Negotiation* (DBN). The main idea was to exchange the predefined static sequence, with a schedule created dynamically with incoming beam requests. In more details, a request triggered by the accelerator operator (on behalf of the accelerator experiments) would describe exactly the beam needed at a given moment. The request would be sent to a CT of the top-most accelerator of a beam. The CT would contact the CTs of all dependent accelerators, negotiating the injection times between the machines, and so, would dynamically schedule the beam, Figure 4. This approach would make each CT more responsive and put less run-time constraints on other accelerators.

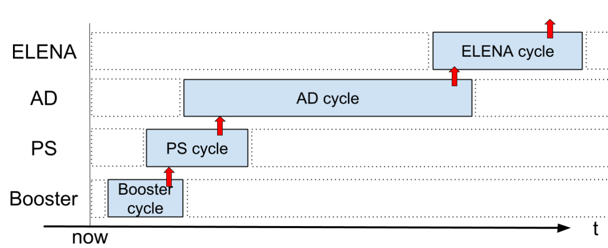

Figure 4: DBN schedules dynamically a beam for ELENA. Each CT reserves a required time slot. The unasigned time is available for other beams.

The general ideas were presented and discussed with the AD and ELENA operation groups [5, 7]. Analysis of usage scenarios helped to verify them, and to define a list of additional features required to operate the accelerators [8]. Some of the interesting features, especially those that contrast with the CBCM model or considerably extend its functionality, are presented in the Table 1.

DBN for AD and ELENA

The DBN model was implemented in a form of a reusable engine [7]. This allows to create a new instance of a DBN CT in separation from the existing ones. Also, any extensions or bug-fixes in the engine in one go improve all existing instances. The CT instance for AD was deployed and connected to the LIC CT in 2014. The CT of ELENA was first deployed in isolation in 2015 to be available for the local tests of the machine. Then it was connected to the CT of AD in 2016, to allow joint operation and beam negotiation. Since then the model has proven its efficiency by successfully executing (since the data is collected at the end of 2015) around 320,000 AD and 4,300,000 ELENA operational requests.

DBN for LHC Injectors

Migration of the LIC CT from the CBCM to the DBN model would be a good idea for a number reasons. First of all, as explained in the chapter on the CBCM model, it has the potential of providing additional beam time simply be optimizing the beam scheduling and removing the constraints as explained in the previous chapters.

Secondly, thanks to the independent scheduling of accelerators certain configurations are much easier to achieve. For instance, it can be used to test an accelerator (potentially without particles inside) in a complete isolation from the others. This configuration is possible with the CBCM but requires extra work. Another example concerns linacs, which at the moment are handled as one entity with their parent accelerators (e.g. Linac2 with Booster, Linac3 with LEIR). This does not allow them to work in isolation, which sometimes actually is the case (e.g. LBE/LBS measurement lines for Linac3). Programming a linac in isolation with its own CT could be beneficial in this case. As a third example, let us consider adding a new accelerator. In the DBN model it works out of the box and is safe. First the new CT is instantiated and tested in a complete isolation. Later, it is easily connected to the existing CTs through the well-defined interfaces of the beam negotiation mechanism, a scenario already verified by the AD/ELENA case.

Thirdly, a big gain would come simply from the removal of the CBCM model. It is clear that having one common

ISBN: 978-3-95450-193-9 doi:10.18429/JACoW-ICALEPCS2017-TUPHA084

Table 1: Comparison of functionalities of CT models

CBCM model	**DBN model**
A common sequence makes the machines strongly coupled. Decoupling is possible but requires extra work.	The machines are independent (decoupled).
Each accelerator is bound within a sequence of the same length and structure.	No common boundaries or interdependency on the sequence structure of other accelerators. No boundaries within a single accelerator as there is no sequence.
It is impossible to update the sequence of one machine in separation (update of all at the same time). The sequence is exchanged with a new one only at its end (low responsiveness).	Beams requested as needed, scheduled on the runtime independently for each accelerator. No concept of a static sequence.
Static sequence structure defined by the operation requires more human work and time.	Simpler interface for the operation which only requests beams assigning them priorities. Scheduling done automatically by the system. Possibilities for runtime optimizations.
Definitions of programmed beams (cycles and their connections across accelerators) are static.	Definition of a programmed beam (cycles and their connections) may dynamically change from request to request. Requesting a new beam is inexpensive.
The model assumes that particles have to be produced regularly, there is always someone waiting for them.	Whether the requests are regular or irregular, the system is flexible to quickly schedule beams and provide particles as requested.
Best suited to play short cycles between 1.2 up to a few seconds. Cycle length limited by the sequence length.	Suited to play cycles of any length.
The injection schema from one accelerator to another is static and predefined via the beam structure.	The injection schema can be static or dynamic. The later maybe useful in the case of multi-injection, when the relative time of injections is not known a priori.
Length of a cycle defined in BPs. A cycle may take more space in a sequence than really needed.	The length of a cycle does not depend on the BP. The space occupied in a sequence corresponds to the real needs.
Cycles have static lengths.	A cycle may dynamically extend its length, for instance through a pause mechanism.

system is easier than having two. This is true both for the developers and the users.

SUMMARY

The paper presented the main functionalities of the timing system at CERN. It concentrated on the sequencing especially as provided by the two models, the CBCM and the DBN. The capabilities of the two models were compared, and the differences illustrated with a few examples.

In the past the LIC was used in a much more regular way, and the CBCM was a simple and perfect solution to repeat the same sequence again and again. However, since the new types of accelerators were added and the number of experiments and beam types has increased and with them the complexity and the number of possible execution scenarios. The authors believe that the DBN model, with certain adaptations to the LIC environment, should be well suited to handle the growth, and it has a potential to provide more beam time, and simplify some of the timing use cases.

REFERENCES

[1] CERN Homepage - The Accelerator Complex `https://home.cern/about/accelerators`

[2] J. Lewis *et al.*, "The Central Beam and Cycle management of the CERN accelerator complex", in *Proc. ICALEPCS'99*, Trieste Italy, 1999

[3] J. Lewis *et al.*, "The evolution of the CERN SPS timing system for the LHC era", in *Proc. ICALEPCS'03* Gyeongju Korea, 2003

[4] P. Collier *et al.*, "Standard Filling Schemes for Various LHC Operation Modes (Revised)", LHC-Project Note 323 Revised, Geneva Switzerland, 2003

[5] A. Dworak *et al.*, "AD Central Timing Technical Specification", Geneva Switzerland, 2013, `https://edms.cern.ch/document/1721980/1`

[6] CERN Homepage - The Antiproton Decelerator `https://home.cern/about/accelerators/antiproton-decelerator`

[7] A. Dworak *et al.*, "ELENA Central Timing Technical Specification", Geneva Switzerland, 2016, `https://edms.cern.ch/document/1730903/0.1`

[8] A. Dworak *et al.*, "How do we program accelerators at CERN? Scheduling beams with the new AD central timing", CHEP-16 San Francisco USA, 2016