How do we program accelerators at CERN?
Scheduling beams with the new AD central timing

Andrzej Dworak and Jean-Claude Bau
CERN, CH-1211 Geneva 23, Switzerland
E-mail: andrzej.dworak@cern.ch

Abstract. The central timing (CT) is a system responsible for driving an accelerator behaviour. It allows operation teams to interactively select and schedule cycles. While executing a scheduled cycle a CT sends out events which provide (a) a precise synchronization and (b) information what to do - to all equipment operating an accelerator. The events are also used to synchronize accelerators between each other, which allows passing particles between them.

At CERN there are currently ten important accelerators. Each of them is different and has some unique functionalities. To support the variety and not to constrain operation teams there are three major types of CT systems. The one which has been developed most recently handles the Antiproton Decelerator (AD). Uniqueness of the AD machine comes from the fact that it works with antimatter and instead of accelerating particles it decelerates them. As a result, an AD cycle differs from other machines and required development of a new CT.

In this paper we describe the differences and systems which has been developed to support the unique AD requirements. In particular, a new AD CT is presented and functionality it offers to operation teams who program the machine. We present also the central timing extensions planned to support a new decelerator ELENA, which will be connected to AD to further slow down the particles. We show that with these extensions the new central timing becomes a very generic system. Generic to a point where it is valid to ask the question if it could be used as a common solution for most CERN accelerators.

1. Introduction
The CERN accelerator complex, see Figure 1, consists of a succession of machines that are used to accelerate particles to increasingly higher energies. The AD is a unique machine that decelerates the particles. It captures antiprotons, created by firing PS protons into a block of metal, and slows them down to allow to trap them [1].

In 2013 the AD accelerator was programmed by the same central timing that programs the LHC injectors (LIC CT): Linac2, Linac3, Booster, PS, SPS, and LEIR [2] [3]. Because of the differences between the AD and the LIC machines, for a summary see Table 1, the programming of AD using the LIC CT was neither easy nor optimal. In addition, the planned integration of a new accelerator, ELENA, which will allow to further slow down the antimatter, would complicate the old process even more. For these reasons, it was decided that the two accelerators (AD and ELENA) require a dedicated central timing system [4].
Figure 1. The CERN accelerator complex with the AD and ELENA machines in the center.

Table 1. Differences between the LIC and AD/ELENA cycles, and the way they are executed. It is clear that the old CT model is not well suited for efficient execution of AD/ELENA cycles.

| LIC cycles | AD and ELENA cycles |
|------------|---------------------|
| Short cycles between 1.2 up to a dozen of seconds. | Short and long cycles up to hundreds of seconds. |
| Predefined length of a cycle. | Length of a cycle may dynamically change because of a pause mechanism, or dynamically negotiated multi-injections. |

Particles have to be produced regularly, there is always someone waiting for them. Cycles are grouped in a plan called a supercycle which is constantly repeated. Cycles are predefined and continuously executed.

When ready, experiments request particles. This may be quite irregular. The system should be flexible to quickly schedule cycles and deliver the particles when requested.

Definitions of programmed cycles and their connections across accelerators are static. Definition of a programmed cycle may dynamically change from cycle to cycle. Defining a new cycle or updating an existing one should be inexpensive.
Figure 2. Decomposition of the AD timing. Specific accelerator logic extracted into cycle description and beam requests (both in green).

2. The new system

2.1. System components

CTs are complex systems. They use dedicated hardware, consist of low level real-time tasks, and non-trivial algorithms that must efficiently schedule and execute cycles. Schedules of many accelerators (most notably LHC injectors) are prepared by operation teams who, using dedicated GUI applications, define sequences of cycles to be executed. A CT system must verify the feasibility of execution of the sequence before it accepts it. Also, while executing a sequence there are many run-time parameters and rules that need to be verified. Algorithms that perform the actions are tightly coupled with the timing concepts, and at the same time they heavily depend on features specific to an accelerator. Because of that they require both expert timing and expert accelerator knowledge and therefore contribute considerably to the overall complexity of the system.

To create at once a more performant and simpler system, a strategic decision was made to decompose the AD CT into several subsystems, where each subsystem would require expert knowledge from only one domain. This decision helped to obtain a modularized solution with generic, easily reusable components and clear separation of responsibilities. Each subsystem was then developed by experts in the corresponding domain. Most importantly, the Operation team, which consists of accelerator experts, developed applications which encode most of the accelerator-specific knowledge. In particular, the applications provide cycle definitions, and depending on experiment needs, they prepare so called beam requests which specify how cycles belonging to various accelerators depend on one another. This approach allowed the timing team, which consists of software and controls engineers, to focus on developing the CT core system—a generic engine which efficiently schedules and executes cycles requested by the Operation team.

The new system was designed and developed in 2013. The AD CT was deployed in operation in 2014, and since then it successfully serves its purpose. In 2015 it was reused to provide the first version of the ELENA CT. In 2016 the generic engine was extended to provide a feature-full, operation-ready ELENA CT [6].

Figure 2 presents all system components and data typically exchanged between them. The components are:

- Cycle Editor - a GUI application used by operation to define cycles
Figure 3. A typical AD cycle (upper part of the Figure) as programmed by a machine operator, transformed to the timing representation (lower part of the Figure).

- LSA - a service that stores all the cycle definitions
- Make Rules - converts cycles to a format understood by the CT
- Beam Request Server (BRS) - requests the CT to play specific cycles
- PS CT the PS accelerator central timing (part of the LIC). The AD CT negotiates with it the time of particle injection (into AD)
- ELENA CT the ELENA accelerator central timing. The ELENA CT negotiates with AD CT the time of particle extraction (from AD)
- GUI apps, FECs hardware and software clients of the events and of the context data provided by the timing system

2.2. Defining a cycle

An accelerator expert defines a magnetic cycle using the Cycle Editor GUI application. They set the magnetic cycle (magnetic field in main dipoles), and anchor specific machine events (e.g. start electron cooler) at chosen offsets from the beginning of the cycle. The cycle description may contain additional information, i.e. particle type, harmonic number, accelerator mode, etc.

When the description is ready it is stored in the LSA database, from which it may be retrieved for further modifications, cloning, or inspection.

The LSA database may contain hundreds of cycle definitions, and history of their changes. The CT, being an embedded system, has more constraints and usually holds no more than tens of cycles. That is why, depending on the research program, the Operation team chooses some of the cycles stored in the LSA and sends them to the CT.

When a cycle is sent to the CT it is being transformed by the so called Make Rules. The Make Rules map a representation understood by the machine experts into a representation which is understood by the CT, see Figure 3. For example the specific machine processes are replaced with names of timing events which are going to be distributed to the clients.
2.3. Beam request and cycle scheduling

A cycle which is defined in the CT is ready for execution. To start the execution the CT needs a request which specifies which cycle to play and which may define additional parameters of execution. A request may come from the BRS, or from another CT (in the case of AD it is the ELENA CT which may request it). In both cases the request format is the same, but it may differ in how it is filled.

A requested cycle may be played with or without particles. In the first case the time when the cycle could be played is negotiated with a dependent CT. The CTs negotiate injection/extraction times. Of course, the dependent CT may have other negotiation dependencies. The final placement of a cycle depends on availability of all accelerators in the negotiation chain. In the case of AD that would be: Booster, PS, and AD; and for ELENA: Booster, PS, AD, and ELENA. The chain of connected interdependent cycles is called a beam (hence a beam request, not just a cycle request), see Figure 4; In the second case, when the cycle is played without particles, there are no dependencies on other accelerators. The requested cycle is scheduled as soon as possible, behind the last scheduled cycle, or between already scheduled cycles if the gap between the two is long enough.

2.4. Cycle execution

Figure 5 presents a typical AD cycle and states it goes through as it is being executed. A cycle which is scheduled gets a pending state. In this state it is still possible to cancel it (the cycle will not be executed). At a certain moment, when the cycle data has to start to be distributed to clients, and it is too late to remove it, the cycle enters a waiting state. It awaits its exact start and when the time comes it is executed. At that moment, the CT starts sending the events and context data which describe the cycle.

In case of a simple cycle, where there are no pauses nor multi injections, the cycle is executed linearly from the beginning till its end. In case of more complex cycles, for example cycles with more than one injection, a CT will repeat the negotiation process to schedule further injections.

A cycle which is executed without particles may be paused for machine inspection. It remains in a pause state as long as the operation does not release it. Upon release it returns to a standard execution mechanism.

Figure 4. Various cycles scheduled across a few accelerators. The blue cycles are connected into a so called beam which will guarantee passing of particles starting from Booster up to ELENA.
2.5. Timing distribution and reception

The new CT model does not influence the distribution, nor the client side part. Timing events are encoded, distributed and received by the standard CERN timing hardware and software components. They are used in the same way as in the other accelerators [5] to trigger interrupts, or to generate pulses using dedicated hardware components.

3. Conclusions

The new CT model used by the AD and ELENA central timings is a powerful tool. It abstracts definition of a cycle into a generic representation which is defined by the accelerator experts. Similarly, the beam requests are in the hands of the experts who, depending on the experiment requirements, prepare them. Thanks to that the CT engine stays free of accelerator-specific knowledge which makes it simpler and easier to comprehend than other CT models that incorporate the specific logic into their code.

The presented CT model is not only simpler but also a very flexible solution. It concentrates on the generic scheduling and execution algorithms, and guarantees efficient execution of cycles despite their length and complexity. Being free of the accelerator-specific knowledge it could be reused for other accelerators. This would greatly simplify and unify all the existing CT systems. Also, adding and integrating a completely new accelerator to the complex would be much simpler than it is at the moment. The details and feasibility of this solution still need to be studied.

References

[1] The Antiproton Decelerator, CERN home page, https://home.cern/about/accelerators/antiproton-decelerator
[2] Lewis J et al The Central Beam and Cycle management of the CERN accelerator complex, ICALEPCS-99 Trieste Italy, 1999
[3] Lewis J et al The evolution of the CERN SPS timing system for the LHC era, ICALEPCS-03 Gyeongju Korea, 2003
[4] Dworak A et al AD Central Timing Technical Specification, Geneva Switzerland, 2013, https://edms.cern.ch/document/1721980/1
[5] Lewis J et al The CERN LHC Central Timing a vertical slice, ICALEPCS-07 Knoxville, Tennessee, USA, 2007
[6] Dworak A et al ELENA Central Timing Technical Specification, Geneva Switzerland, 2016, https://edms.cern.ch/document/1730903/0.1