



**Detailed Specification of the
FAIR Accelerator Control System Component
„Settings Management System“**

Document Name
F-DS-C-03e

Date yyyy-mm-dd
2012-08-16

Abstract

This document is the Detailed Specification of the accelerator control system component “Settings Management System”. This work package is part of the “Control System Core System Software Packages” work package and covers the PSP codes 2.14.10.1.1 (Settings Management Framework) and 2.14.10.2.1 (Machine Models).



Table of Contents

1.	Purpose and Classification of the Document	4
1.1.	Responsibilities	4
1.2.	Classifications of Requirements	4
2.	Scope of the Technical System	5
2.1.	System Overview	5
2.2.	Limits of the System and Environment	6
2.2.1.	Limits	6
2.2.2.	Interfaces	7
2.2.3.	Environment	7
2.3.	Basis of Concept	7
2.3.1.	Functional Requirements	7
2.3.2.	Non-functional Requirements	10
2.3.3.	General Constraints	11
2.3.4.	Architectural Principles	11
3.	Technical Specifications	12
3.1.	Settings Management Framework	12
3.1.1.	Data import	12
3.1.2.	Accelerator model	12
3.1.3.	Calculation of Data	13
3.1.4.	Data persistence	13
3.1.5.	Data presentation	13
3.1.6.	Data supply	14
3.2.	Machine Models	14
3.2.1.	Circular Accelerators	15
3.2.2.	Beamlines	17
3.2.3.	Modeling of the Accelerator Complex	19
3.2.4.	Technical Systems	20
3.3.	Interfaces of the Settings Management System	20
3.3.1.	Interfaces for the Import of Static Data	21
3.3.2.	Interfaces to Devices	21
3.3.3.	Interfaces to the Timing System	21
3.3.4.	Interfaces to the Interlock System	21
3.3.5.	Interfaces to Services or Applications	22
3.3.6.	Interfaces to Other Systems	22
3.3.7.	Interfaces to the Logging System	22
4.	Quality Assurance, Tests and Acceptance	24
4.1.	Development Methodology	24
4.2.	Quality Assurance System of the Supplier	24
4.3.	FAT	24
4.4.	SAT	24
5.	Documentation	24
6.	Warranty	24
7.	Scope of Delivery	24
I.	Attached Documents	25
II.	Related Documentation	25
III.	Document Information	25
III.1.	Document History	25

List of Tables

Table 1: List of Functional Requirements.....	9
Table 2: List of Non-functional Requirements	11

List of Figures

Figure 1: Settings Management System Overview	6
---	---

1. Purpose and Classification of the Document

The purpose of this document is to specify the Accelerator Control System component "Settings Management System" for FAIR (PSP codes 2.14.10.1.1 and 2.14.10.2.1).

This document is the most detailed type of document in the hierarchy of Control System specifications.

Whenever regulations and requirements are specified in the General Specifications, Technical Guidelines or Common Specifications of the Control System they are only referenced in this document. The related documents are listed in Appendix II.

No legal or contractual conditions are treated in this document. All related information is given in the General Specifications for FAIR II.

1.1. Responsibilities

The responsibilities with respect to changes and modifications of the present document are entirely in the hands of the Controls Department of the GSI Helmholtz Centre for Heavy Ion Research GmbH (GSI) Darmstadt.

For initial information please contact the administration of the Controls Department.

Further information on the organigram, names of responsible persons and task leaders, as well as the agreed document release and approval procedure is summarized in the organizational note 'Controls Project for FAIR'.

1.2. Classifications of Requirements

The following definitions of requirement classifications are being used throughout the document:

- **"Must"** or **"shall"** or **"is required to"** is used to indicate mandatory requirements, strictly to be followed in order to conform to the standard and from which no deviation is permitted.
- **"Must not"** or **"shall not"** mean that the definition is an absolute prohibition of the specification.
- **"Should"** or **"is recommended"** is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others or that a certain course of action is preferred but not required.
- **"Should not"** or **"is not recommended"** mean that there may exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications should be understood and the case carefully weighted before implementing any behavior described with this label.
- **"May"**, which is equivalent to **"is permitted"**, is used to indicate a course of action permissible within the limits of the standard.

2. Scope of the Technical System

2.1. System Overview

A central part of the control system on the middle layer is the component for generating and managing settings for the whole accelerator complex. The main task of this component is the generation of consistent data for the synchronized operation of all devices and machines.

The user interfaces together with the application logic resides above the settings management system in the application layer. Applications that make use of the settings management system and their application logic are specified in the detailed specification for user applications [2].

To fulfill its task, the settings management system must contain a machine model representing individual machines together with their devices as abstractions characterized by a set of high-level physics parameters describing the possible states and actions of the machine. The model must provide the mechanisms to calculate the set values for all devices from the high-level physics parameters. Moreover, the model must contain additional intermediate parameters allowing to access quantities like magnet strengths, fields, etc. Such quantities are required e.g. for analysis of beam diagnostics data. The model must contain an explicit representation for the dependencies between all parameters.

Furthermore, the settings management system must also contain a model representing the interdependencies of and the interactions between the various accelerators of the whole facility. This model must allow the description of beam production chains (i.e. the sequence of accelerators and actions required to deliver beam from source to target) and the combination of several such chains into an execution schedule for parallel operation.

In addition to this modeling part, the settings management system must provide functionality for reading and writing data from and to devices by using corresponding communication mechanisms supplied by the control system. In particular, the system must send the calculated set values to the devices. To ensure offline access and provide a history, the calculated data must be persisted. The settings management system must also provide interfaces to clients for interacting with the system.

The settings management system is logically split into two aspects: a *technical framework* implementing the control structures on the one hand and the *physics model* of the accelerators on the other hand. This separation is necessary because the realization of the physics model requires input from accelerator physicists who should not be concerned with the technical details of the system. Therefore, the framework must provide the possibility for physicists to model accelerators using information about accelerator layout and ion optical design, to define parameters and their dependencies and to supply the corresponding algorithms for the calculation of values. Moreover, the framework must include the communication with the devices and the presentation of all contained information towards client applications.

This document reflects these two aspects in corresponding chapters.
For a system overview, see Figure 1.

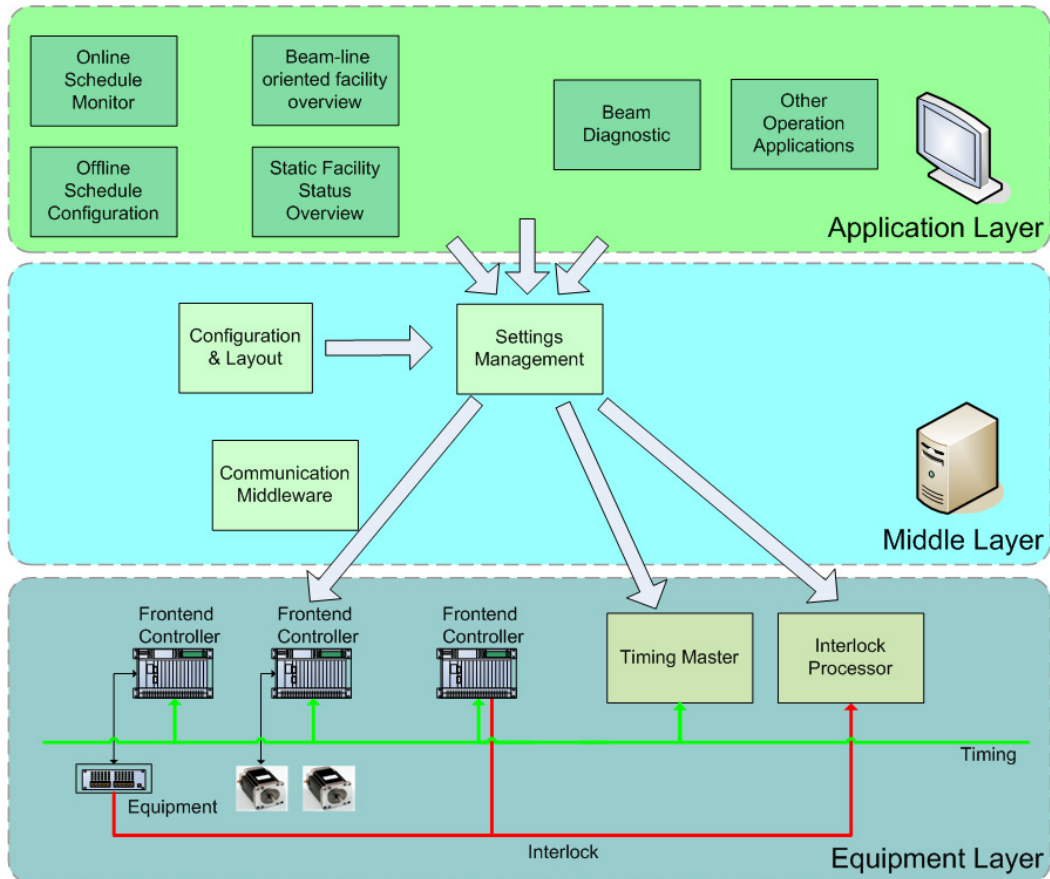


Figure 1: Settings Management System Overview

2.2. Limits of the System and Environment

2.2.1. Limits

The settings management system is no “online” system, i.e. it manages settings for the accelerators and knows the execution schedule, but it does not have an online reflection of what is currently executed in the machine (this is typically only known by the timing system).

The settings management system provides a generic interface for user applications or other services that make use of its functionality. This interface is mainly aimed at triggering actions and querying data, it does not impose any user work flow.

To ensure the integrity of the data, the settings management system depends on external input. In particular the correctness of configuration data of the accelerator facility (machine layout, device limits, calibration curves, ...) cannot be verified within the system itself.

Data supply with set values for devices that take part in the settings management must either be done through the settings management system or, if it has to be done in another way, the system must be informed about these changes.

Security aspects cannot be handled by the settings management system alone, but need to be enforced on different levels of the control system.

2.2.2. Interfaces

As a central part of the control system, the settings management system interacts with many other components of the control system. The following paragraphs shortly mention those interfaces to get a full picture of how the settings management system is embedded into the control system.

Data from devices and possibly also other services is read and set using middleware communication systems.

The settings management system represents the basis for user applications, e.g. those specified in [2]. It therefore must provide a clear, simple interface through which applications can easily access relevant data.

The settings management system must integrate with the timing system [3] which needs to be informed about the execution schedule.

The settings management system must integrate with the interlock system [4] which needs to be informed about the devices taking part in the current execution schedule.

Details about the interfaces are described in 3.3.

2.2.3. Environment

No special environment is required or has to be considered since the settings management system runs in the main computing center of the control system.

2.3. Basis of Concept

2.3.1. Functional Requirements

In general, the framework must provide all necessary functionality to model accelerators and manage settings. It must include functionality to put knowledge about optics, layout of the machine, devices and their interfaces into the system. It must provide the possibility to represent a physics model of the machines.

Besides these basic modeling functionalities, the way how the future machines shall be operated mainly influences the settings management system. Some key aspects of future operations are listed here, not to the full extend, but trying to cover those points that mainly influence settings management. They represent the current state of discussion.

Like today, the operation of the machine will in general consist of the three steps definition, initialization, operation: defining what shall be executed in the machine, initializing values (coming from either theory or known good values) and afterwards operating the machine with those values, i.e. adjusting them to current needs. In the future the focus will move even more to getting an overall picture over the whole machine complex, therefore consistent settings need to be generated and managed covering all involved accelerators.

The GSI/FAIR accelerators and beamlines will be set and manipulated as much as possible using high level physics parameters. The generation and management of complex settings must be based on theoretical machine models

Document Title: Detailed Specification Settings Management System

and ion-optical simulation programs. Device specific information like calibration curves must also be taken into account. Operators and machine specialists must be able to apply corrections (trims) to settings to adapt them to the behavior of the real machine. Applied trims shall be distinguishable from and comparable to settings coming from theoretical calculations. The system must provide the functionality to copy and scale existing settings as basis for new settings. Changes of high level parameters shall propagate down to the devices automatically to be able to consistently change values for the whole accelerator. Singular changes of e.g. a value at injection time must propagate correctly e.g. into the ramp of a bending magnet if necessary.

The synchrotrons will be operated using almost arbitrarily pluggable cycles as well as predefined sequence of cycles (e.g. for the booster mode). Changing of settings can lead to longer or shorter cycles. Changes to the schedule or to settings happen on a daily basis and must therefore be a very flexible procedure. This procedure must update the information in the timing system to adapt the execution schedule.

Changes to settings in one accelerator might also affect neighboring accelerators, therefore changes need to be propagated automatically along the accelerator chain. How accelerators are linked together and depend on each other's settings, must be part of the accelerator model. Applications based on that data are necessary that allow editing cycles in the accelerator chain and assist the user in finding an optimal sequence. For those applications see [2].

Settings that worked well in the past must be usable as a basis for today's operation. To fulfill that requirement, settings must be managed using a consistent data storage, keeping settings and their history. To be able to find good historical settings, mechanisms must be put in place to mark them for easy retrieval. The framework must provide the functionality to consistently store data. For the underlying data storage see also the detailed specification for data management [5].

Variants or families of settings are needed (so-called multi parameter beams): on the one hand to support a larger flexibility of settings and on the other hand to limit the effort to maintain them, it is necessary to keep certain settings as variants. Certain parts of the settings are common, if they are manipulated, their value shall change in all variants, and other parts of the settings are kept and are changeable per variant. E.g. for focus variation the settings for the last two quadrupoles differ between variants, all other settings are identical. The system must allow the user to define a set of parameters and corresponding ranges to parameterize these variants.

In addition to these user defined variants, the system should have the flexibility to parameterize initial settings (theory values) according to certain criteria defined within the model, e.g. variants for light or heavy ions.

When generating initial settings (theory values) it is necessary to follow not only strict implemented physical rules, but also to be able to define certain variations of generated settings, e.g. according to beam characteristics (light or heavy ions, etc.). The system should have the flexibility to be able to define these different dependencies on various 'parameters'.

Document Title: Detailed Specification Settings Management System

Various different beam modes (like commissioning / productive beam, low / high intensity beam) and modes of the machine (like shutdown, production) must be representable and lead to defined actions, e.g. with low intensity beam more manipulations should be possible than with high intensity beam.

Since the settings management system is the component which calculates what exactly devices shall do, it must also have the knowledge about power consumption and heat load in cryogenic systems as a function of time. This information shall be used while setting up an execution schedule to determine if those quantities will stay in certain predefined ranges.

Since energy costs are a growing concern, the system must provide the information necessary to put certain parts of the accelerator into 'stand-by'.

The settings management system fulfills the following functional requirements:

Number	Description of the Requirement
SM_010	Consistent setup of a beam along the accelerator chain must be provided.
SM_020	Consistent setup of a full execution schedule including several beams must be provided.
SM_030	The system must provide manipulating settings on high level physics parameters, including automatic propagation down to hardware parameters.
SM_040	Changes in the overall execution schedule including the flexibility of cycle lengths depending on trimmed values must be supported including an update of the timing system information.
SM_050	The functionality must provide applying historical settings including an easy retrieval of "good" settings.
SM_060	The system must provide offline calculation of settings, to support preparation operations in advance, but also to be able to calculate settings for beams that will never be actually executed in the machine.
SM_070	The system must support variants / families of settings (multi parameter beams).
SM_080	Functionality must be provided to put knowledge into the system about: optics, machine layout, all devices of the control system.
SM_090	Functionality must be provided to represent physics models, including parameter hierarchies (from physics to hardware parameters) and propagation rules.
SM_100	Functionality must be provided to create a consistent model of the accelerator chain with its dependencies.
SM_110	Knowledge about power consumption and heat load in cryogenic systems must be contained and must be used in validity checks.
SM_120	The settings management system should be able to support an energy saving scheme.
SM_130	Parameterization of the initial theory values should be possible.
SM_140	The settings management system must support commissioning of one part of the machine including necessary changes to the system while other parts of the machine are in production mode.

Table 1: List of Functional Requirements

2.3.2. Non-functional Requirements

Operational procedures executed by the operator should not be impaired by the performance of the control system. This requirement on the whole control system implies performance requirements on the different subsystems. For operational procedures involving the settings management system (e.g. setup of execution schedule or trims) this means that the performance requirements cannot be satisfied by the settings management system alone (e.g. performance depends here also on communication times with devices and their implementation).

One performance requirement is that the changes to settings can be performed within a certain time limit, starting from the operators click on some “apply” button to the response of the system. These performance requirements on the whole control system are listed below.

Another performance requirement not related to changes of settings is coming from planning applications. They need to quickly create execution schedules including e.g. drag and drop of beams into a given schedule. The settings management system must therefore provide a quick calculation of the timing information for beams and execution schedules. To keep such applications responsive, the GUI guidelines demand a response time of 100ms.

Since the settings management system is responsible for supplying devices with data it is one of the essential control system components for accelerator operation. A reasonable effort should be made to maximize the availability of the system. This is also a requirement on other components the system relies on.

The settings management system must be able to serve a number of clients in parallel. It is assumed to have about twenty client applications per console and a number of about fifty logical consoles.

The settings management system fulfills the following non-functional requirements:

Number	Description of the Requirement
Performance	
SM_400	Simple changes affecting only one device must not take longer than a second.
SM_410	Changes affecting a limited number of devices within a single accelerator should be executed within one cycle.
SM_420	Changing settings for one accelerator must not take longer than ten seconds.
SM_430	Setup for one beam production chain or a complete execution schedule must not take longer than thirty seconds.
SM_440	Calculation of the timing information for the planning of an execution schedule must not take longer than 100ms.
Availability	
SM_450	The settings management system must be always available, except for maintenance.
Scalability	
SM_500	The settings management system must scale and finally support up to 1000 parallel clients.

Extensibility	
SM_550	The settings management system must be easily extensible through separation of concerns, e.g. by separating the settings management framework from the machine models.

Table 2: List of Non-functional Requirements

2.3.3. General Constraints

Decision was taken in 2009 that main parts of the settings management system will be realized using a software product from CERN called LSA (**LHC Software Architecture**). It is highly database driven and has one data model for all accelerators. Since LSA is a modular framework that is easily adaptable, it will be the core component for settings management. For detailed information on the LSA system see [7] and first results at GSI see [8].

For covering the needs of a full beam- and settings management system, additional functionality is needed, which is not or not yet covered by the LSA framework as it is now (spring 2012). This has to be taken into account for the design and implementation phases. These aspects are especially focused upon in chapter 3.

2.3.4. Architectural Principles

The software architecture guideline for the control system [6] generally applies.

However, since as described in 2.3.3 General Constraints the LSA system is set as central component for the realization of the settings management system, the architecture inherent to the LSA system has to be taken into account and might restrict certain design choices.

3. Technical Specifications

The specification first concentrates on the generic functionality that the settings management framework must provide, see section 3.1. Then the machine models are described that must be delivered in order to model the FAIR accelerator complex using the framework, see section 3.2. Finally the description focuses on the interfaces towards other components of the control system, see section 3.3.

3.1. Settings Management Framework

A framework must be supplied that provides generic mechanisms for accelerator settings management.

Data in the settings management framework can be separated into different types of data: configuration data; data that is necessary to calculate settings (e.g. the model); and data that is generated and persisted or sent to the devices.

3.1.1. Data import

The calculations of the settings management system are based on data from other systems. Therefore data must be imported about the layout of the machine, the devices and their attributes (e.g. data types, properties, calibration curves, min/max values, ranges, units, preparation times, ...), ion optical information, and available communication interfaces.

To ensure extensibility this data must be kept in generic structures that allow adding new accelerator information in an easy way.

The consistency of the imported data must be ensured. This can only be achieved by pushing newly available configuration information into the settings management system. The source system from which the data is imported must ensure the correctness of the data.

The data import process should be automated wherever possible.

3.1.2. Accelerator model

The framework must provide the functionality to model accelerators and also dependencies between them. This includes modeling parameter hierarchies together with propagation rules for calculating values along the hierarchy from physics to hardware parameters.

Because the FAIR machine complex consists of an integrated chain of accelerators that depend on each other, the framework must provide the functionality to define rules also for propagating value changes across accelerator boundaries (e.g. propagation of energy changes to neighboring accelerators).

The accelerator model must be very well separated from the framework logic in order to allow physicists to contribute their knowledge in a well-defined work environment. For detailed information on the machine model see also section 3.2.

Document Title: Detailed Specification Settings Management System

Based on the static data imported into the system and the machine model the framework must allow representing processes that take place in single accelerators (like injection, ramp, extraction) and beams travelling through the machine complex (from source to experiment).

3.1.3. Calculation of Data

The framework must allow creating an execution schedule (consisting of several execution patterns) for the whole machine complex.

The system must support generating a set of theory values that can be used as a base for the operation of the machine. Changes to these settings for fine tuning during operation must be clearly distinguished from the theory values. It is also necessary to keep track of these changes to make every step reproducible and revertible.

A “setting” in this sense is a value connected to a parameter of the hierarchy and a process taking place in the accelerator. Settings must be typed and data types like scalar, array, array2d, function type (representation of mathematical functions, e.g. current over time), must be supported by the framework.

The system must allow calculations independent of whether the device data is actually sent to the devices or not. This is necessary for offline preparation of settings and analysis purposes.

During calculation several checks are performed by the framework. A value change could lead to a change in the schedule (e.g. if energy is changed), the framework must in this case adapt the execution schedule and calculated values accordingly. If the calculated values violate limits (e.g. min/max values) the system must prevent those values from being sent to the devices.

The framework must provide the functionality to define variants of beam parameters (so-called multi parameter beams) and to calculate set values for a whole range of variants the beam depends upon (e.g. energy variations).

3.1.4. Data persistence

Since the settings management system is an offline system that must keep its data also for offline calculation and analysis, the framework must allow all data to be persisted. Calculated settings must be persisted together with their history independent of whether the device data is actually sent to the devices or not.

The settings management framework therefore will not function without the persistence layer. This imposes requirements on the persistence layer, which is specified in the detailed specification for data management [5].

3.1.5. Data presentation

The settings management framework must provide interfaces to generic applications to allow access to its logic and contained information in an accelerator-independent way, i.e. the framework must internally use only *one* data model for all accelerators.

The interface must be structured to provide easy client access to the system (e.g. one part of the interface provides functionality for creating and modifying beams and patterns, another part offers the functionality for modifying settings).

Document Title: Detailed Specification Settings Management System

Note: The settings management system will be realized using an existing software component as basis (see 2.3.3). Since this system was designed to be platform-independent, it was realized in Java. The main focus therefore lies on providing Java interfaces for Java clients.

3.1.6. Data supply

To ensure consistency, the settings management system must be the master for set value generation.

For performance reasons, the system must be able to send only changed data to the devices. Based on the assumption that no other system does send set values, the data supply in the system must be organized in such a way that only modified data is sent to devices. The system must also allow to resend data to dedicated devices, to be used e.g. after a repair of a device.

Besides data for beamline devices, this data includes schedule information for the timing master which – from the settings management point of view – must be treated as another “device”.

For consistency reasons, the system must ensure transactional data supply; i.e. that set values are only made active in the devices, if all devices acknowledge the data changes or roll back the changes otherwise. This is also a requirement for the mechanisms used for communicating with the devices.

3.2. Machine Models

The FAIR accelerator complex consists of several accelerators connected by a mesh of beamlines. From the functional point of view, each accelerator and each beamline require their own machine model. However, given the large number of accelerators and beamlines, the model part of the settings management system must make use of the commonalities among the various accelerators and beamlines in order to keep the structures compact, efficient, and maintainable.

Thus, the structures describing the accelerators and beamlines as well as the algorithms used to calculate set values must be generic, that is, independent of the particular accelerator or beamline, whenever this is reasonably achievable. Of course, any accelerator or beamline might have their peculiarities which will have to be modeled separately. Supporting generic and specific modeling in an efficient way is one of the fundamental requirements on the modeling capabilities of the settings management framework.

Concerning the machine models themselves, a model for an accelerator or beamline is considered generic if the following criteria are met:

- The operational interface to the model must be described by a unique set of high-level physics parameters independent of the accelerator or beamline. Such parameters comprise, among others, beam parameters, parameters describing the ion optical properties of the system, parameters for certain operational modes.
- The model must contain the same set of intermediate parameters, like magnet strengths, fields for each accelerator or beamline.

- The structure of the dependencies between the parameters (i.e. the parameter hierarchy) must be independent of the accelerator or beamline, unless specific properties of a particular accelerator or beamline make a deviation from this rule unavoidable. This does, of course, not apply to parts of the parameter hierarchy which are unique to a particular accelerator or beamline because they are used to model a specific feature.
- The manipulations of the beam in an accelerator or beamline during a cycle or supercycle must be described by structures which are independent of the particular accelerator or beamline.
- The algorithms for calculating the values of all parameters in the parameter hierarchy of a particular accelerator or beamline must be applicable to all structurally identical parts of the parameter hierarchies for different accelerators or beamlines. Of course, the previous requirements serve exactly the purpose of letting the algorithms choose the appropriate calculations from the generic or common structures describing parameter hierarchies and beam manipulations.
- The data structures used to represent the values for the parameters of accelerators and beamlines must be efficient enough to comply with the performance requirements on the settings management system. In particular, the data structures representing parameter values as a function of time must use sparse or compressed representations if this is necessary to keep the time to persist the data small.

In addition to the models for individual accelerators and beamlines, the model part of the settings management system must also contain a model of the complete accelerator facility. This model must provide representations for the complex execution patterns during parallel operation, describing the properties of the different beams from source to target, the timing information of the pattern, and the beam transfers between accelerators or to experiments. Furthermore, it must take into account changes to the schedule through modifications of parameters in individual accelerators (e.g. ramping speed), and it must provide alternative schedules for beam abort schemes.

3.2.1. Circular Accelerators

The FAIR accelerator complex comprises between five and ten circular accelerators, i.e. synchrotrons or storage rings, where the exact number depends on the construction stage. While these accelerators each serve their own purpose in the operation of FAIR, the sequences of beam manipulations executed by these machines are chosen from a common set of basic manipulations, like injection, acceleration or deceleration, extraction, changes of the bunching structure, stacking, and cooling. It is essential that the settings management system contains a generic model for circular accelerators allowing the calculation of values to realize all those beam manipulations. Thus, if at least two circular accelerators employ a particular beam manipulation, the description of this beam manipulation, the set of parameters used to quantify it, and the algorithm used for calculating the values of all dependent parameters must be applicable for both machines.

Document Title: Detailed Specification Settings Management System

This general requirement notwithstanding, the generic model for circular accelerators must in particular have the following properties:

- Each circular accelerator must have an identical set of high-level physics parameters used to describe and modify the beam properties at injection and extraction, if beam is extracted from the accelerator. These parameters are used to ensure the consistency of the beam properties within an accelerator chain from source to target. They will usually be supplied during the planning phase of a beam pattern. However, they may also be changed later, inducing corresponding changes in neighboring beamlines and accelerators. Specifically, the parameters describing the beam are:
 - Particle identifier (Element number, Isotope number)
 - Charge state
 - Kinetic energy per nucleon
 - Number of particles
 - Time structure

It should be noted that the particles revolving in a circular accelerator are not necessarily stable: On the one hand, they may undergo radioactive decay during the storage time, on the other hand, the particle species may be changed deliberately. Furthermore, it may be possible to store several types of particles at the same time in a circular accelerator. In these cases, the settings of the circular accelerator are still characterized by a single representative particle type.

- Each circular accelerator must have an identical set of high-level physics parameters used to describe and modify the transversal and longitudinal properties of the beam. These include:
 - Transversal tunes
 - Chromaticity
 - Momentum compaction factor
 - Transversal emittance
 - Longitudinal emittance
- Each circular accelerator employing RF manipulations (like bunching or debunching, acceleration or deceleration, bunch merging or compression, RF stacking) must have a suitable set of high-level physics parameters, identical for each particular RF manipulation, to describe and modify the manipulation. These include:
 - Bucket fill factor
 - Bunch pattern
 - Timing parameters (e.g. merging or compression time)
- Each circular accelerator employing cooling (electron or stochastic cooling) must have a suitable set of high-level physics parameters, identical for each particular cooling scheme, to describe and modify the cooling scheme.
- Each circular accelerator must have a suitable set of parameters for correction of the closed orbit. However, the realization of an orbit correction scheme is outside the scope of the model. (This must e.g. be treated by a dedicated application that also measures the beam position).

Document Title: Detailed Specification Settings Management System

- Each circular accelerator from which beam is extracted must have a suitable set of parameters to describe and modify the extraction scheme. In this case, it may or may not be possible to use identical parameters for the same extraction scheme in different accelerators, because some extraction schemes, like slow extraction, may require machine specific calculations. Nevertheless, parameters describing the same physics quantities must be identical in all accelerators.
- Each circular accelerator must have an identical set of parameters describing the discretization of time functions and the timing of changes in values (e.g. rounding times).
- Apart from the input parameters listed above, the model of each circular accelerator must also contain a set of derived physics parameters. These are required for operational and analysis purposes and include:
 - Magnetic and electric rigidity
 - Revolution frequency
 - Synchrotron frequency
 - Bunching factor
 - Transverse space charge tune shift
- In addition to the high-level parameters for operation, the model of each circular accelerator must also contain a set of derived parameters related to the properties of independent devices. These include:
 - Strengths of magnets and electrostatic deflectors
 - Fields of magnets and electrostatic deflectors
 - Total voltage of RF cavities per harmonic, if present
- For each circular accelerator the same set of identifiers must be used to characterize the beam manipulations (e.g. Injection, Ramp, Extraction, Merging).
- The model must allow the application of changes to time functions by modifying scalar input values. For instance, it must be possible to change the transversal tunes during extraction by entering the new values as scalars. The model must then incorporate these changes continuously into the new time function for the tunes.
- The model must contain generic algorithms for calculating values for all generic beam manipulations, as well as specific algorithms for all specific features of certain accelerators.
- All algorithms for calculating the values of circular accelerators must take into account space charge corrections, unless it can be proven that such corrections are negligible under all operational conditions.

3.2.2. Beamlines

The FAIR accelerator complex comprises an abundance of beamlines connecting sources, accelerators, and experiments. While the beamlines are large in number and the combinatorial complexity of all possible connections is in itself challenging for the model of the complete facility (see below), the model for an individual beamline is relatively simple compared to the model for circular

Document Title: Detailed Specification Settings Management System

accelerators. In particular, a beamline usually serves only to transport the beam and does not perform beam manipulations, with the notable exception of devices like strippers, targets, choppers and bunchers which change the beam properties.

Apart from the parameters for the description of the beam properties, the model for beamlines consists mainly of a model for independent magnets or electrostatic deflectors (or, more generally, for independent groups of such devices), each equipped with their own set of parameters, and a model for devices changing the beam properties. Thus, the generic model for beamlines must have the following properties:

- Each beamline section free of devices changing the beam properties must have an identical set of high-level physics parameters used to describe and modify the beam properties during transport of the beam through this section. These parameters are used to ensure the consistency of the beam properties within an accelerator chain from source to target. They will usually be supplied during the planning phase of a beam pattern. However, they may also be changed later, inducing corresponding changes in neighboring beamlines and accelerators. Specifically, the parameters describing the beam are:
 - Particle identifier (Element number, Isotope number)
 - Charge state
 - Kinetic energy per nucleon
 - Particle number
 - Time structure

It should be noted that for beamlines behind strippers or targets the beam contains actually a mixture of different particle types. In this case, the settings of the beamline are still characterized by a single representative particle type. The model is not required to contain information about all possible particle types that will be produced.

- For each stripper or target, the model must contain calculations of the energy loss and the equilibrium charge state. The model is not required to contain information about all possible secondary particles.
- For each chopper and buncher, the model must take into account the change of the time structure of the beam.
- For each beamline every independent group of magnets or electrostatic deflectors must be described by a strength parameter at the highest level of the hierarchy.
- Even in beamlines, groups of independent devices are sometimes used to perform coordinated activities (e.g. a chopper chicane forming a closed bump using two steering magnets and an electrostatic deflector). In such cases the coordinated activity must be modeled by introducing a high-level parameter for controlling the independent devices simultaneously.
- In addition to the high-level parameters for operation, the model of each beamline must also contain a set of derived parameters related to the properties of the independent devices. These include:
 - Fields of magnets and electrostatic deflectors

Document Title: Detailed Specification Settings Management System

- For each beamline the same set of identifiers must be used to characterize the different beam transport modes.
- The model must contain generic algorithms for calculating values for all generic beam transport modes, as well as specific algorithms for all specific features of certain beamlines.
- The model must support the calculation of time dependent changes of the set values of beamline devices in accordance with the scheduled execution pattern during parallel operation. In particular, constraints of the devices on the time required to change their set value must be taken into account.

3.2.3. Modeling of the Accelerator Complex

One of the main challenges for the settings management system arises from the complexity of the planned parallel operation schemes of the FAIR facility. These operation schemes necessitate, for instance, the periodic operation of accelerators or beamlines with different settings for the transport of beam to several experiments. For any given experiment, the beam passes through a chain of accelerators and beamlines on its way from source to target. Therefore, the chain of accelerators and beamlines is not only a sequence in space, but also in time. If several such chains are to be operated at the same time, a complex pattern arises which must satisfy time and space constraints. The model of the accelerator complex must provide a suitable representation of such patterns.

In particular, the model must support the following concepts:

- The model must contain a representation of the physical chain of accelerators and beamlines required to deliver the beam from the source to the target.
- The model must contain a description of the beam properties along the chain of accelerators and beamlines. The relevant beam properties are:
 - Particle identifier (Element number, Isotope number)
 - Charge state
 - Kinetic energy per nucleon
 - Number of particles
 - Time structure
- Typically, the beam properties listed above do not remain fixed along the chain of accelerators and beamlines. Rather, they are changed either by insertion devices (strippers, targets) in beamlines or by accelerators. The model must contain a description of the changes of the beam properties along the chain of accelerators and beamlines.
- The model must contain a description of the transfer of the beam between sources, accelerators, and experiments. Based on this description, the model must supply data to the timing system to activate synchronization mechanisms if necessary.
- The model must provide a representation for the sequence of beam manipulations and beam transport processes along the chain of accelerators and beamlines. That is, this representation must allow

Document Title: Detailed Specification Settings Management System

deducing the sequence of events necessary to produce the beam for a particular experiment.

- The model must contain a representation for the execution pattern of all chains of beam production to be executed in parallel. Based on this representation, the model must supply data to the timing system for the execution of the pattern.
- The description of an execution pattern must allow the definition of alternative execution sequences in particular chains to support beam abort schemes. For instance, a target accelerator may not be ready to accept beam at the scheduled time of the transfer due to an interlock. In this case, the settings management system must provide an alternative execution branch to the timing system and corresponding settings to the devices that will safely dispose the beam.
- The scheduled execution pattern may change if certain parameters are modified during operation. Such parameters include the beam properties (e.g. the extraction energy), but can also be parameters of individual accelerators (e.g. the ramping speed in a synchrotron). The model must support the adaptation of a given pattern due to such changes. Moreover, the model must distinguish between changes that do not affect the timing of the pattern and changes that do.

3.2.4. Technical Systems

Since the settings management system contains the complete information about the presently scheduled execution pattern, it can be used to calculate data concerning the power consumption when executing the pattern. In particular, data for the instantaneous pulse power taken from the electrical network and for the heat transfer to cryostats of superconducting machine parts can be predicted. In the case of the electrical power, there are restrictions on the power spectrum of the instantaneous pulse power which must be used to accept or reject a given execution schedule. Therefore, the model must contain parameters and algorithms to compute this power spectrum. Moreover, the restrictions on the power spectrum must be accessible for comparison within the settings management system.

In the case of the heat transfer to cryostats, the average heat load must be supplied to the cryogenic system. Therefore, the model must contain parameters and algorithms to compute this average heat load.

3.3. Interfaces of the Settings Management System

As the central component for settings management, the settings management system must have interfaces to many components of the control system, which are described in this chapter.

In the context of the FAIR and GSI facility it must also be possible to integrate existing legacy components. Interfaces for this integration might be necessary that are not covered here. Therefore it must be possible to add interfaces to the settings management system at a later stage.

3.3.1. Interfaces for the Import of Static Data

The settings management system needs data that is contained in other systems. As mentioned above, this data must be imported into the settings management system. Specific interfaces for each of the source systems need to be provided, automatic import mechanisms should be foreseen wherever possible.

3.3.2. Interfaces to Devices

To communicate with devices, these devices must be known beforehand to the settings management system (this must be part of the import of relevant data into the system, see section 3.1.1). Changes to machine layout or devices that have an impact on the settings management must be communicated to the settings management system. This process should be automated wherever possible.

The calculated set values must be communicated to the devices using mechanisms provided by the front-end controller software [10]. No matter which technology devices employ, a unified interface must be provided that must be used by the settings management system. Data exchanged with the devices must be self-describing to achieve a separation of concerns and to relieve other components from knowing too many details about the devices.

As mentioned above (see section 3.1.6), the communication with the devices must support transactional behavior.

3.3.3. Interfaces to the Timing System

Planned execution schedules and changes to these schedules need to be communicated to the timing system [3]. The timing system must therefore provide an interface which the settings management system must use. With this interface it must be possible to communicate complete execution schedules or only relevant changes to an existing execution schedule.

The scheduled information must include tables of single events describing single synchronized actions within an accelerator. It must include the description of processes that are executed in the accelerator, possible alternatives and their execution conditions. The information must include description of beam production chains and the description of a combination of several such chains into an execution pattern for parallel operation.

3.3.4. Interfaces to the Interlock System

The interlock system must be informed about devices necessary for beam production, because that information has to be taken into account during the evaluation of interlocks. Since the settings management system manages all beam production chains (i.e. the sequence of accelerators and actions required to deliver beam from source to target) and therefore knows which devices are necessary for a given beam production chain, it must communicate this information to the interlock system, independent of whether set values are generated for a particular device or not.

Based on the information about necessary devices, the settings management system must generate a list of necessary devices per beam production chain to communicate this list to the interlock system [4]. For this purpose, the interlock system must provide an interface which the settings management system must

use. Changes to the setup of beam production chains must be communicated whenever they affect the interlock configuration. With the interface of the interlock system it must be possible to communicate either full interlock configuration for all beam production chains at once or only relevant changes to an existing configuration.

3.3.5. Interfaces to Services and Applications

The settings management system must provide interfaces to applications to allow access to its logic and contained information in an accelerator-independent way. These interfaces should not be application specific but be more general and allow their reuse from many applications. Since the task of the applications, i.e. the presentation layer of the control system, is to provide user access to the control system and represent data and functionality of the underlying components, the interface towards applications must be clear and simple, always combining logically connected functionality in single execution blocks that can be triggered from applications (e.g. the client can send one request for a value change and the system handles within the processing of this one request the calculation and propagation of the values and the sending of new set values to the devices).

In particular, the interface to services and applications must provide the following functionality:

- Definition and modification of beams, including multi-parameter beams
- Definition and modification of execution schedules
- Retrieval of information about defined execution schedules and beams
- Generation of initial settings
- Modification of settings
- Retrieval of actual and historical settings
- Access to static information like machine layout, installed devices and their characteristics, optics

3.3.6. Interfaces to Other Systems

Since the settings management system knows about accelerator dependencies, which settings are generated and sent down to the hardware, which processes are ready for execution, etc., it is in general a necessity that it informs other systems about parts of that information.

Besides applications which typically rather use or “call” the functionality of the settings management system, it must also be possible for the settings management system to push its information to arbitrary interested clients.

3.3.7. Interfaces to the Logging System

The settings management system provides logging mechanisms to be able to log which actions were performed by the system, e.g. which values for which devices were sent to the hardware. For this purpose it must use the operational logging system [9].

Document Title: Detailed Specification Settings Management System

Since the settings management system is a component of the middle layer of the control system, logging of information must be done on the level of the system itself but also on the level of applications using the system. For debugging purposes, the settings management system must additionally log information about calls from the applications (i.e. which application from where called what), which must be done using the diagnostic logging system [9].

4. Quality Assurance, Tests and Acceptance

The system to be built must adhere to the guidelines and recommendations for software developments in the FAIR accelerator control system context, as referenced in the FAIR Common Specification F-CS-C-01e (Common Specification Accelerator Control System). The supplier of the work package must identify the relevant standards and recommendations before start of the development. Details must be fixed as part of the technical design concept in the initialization phase.

4.1. Development Methodology

The settings management system shall be developed in an iterative and incremental methodology.

Each iteration cycle must result in a running system which can be evaluated and tested at FAIR site. The first iteration, which has to be available as early as possible, must concentrate on the most critical functionality. In successive iterations, the system is enhanced by adding features until the desired total functionality is reached.

In the initialization phase, the technical design concept and the plan for the iterations must be developed, and must be approved by the FAIR contracting body. At end of each iteration cycle the achieved status of the system will be evaluated and the iteration plan will be adjusted. Each iteration cycle must be approved by the FAIR contracting body before it can be started.

4.2. Quality Assurance System of the Supplier

Generally, the software specific measures of Quality Assurance described in Common Specification "Accelerator Control System" [1] fully applies.

4.3. FAT

The Common Specification "Accelerator Control System" [1] fully applies.

4.4. SAT

The Common Specification "Accelerator Control System" [1] fully applies.

5. Documentation

The Common Specification "Accelerator Control System" [1] fully applies.

6. Warranty

The conditions and warranty period specified in the Contract applies.

7. Scope of Delivery

All components of the settings management system being subject of this Detailed Specification or resulting from "Common Specifications Accelerator Control System" [1] are in the scope of delivery.

I. Attached Documents

List of abbreviations for controls (Abbreviations_Controls.pdf).

II. Related Documentation

- [1] F-CS-C-01e, FAIR Common Specification "Accelerator Control System"
- [2] F-DS-C-04e, FAIR Detailed Specification "User applications"
- [3] F-DS-C-05e, FAIR Detailed Specification "General machine timing system"
- [4] F-DS-C-08e, FAIR Detailed Specification "Interlock system"
- [5] F-DS-C-24e, FAIR Detailed Specification "Data management"
- [6] F-DG-C-03e, "Software Architecture Guideline"
- [7] LHC SOFTWARE ARCHITECTURE [LSA], G. Kruk et al, Proceedings of ICALEPCS 2003, Knoxville, Tennessee, USA, Paper code: WOPA03
- [8] GSI Scientific Report 2009, Article 33, Page 123, Online Version:
<http://www.gsi.de/informationen/wti/library/scientificreport2009/PAPERS/FAIR-ACCELERATORS-33.pdf>
- [9] F-DS-C-10e, FAIR Detailed Specification "Diagnostic Logging System"
- [10] F-DS-C-01e, FAIR Detailed Specification "FEC software framework"

III. Document Information

III.1. Document History

Version	Date	Description	Author	Review / Approval
0.1	01. Dec. 2009	Draft version, old document structure	Fitzek	
0.2	15. Jan. 2012	Draft version, changed document structure	Fitzek	
0.3	29. Feb. 2012	Draft version	Fitzek, Müller, Ondreka	
1.0	08. Mar. 2012	Final version	Fitzek, Müller, Ondreka	CCT
3.0	16. Aug. 2012	Incorporated FAIR review comments	CCT	