LLRF SYSTEM REQUIREMENT ENGINEERING FOR THE EUROPEAN XFEL

S.N. Simrock#, G. Ayvazyan, Z. Geng, M. Grecki, DESY, Hamburg, Germany
B.Aminov, CRE, Wuppertal

Abstract

The LLRF system of the European XFEL must fulfill the requirements of various stakeholders: Photon beam users, accelerator operators, rf experts, controls system, beam diagnostics and many others. Besides stabilizing the accelerating fields the system must be easy to operate, to maintain, and to upgrade. Furthermore it must guarantee high availability and it must be well understood. The development, construction, commissioning and operation with an international team requires excellent documentation of the requirements, designs and acceptance test. For the RF control system of the XFEL the new system modelling language SySML has been chosen to facilitate the system engineering and to document the system. SysML uses 9 diagram types to describe the structure and behavior of the system. The hierarchy of the diagrams allows individual task managers to develop detailed subsystem descriptions in a consistent framework.

We present the description of functional and non-functional requirements, the system design and the test cases.

CONCEPT OF SYSTEM ENGINEERING

Systems Engineering signifies both an approach and, more recently, as a discipline in engineering. The aim of education in Systems Engineering is to simply formalize the approach and in doing so, identify new methods and research opportunities similar to the way it occurs in other fields of engineering. As an approach, Systems Engineering is holistic and interdisciplinary in flavor.

Systems engineering is a robust approach to the design, creation, and operation of systems. In simple terms, the approach consists of identification and quantification of system goals, creation of alternative system design concepts, performance of design trades, selection and implementation of the best design, verification that the design is properly built and integrated, and post-implementation assessment of how well the system meets (or met) the goals.

- Understand the whole problem before you try to solve it.
- Translate the problem into measurable requirements
- Examine all feasible alternatives before selecting a solution.
- Make sure you consider the total system life cycle. The birth to death concept extends to maintenance, replacement and decommission. If these are not considered in the other tasks, major life cycle costs can be ignored.
- Make sure to test the total system before delivering it.
- Document everything.

#stefan.simrock@desy.de

The seven-task process defined above is an excellent representation of systems engineering as is presently practiced and should serve to avoid most of the problems that have plagued the development of large, complex systems in the past.

SYSML LANGUAGE

The Systems Modeling Language (SysML), is a Domain-Specific Modeling language for systems engineering. It supports the specification, analysis, design, verification and validation of a broad range of systems and systems-of-systems. SysML was originally developed by an open source specification project, and includes an open source license for distribution and use. SysML is defined as an extension of a subset of the Unified Modeling Language (UML) using UML's profile mechanism. SysML uses seven of UML 2.0's thirteen diagrams, and adds two diagrams (requirements and parametric diagrams) for a total of nine diagram types. SysML also supports allocation tables, a tabular format that can be dynamically derived from SysML allocation relationships.

Table 2: Diagram Types

SysML Diagram	Purpose
Activity	Show system behavior as control and data flows. Useful for functional analysis.
Block Definition diagram	Show system structure as components along with their properties, operations and relationships. Useful for system analysis and design.
Internal Block diagram	Show the internal structures of components, including their parts and connectors. Useful for system analysis and design.
Parametric diagram	Show parametric constraints between structural elements. Useful for performance and quantitative analysis.
Requirement diagram	Show system requirements and their relationships with other elements. Useful for requirements engineering.
Sequence diagram	Show system behavior as interactions between system components. Useful for system analysis and design.
State Machine diagram	Show system behavior as sequences of states that a component or interaction experience in response to events. Useful for system design and simulation/code generation.
Use case diagram	Show system functional requirements as transactions that are meaningful to system users. Useful for specifying functional requirements. (Note potential overlap with Requirement diagrams.)

LLRF SYSTEM

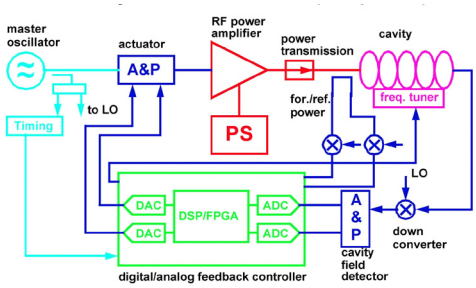

Figure 1: LLRF System Architecture

The architecture of the RF system for the European XFEL is shown in Figure 1. A power amplifier provides the rf power necessary for establishing the accelerating fields in the cavities. The cavity field is measured and the compared to a set-point. The resulting error signal is amplified and filtered and drives a controller for the incident wave to the cavity. A frequency and phase reference system provides the necessary rf signals.
Frequency tuner (motor and piezo) are used for slow and fast resonance control. The timing system provides triggers for pulsed operation and clocks for data acquisition.

LLRF DIAGRAMS

Several types of diagrams are used to describe the LLRF system:

- Requirement diagrams
- Structural diagrams
- Behavioral digrams
- Parametric diagrams

In the following we describe some examples of diagrams which represent the view of the llrf project team. A view is a representation of a whole system from the perspective of a single viewpoint.

Context Diagram

A System Context Diagram (SCD) is the highest level view of a system, similar to Block Diagram, showing a (normally software-based) system as a whole and its inputs and outputs from/to external factors. The best System Context Diagrams are used to display how system inter operates at a very high level or how systems operate and interact logically. The system context diagram is a necessary tool in developing a baseline interaction between systems and actors; actors and system or systems and systems.

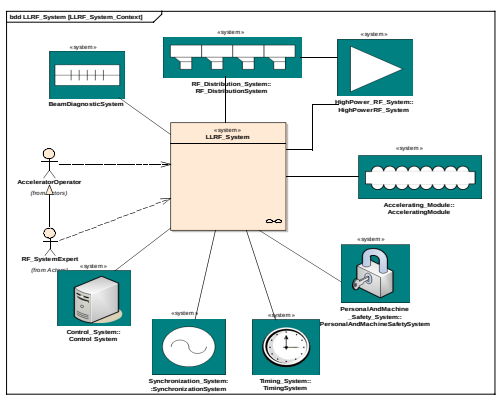

Figure 2: System context diagram

LLRF Use Cases

The use case technique is used in software and systems engineering to capture the functional requirements of a system.

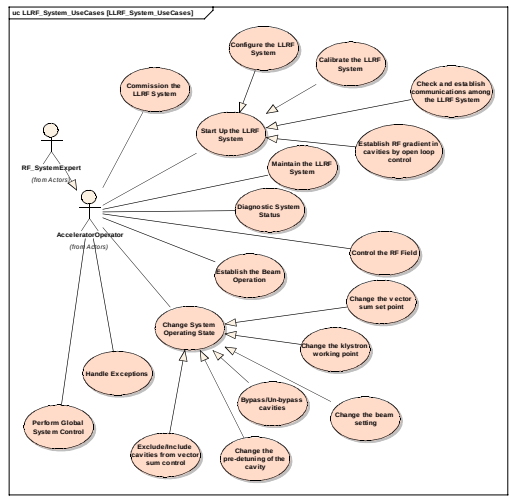

Figure 3: Use cases for the LLRF system

Use cases describe the interaction between a primary Actor (the initiator of the interaction) and the system itself, represented as a sequence of simple steps. Actors are something or someone which exist outside the system under study, and that take part in a sequence of activities in a dialogue with the system to achieve some goal. They may be end users, other systems, or hardware devices. Each use case is a complete series of events, described from the point of view of the Actor.

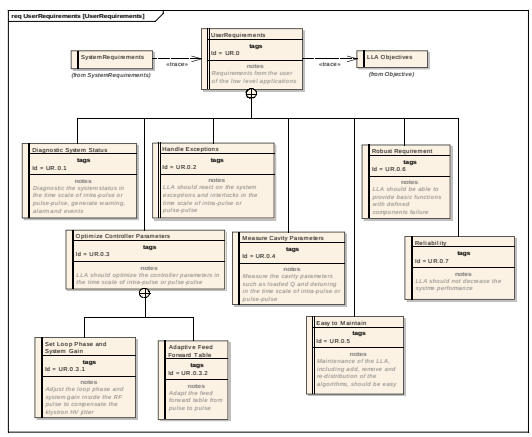

Figure 4: Requirements for an LLRF Subsystem

Requirements

In engineering, a requirement is a singular documented need of what a particular product or service should be or do. It is most commonly used in a formal sense in systems

engineering or software engineering. It is a statement that identifies a necessary attribute, capability, characteristic, or quality of a system in order for it to have value and utility to a user.

In the classical engineering approach, sets of requirements are used as inputs into the design stages of product development. Requirements show what elements and functions are necessary for the particular project.

The requirements development phase may have been preceded by a feasibility study, or a conceptual analysis phase of the project. The requirements phase may be broken down into requirements elicitation (gathering the requirements from stakeholders), analysis (checking for consistency and completeness), specification (documenting the requirements) and verification (making sure the specified requirements are correct)

Block Definition Diagram

Block diagram is a diagram of a system, in which the principal parts or functions are represented by blocks connected by lines, that show the relationships of the blocks. They are heavily used in the engineering world in hardware design, software design, and process flow diagrams.

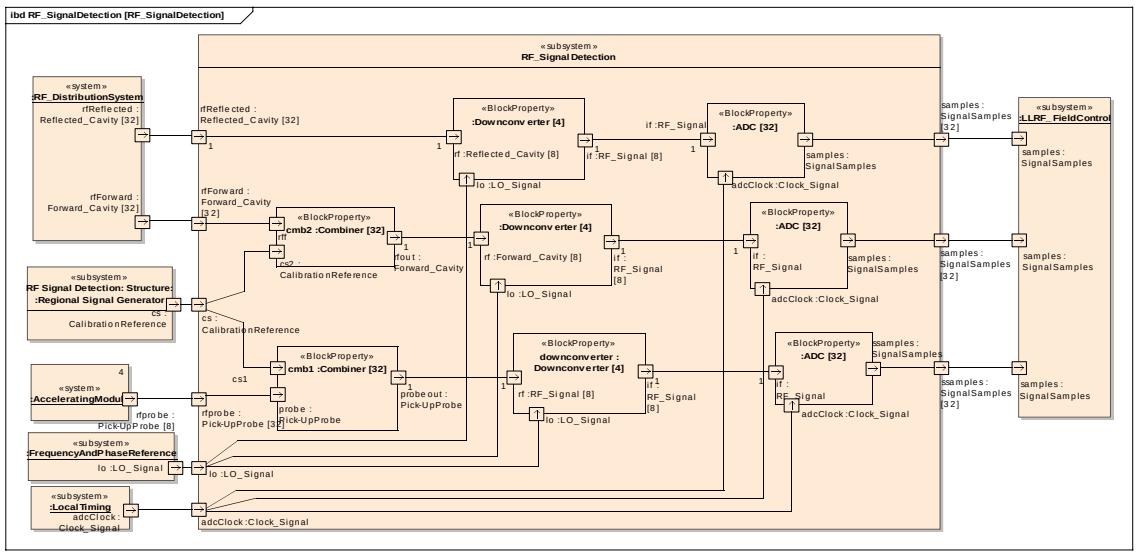

Figure 5: Example for an internal block diagram showing the interfaces and item flow between the LLRF subsystems.

Activity Diagram

An activity diagram is a diagram that shows activities and actions to describe workflows. In the Unified Modeling Language an activity diagram represents the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

In SysML the activity diagram has been extended to indicate flows among steps that convey physical element or energy. Additional changes allow the diagram to better support continuous behaviors and continuous data flows.

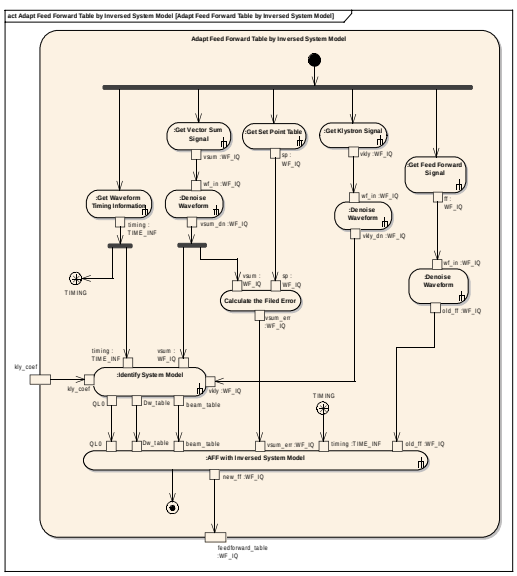

Figure 6: Example for an activity diagram.

SUMMARY

Complex systems such as the LLRF control for the European XFEL require work-processes and tools to guarantee a successful outcome. The attempt to combine of modern system engineering methodologies with the modeling language SysML and the SysML Modeling tool Enterprise Architect (EA) has proven to be very promising in a large international collaboration between research labs, universities and industry.

REFERENCES

[1] (1995) NASA Systems Engineering Handbook. NASA. SP-610S.

[2] Weilkiens, T. , “Systems Engineering with SysML/UML”, 1st edition, 2008, Morgan Kaufmann Publishers Inc., ISBN 0123742749