Mem. S.A.It. Vol. 88, 141
© SAIt 2017

Memorie della

System Modeling of a large FPGA project: the SKA Tile Processing Module

C. Belli and G. Comoretto

Istituto Nazionale di Astrofosica – Osservatorio Astrofisico di Arcetri, Largo Enrico Fermi 5, I-50125 Firenze, Italy, e-mail: belli@arcetri.astro.it

Abstract. Large projects like the SKA have an intrinsic complexity due to their scale. In this context, the application of a management design system becomes fundamental. For this purpose the SysML language, a UML customization for engineering applications, has been applied. As far as our work is concerned, we focused on the SKA Low Telescope - Tile Processing Module, designing diagrams at different detail levels. We designed a conceptual model of the TPM, primarily focusing on the main interfaces and the major data flows between product items. Functionalities are derived from use cases and allocated to hardware modules in order to guarantee the project's internal consistency and features. This model has been used both as internal documentation and as job specification, to commit part of the design to external entities.

Key words. SKA – Square Kilometre Array – SysML – Systems Modeling Language – TPM – Tile Processing Module – FPGA – Field Programmable Gate Array – SKA Low telescope – LFAA – Low Frequency Aperture Array

1. Introduction

The Square Kilometre Array (SKA) is an international project to build a radio telescope tens of times more sensitive and hundreds of times faster at mapping the sky than todays best radio astronomy facilities. It will be built in part in Australia and in part in South Africa. The SKA is mainly divided into three telescopes, Low Telescope, Mid Telescope and Survey Telescope, according to the frequency band in which they operate. The construction of the Survey Telescope is currently deferred (Dewdney et al. 2015).

The Low Telescope, or Low Frequency Aperture Array (LFAA) covers the frequency band from 50 MHz to 350 MHz. LFAA is made up of 2^{17} stationary antennas with no moving parts. Signals from these antennas are grouped into 512 stations of 35 m each, which are electronically steered using digital signal processing techniques.

SKA management requires the application of a committed design system to avoid inconsistencies as much as possible and to identify scientific needs and the main project functionalities from the very beginning of the development cycle. We contributed to and still working on the design by applying a Systems Modeling Language, SysML. The tool we used for the design is No Magic Cameo Systems Modeler. In particular, we focused on the design of the Tile Processing Module, TPM, of the Low Telescope, specifically, we concentrated on the TPM firmware. A detailed description of TPM is presented in Chap.3.

It is important to note that the main scope of the application of a System Modelling Language is not the direct writing of the VHDL code, but to provide the programmer of the VHDL code with a description as accurate as possible of the main functionalities that the algorithm to be developed should have.

2. System Modeling Language

The SysML modeling language has been developed starting from the Unified Modeling Language (UML). UML is mainly used for software applications, while SysML gives to the designer the possibility to apply it to any system from an engineering point of view (INCOSE 2006).

One of the main advantages in applying the SysML is its interdisciplinary approach (Delligatti 2013), it can be applied indiscriminately from very small devices to extremely large and complex systems, like SKA. Other advantages are connected to the project development cycle: customer requirements and main project functionalities are identified from its very beginning; technical and economical aspects are considered in parallel during the whole design phase and at different scale level; they can therefore be controlled more efficiently.

SysML allows the design of several kind of diagrams. Each diagram highlights only some particular aspects of the project (Friedenthal et al. 2011). The model behind the SysML describes the elements and relations in the project consistently. Therefore a change in one of the diagrams causes a change in all the connected ones.

Considering the model we developed for the TPM firmware, we used four diagram types:

- USE CASE DIAGRAM: answers the question ”how is the system used?”, identifies the main actors and how they interact with the system.
- ACTIVITY DIAGRAM: shows what happens when one of the use cases is implemented. It can also show alternatives and parallel actions in the work-flow.
- BLOCK DEFINITION DIAGRAM: describes the system’s architecture and shows the hierarchy in terms of blocks and sub-blocks.
- INTERNAL BLOCK DIAGRAM: shows the internal structure and the connections between blocks in terms of properties and links between them.

2.1. SysML application to LFAA

Fig. 1 shows the structural breakdown of SKA Telescope. The level of details goes down to the main components of the tile processing module, highlighting the TPM Firmware for which we design some of the diagrams. We concentrated on those we consider more interesting.

This design system is modular since all the small parts developed by the various consortia, related to the part they are working on, can be combined to form the global SKA SysML model.

The Low Telescope model is made up of several diagrams that can go into the level of detail desired. Through different types of diagrams it is possible to highlight different points of view of the same aspect and the relations among different aspects, in order to better analyse them.

2.2. SysML Model definition

The application of SysML to SKA starts from the identification of the main requirements. From the scientific point of view, the important questions it has to answer are ”What should the system do” and ”How should the system operate”. Some of the items considered during the development cycle are:

- the system should operate in the real world: it will encounter problems connected to its management by people.
- the technology used during the design phase: existent technology can be used, which therefore could become outdated at the end of the construction phase or, on the contrary, developing technology can

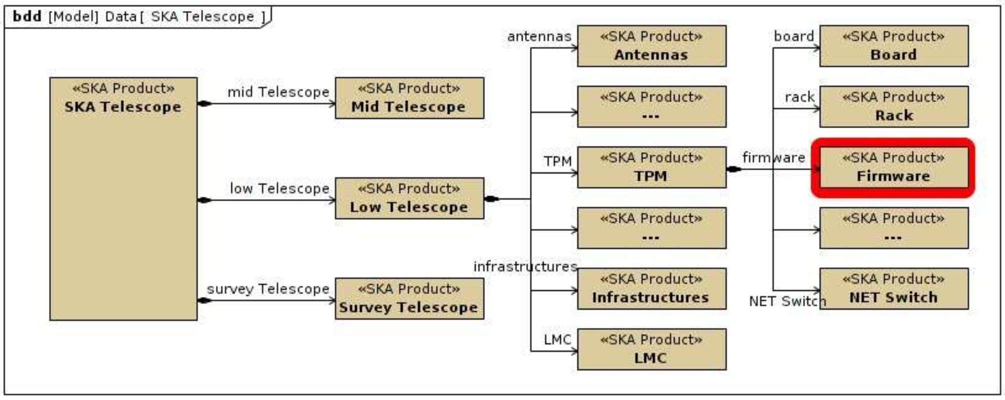

Fig. 1. Structural decomposition of SKA

be used whose real potentiality is not at present known.
- the system should operate within a legislative context and respect the environment.
- the design process is iterative since during the design itself, new requirements and constraints will arise and should be integrated in the overall design of the system.

The process in Fig.2 shows the main steps that lead to the definition of a SysML model. The starting point for the development of the SysML model is the identification of the main requirements that the system should provide. Considering these requirements, the ”USE CASE” diagrams have been developed, which becomes the starting point for identifying the constituent elements of the system and the actions that each of these elements is able to perform. Furthermore, the main characteristics of the elements have been defined, considering the data they are constituted by and the interfaces between those data.

3. Tile Processing Module

The TPM structure is described in Fig.3. The Low Telescope is made up of an extremely large number of LFAA stations, each station consists of 256 antennas, divided in 16 tile of 16 antennas each (Comoretto 2015).

Each Tile is composed of a tile processing module (for data processing) which digitalizes the signal coming from the 16 antennas and processes it (Comoretto 2016). The signal coming from the 16 antennas, with two polarisations, is converted to digital by 16 ADC with two channels each.

Every TPM is made up of two FPGA, interconnected between them, each processes the signal coming from 8 antennas. An external memory is connected to each FPGA and each TPM has an internal control system which interacts with both the FPGAs. An Ethernet interface is also connected to each FPGA, to exchange data between boards and to send the processed data to the correlator.

3.1. Firmware Diagrams

Following the process described in Chap.2, we present the diagrams that have been used. As anticipated, four different kinds of diagrams have been designed to describe the firmware, the first one is the Use case Diagram, followed by Actvity Diagrams, Block Definition Diagrams and an Internal Block Diagrams.

3.1.1. Use Case Diagram

This diagram describes the high level functionalities and their relations to external actors. The objective of the diagram in Fig.4 is the production of a Station Beam for each station, which

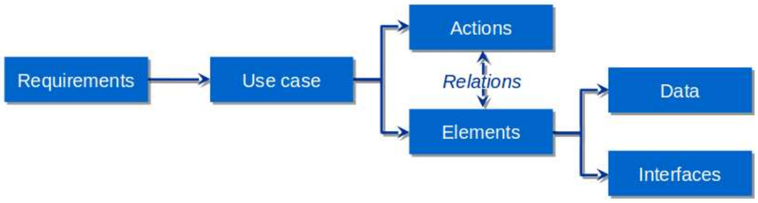

Fig. 2. SysML approach to system design

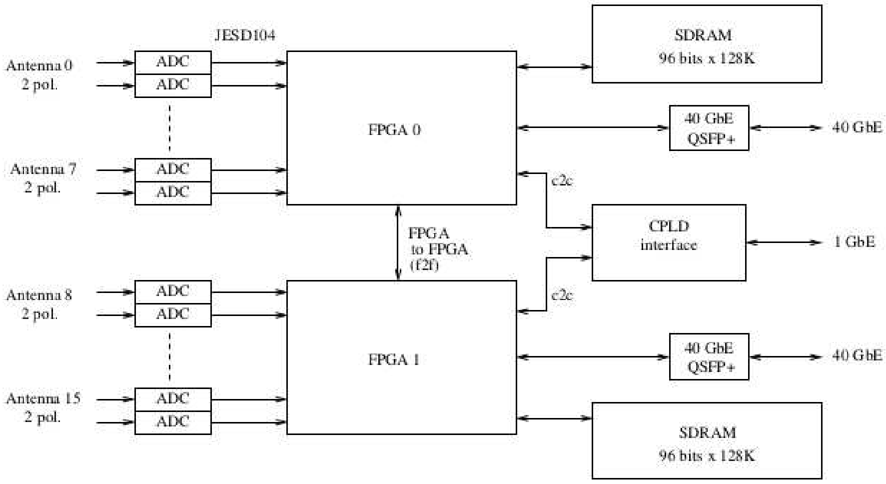

Fig. 3. Tile Processing Module Structure

represents the signal coming from a specific sky region.

For the production of the beam it is necessary to collect the data from the antennas, correlate these data through a correlator, and control the process through a monitoring and control station.

Furthermore, the diagram shows that the action "Produce Station Beam" to be performed necessitates the implementation of all the actions indicated in the lower part of the diagram.

3.1.2. Activity diagram

To analyse the functionalities in detail, we used the activity diagram in Fig.5. The "Produce Station Beam" activity is central in the Use Case Diagram in Fig.4. The columns represent the blocks, i.e. the physical objects which will implement the actions that they were assigned.

The rectangles represent the actions, while the arrows symbolize the passage of information, i.e. the activities sequence to be followed. These sequences can be more or less complicated, in Fig.5 it is possible to note that the upper left part is more linear, while the lower right part presents a more convoluted sequence of actions.

3.1.3. Block Definition Diagram

The blocks defined in the activity diagram in Fig.5, are summarized and presented in a more hierarchical way in the block definition diagram in Fig.6, in the form of blocks and sub-blocks.

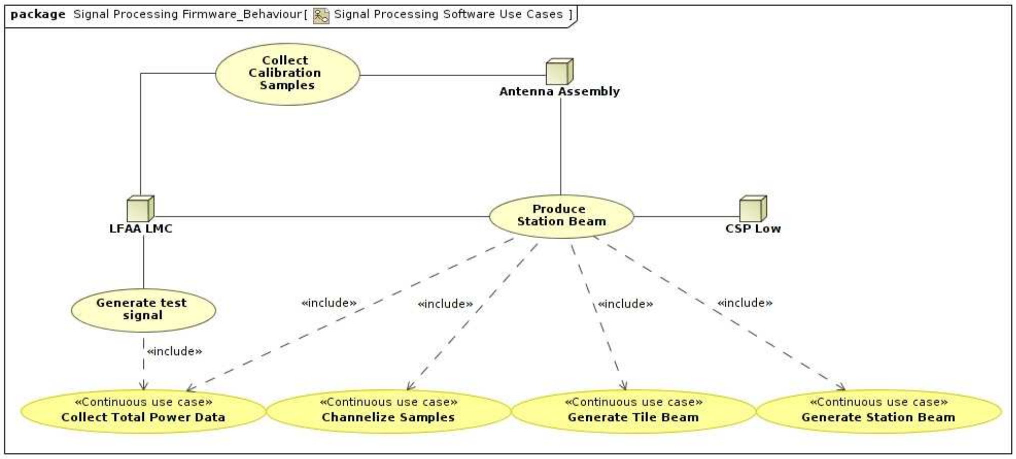

Fig. 4. Firmware - Use Case Diagram

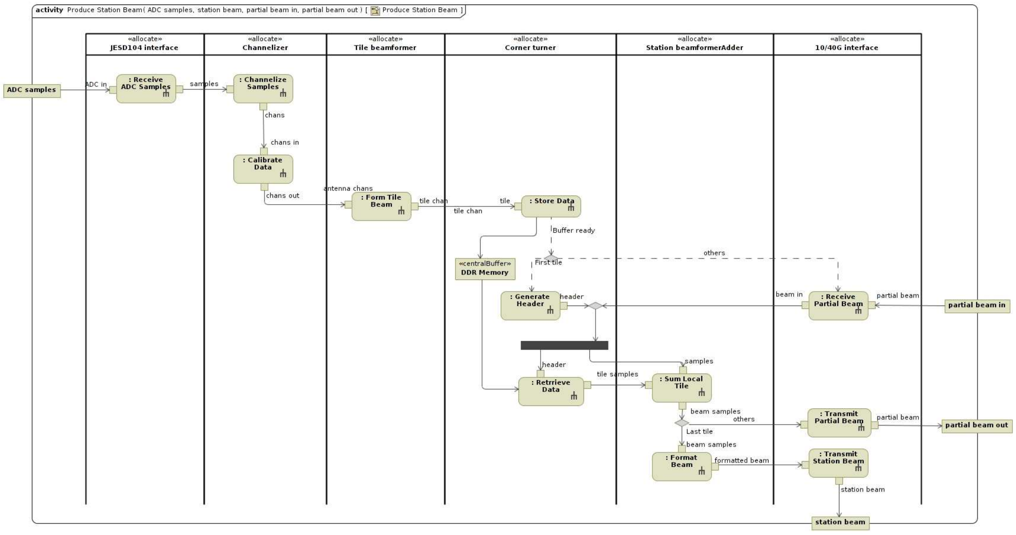

Fig. 5. Firmware - Activity Diagram

The "parts" included at the bottom side of each block are the related sub-blocks that assemble the block they are contained in. The diagram indicates also multiplicity information and may contain other kinds of relations, such as use of external blocks.

3.1.4. Internal Block Diagram

Fig.7 shows how the blocks are physically connected between them. The diagram highlights the relations among blocks, defining the interfaces they are characterized by, i.e. the relations between blocks. The interfaces in the di-

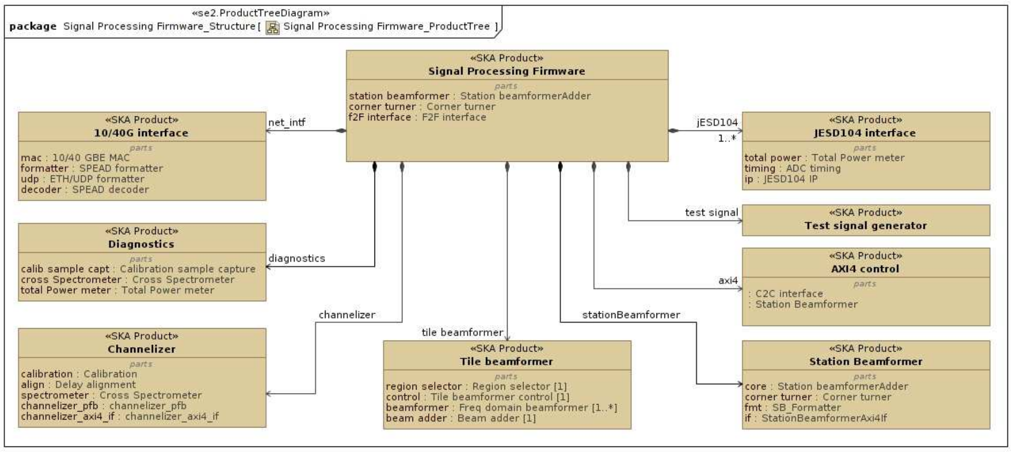

Fig. 6. Firmware - Block Definition Diagram

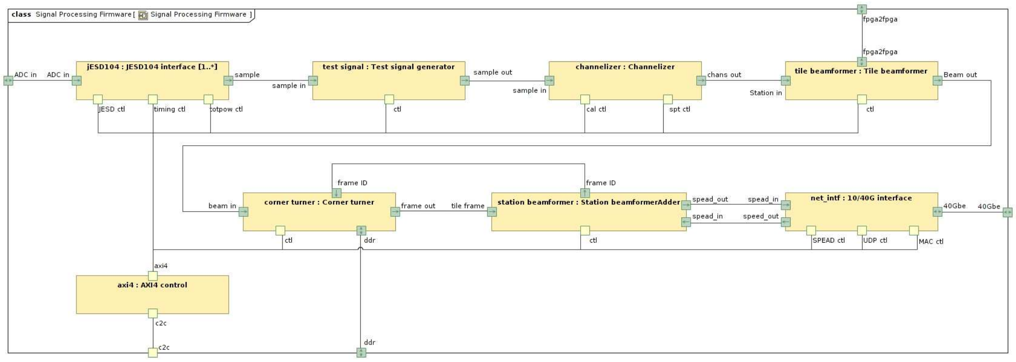

Fig. 7. Firmware - Internal Block Diagram

agram are represented by blocks; however they can also be described by specific objects.

3.2. Station Beamformer Diagrams

All the process described in Par.3.1 can be repeated in greater detail. The Station Beamformer activity, represented by an element in Fig.6, is described in Fig.8. This activity diagram depicts a more convoluted process: various parts interact with each other, exchanging information; alternative solutions can be undertaken depending on the element of the series of data that is going to be processed. Furthermore, there are also interconnections with other diagrams, with information coming from, or going to blocks belonging to other diagrams.

The described activity is related to the formation of a Station Beam, i.e. the combined signal of 256 antennas. The activities in the diagram in Fig.8 have combined together 16 antennas in a tile beam and stored it in a memory. Net interfaces receive information related to the pack of data processed and sum it to the archived data. There is a series of data that

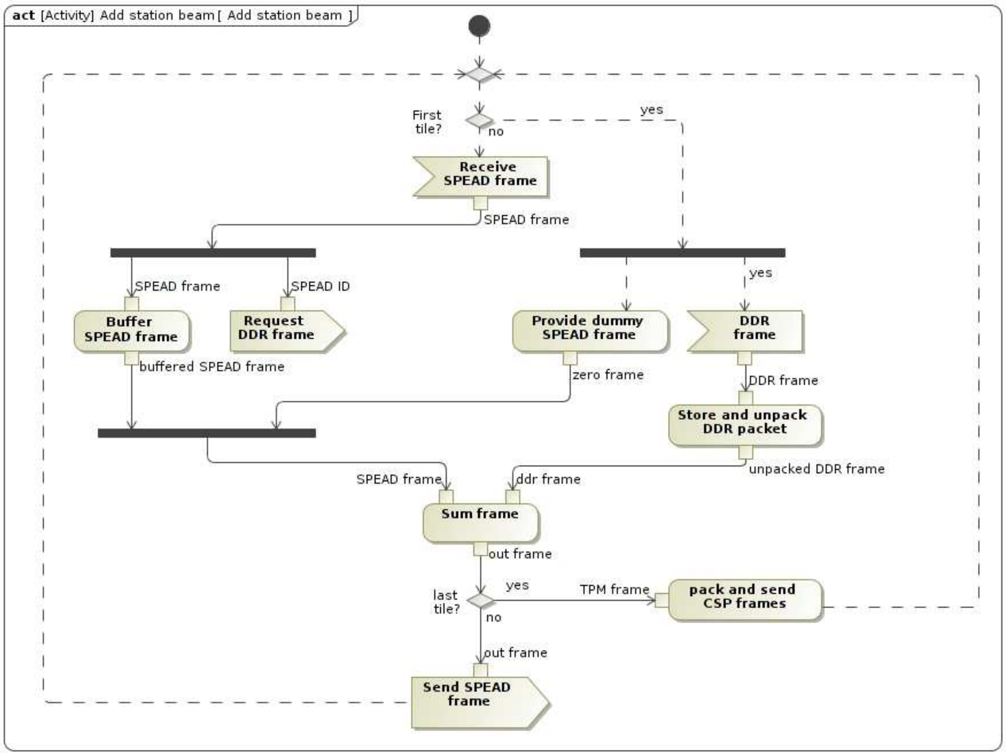

Fig. 8. Station Beamformer - Activity Diagram

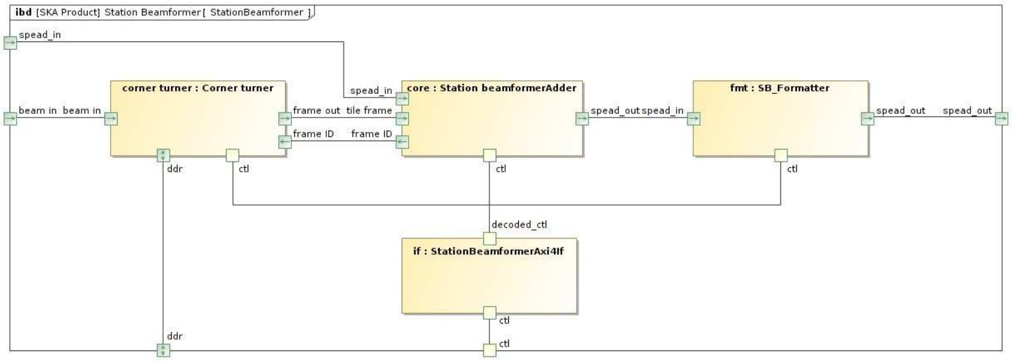

Fig. 9. Station Beamformer - Internal Block Diagram

goes backwards and forwards between memories and net interfaces. The Internal Block Diagram in Fig.9 shows the interfaces and the connections between blocks identified in the activity diagram of Fig.8; in this case we can also note bi-directional connections. It is possible to go down to even greater detail, analysing the composition of the blocks identified in this latest level, defining the interfaces that characterise them. At this latest level of detail, all the information necessary for the composition of the VHDL code is available.

The code can also include information on costs and consumption of each single module; in this way, when the codes of all the modules are ready, it is possible to have information on the overall cost and consumption of the system.

4. Conclusions

Considering the complexity of the SKA project, a designing management system is important and necessary for its development. There are various computer applications able to produce SysML diagrams; in this case we used No Magic Cameo Systems Modeler.

We developed a system of diagrams for the description of the functionalities and behaviour of the TPM Firmware. We started from the definition of the Use Case Diagram; we identified the actions to be undertaken that are part of the system, the blocks that will perform these actions and their behaviour. The diagrams presented in the current paper represent a selection of those developed to describe the project. The diagram system developed includes the main actors and the activities which are part of and act in the project. The detail level until which the iterative process of designing diagrams has been carried out is the one that allows a programmer, even from an external company and therefore with a restricted knowledge of the project, to write the VHDL code of the single modules.

As highlighted before, the objective of the development of a SysML model is not the direct production of the code from the diagrams designed, but provides all the necessary information and detailed description of the functionality that the VHDL code shall supply.

Acknowledgements. This work is supported by the Italian participation to the SKA project.

References

Comoretto, G. 2015, LFAA Tile Beamformer structure, INAF - Osservatorio Astrofisico di Arcetri, Internal Report n. 2/2015

Comoretto, G. 2017, MmSAI, 88, 154

Delligatti, L. 2013, SysML Distilled: A Brief Guide to the Systems Modeling Language (Pearson Education, New Jersey)

Dewdney, P., et al. 2015, SKA1 System Baseline V2 Description, `https://www.skatelescope.org/wp-content/uploads/2014/03/SKA-TEL-SKO-0000308_SKA1_System_Baseline_v2_DescriptionRev01-part-1-signed.pdf`

Friedenthal, S., et al. 2011, A Practical Guide to SysML: The Systems Modeling Language (Morgan Kaufmann, Amsterdam)

INCOSE 2006, Systems Engineering Handbook