Deploying Model-Based Systems Engineering in Thales Alenia Space Italia

Elvira Caliò, Fabio Di Giorgio
Thales Alenia Space Italia
Via Saccomuro, 24 – 00131 Roma, Italia
name.surname@thalesaleniaspace.com

Mauro Pasquinelli
Thales Alenia Space Italia
Strada Antica di Collegno, 253 - 10146 Torino, Italia
name.surname@thalesaleniaspace.com

***Abstract*— In the last decade, Thales Alenia Space started studying the transition of its own systems engineering methods from standard requirement and document based ones to innovative approaches taking care of concurrent engineering, enhanced collaboration, model based system engineering methods and tools and tool-chains for overall engineering environments. In the field of system architectures analysis and definition, TAS has deployed internally a tooled-up approach, which is being extended to other system and multidisciplinary engineering activities.**

Despite an investment needed to set-up the tooled-up approach, it allows to relate in a same model customer needs and architecture constraints, furthermore ensuring overall document consistency with the design by means of automatic documentation generation from the model, simplifying alignment in case of model update.

Moreover, traceability links between requirements and model facilitate impact analysis for system evolution and maintenance, and will allow modifying the architecture baseline, taking care of previous justifications.

This paper presents an outline of the TAS Model-based engineering method (ARCADIA), of the use of the related tooling, and some examples derived from the application to space projects in the Domain Observation and Navigation Italy and in the Domain Exploration and Science Italy. Observed benefits of this approach, additional needs which have been managed (such as the extension of the approach to cover the geometry of the physical components) are presented in the conclusions.

(*Abstract*)

Keywords — Systems Engineering; Model-Based; MBSE; tooled-up approach; Capella.

I. Introduction

As a Large System Integrator, Thales Alenia Space, like most of the Thales Group Entities, is very focused on System Engineering, which covers most areas of its activity spectrum, covering Observation, Navigation, Space Exploration and Science and Telecommunications.

Besides actively sponsoring the achievement of INCOSE CSEP (Certified System Engineering Professional) among its employees and with the goal of fostering a common tooled up approach and use of the same reference architectures, Thales has conceived a solution based on these core elements:

- a System Engineering methodology, called Sys-EM, which defines the successive stages of the overall engineering process,
- a Model-based engineering method for systems, hardware and software architectural design, called ARCADIA
- a THALES internal system modeling tool, Melody Advance, now released in the Open Source as Capella (https://www.polarsys.org/capella)

II. ARCADIA and Capella

A. *Comparing ARCADIA vs. INCOSE approach*

ARCADIA (ARChitecture Analysis and Design Integrated Approach) is a Model-Based engineering method for systems, hardware and software architectural design. It has been developed by Thales between 2005 and 2010 [2] through an iterative process involving operational architects from all the Thales business domains (transportation, avionics, space, radar, etc.).

Traditionally, emphasis has been put on a good definition of (mainly functional) requirements, and their allocation to each product item (broken down in subsystems, software, hardware items...), and associated traceability.

This is clearly necessary, but in many cases, engineering appears to be reduced to writing requirements and allocating them to some breakdown items, with little or no real analysis, design, check, justification of the relevant breakdown.

Furthermore, this allocation of requirements is often only considered from the functional viewpoint, neglecting other "viewpoints" that usually strongly impact this breakdown, as for instance time performance allocation and safety drivers.

Instead, ARCADIA recommends three mandatory interrelated activities, at the same level of importance:

- Need Analysis and Modeling
- Architecture Building and Validation
- Requirements Engineering

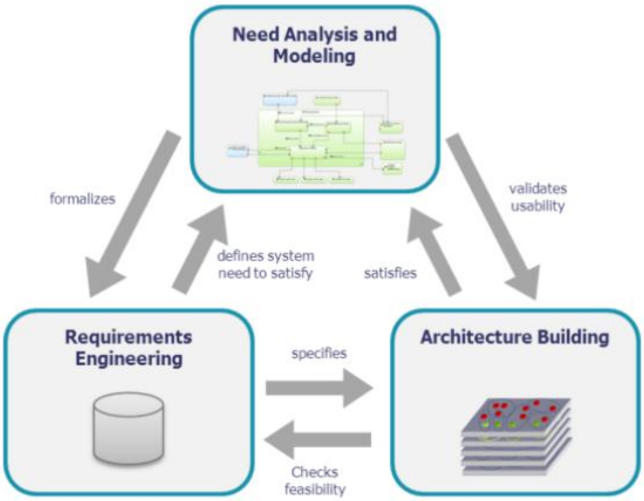

Fig. 1. ARCADIA mandatory interrelated activities

These activities will be carried out through the various detailed steps of the method.

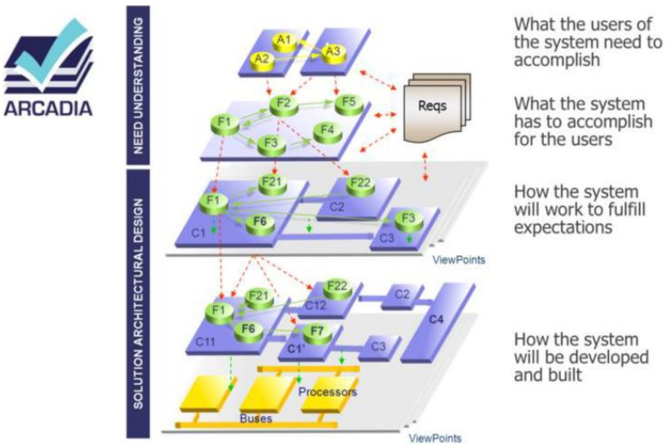

Fig. 2. Steps of ARCADIA method

It is evident that the first activity, "Need Analysis and Modeling" is strongly related to the first Life Cycle Stage as defined by INCOSE SE Handbook [1], namely the Concept Stage, with the same focus on the Stakeholder needs and the definition of an **Operational Concept** (OpsCon or "ConOps") and **Business Requirements** as an input of the following Development Stage.

In the Development Stage, again as defined by the SE Handbook, we pass from the Stakeholder needs to Stakeholder Requirements and to **System Requirements**, i.e. "**what** the System has to accomplish" to fulfill its Mission.

The System Requirements and a design activity leads thus to the definition of a (or more) solution(s) that meets the needs, focusing on "**how** the System will work". Here, the concept of **Viewpoints** is stressed both in the INCOSE approach and in the ARCADIA – Capella one.

At this point, the INCOSE approach foresees the Design Definition process, aimed at "providing sufficient detailed data […] to enable the implementation […]" which is paralleled by the Physical Architecture step in ARCADIA / Capella.

While this description, as depicted in fig. 2, seems to dictate a sequential approach, this is not necessarily the case, with several refinements steps leading to loops and iterations. This is also conceived in the ISO/IEC TR 24748-1 [2] and reported in fig. 3.2 in the INCOSE SE Handbook [1], and the Capella tool supports this by means of traceability between levels and additional capabilities (e.g. diff/merge, libraries, …).

B. *From Method to Tool: Capella*

Capella is an Eclipse based Open-Source system modeling tool, originally developed by Thales Group as Melody-Advance (part of Orchestra system engineering workbench solution).

It embeds ARCADIA method to guide users and relies on UML/SysML widely known standards, but uses a natural engineering semantic (functions, components, data…): Capella embeds a methodology browser, reminding ARCADIA principles to the user and providing efficient methodological guidance

It should be highlighted that SysML is a language, not a methodology: it provides a vocabulary, but tells nothing about using one or the other concept, about structuring models or about following design rules, etc.

The Capella interpretation of the SysML specification is built on the following main drivers:

- Provide engineers all the expression means they need
- Avoid overwhelming engineers with unnecessary complexity
- Help engineers following the Thales methodologies
- Unify the way systems and software architectures are modeled across Thales

As such:

- A large part of the Capella concepts are **directly coming from a subset of UML/SysML ones**. For example, classes, properties, parts, ports, interfaces, actors, interaction model, state machines, etc.
- Some concepts are a **Thales specialization of SysML concepts**. For example, a SysML block does not exist as such in Capella. Instead, it provides Actors, Operational Entities, Logical / Physical Components, Configuration Items which actually are all blocks.

- Some are **additions to the SysML specification**: specialized packages for model structuring, traceability links (between each Sys-EM architecture phases for example), enriched data values
- And a few concepts are **coming from NAF (NATO Architecture Framework) subset**.

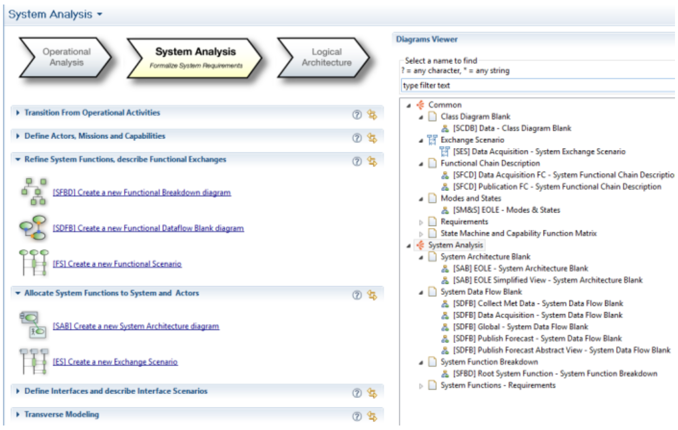

Fig. 3. Capella Methodological Activity Browser

The following sections will detail the application of ARCADIA method through Capella tool to two use cases in Thales Space Division.

III. MBSE APPROACH IN THALES ALENIA SPACE ITALIA

The following sections present two examples of use of the ARCADIA method in TASI space projects and its implementation through the Capella tool. In particular, the first example shows the application of the method for the engineering modeling of an End-to-End Earth Observation system (including both space and ground segments), while the second one shows the application of the method at Segment level (spacecraft design).

A. MBSE applied to an End-to-End Observation System

In 2014 Thales Alenia Space Italia (TAS-I) Domain Observation and Navigation Italy started a pilot project for modeling an End-to-End SAR Earth Observation (EO) System with Melody Advance (now Capella). It represented one of the first examples of exploitation in TAS-I of a new system engineering approach for a very complex system passing from the traditional "requirement-based system engineering", centered on textual requirement database, to a "model-based system engineering" supporting the formalization of Customer requirements in a system model.

The main objectives of the activity were identified as follows:

- manage the System complexity within a collaborative reference environment shared by the team
- use of standardized company engineering approach (ARCADIA) supported by a standard modeling tool (Melody Advance/Capella).
- enhance competitiveness through standardization and formalization of product reference architecture
- confirm MBSE advantages and promote its deployment in TAS-I

A multi-disciplinary engineering team, covering different competence areas within the Earth Observation Domain, has been trained and deployed. A preparation phase has been carried out before starting the project implementation to identify the best modeling approach versus engineering needs and model objectives. This step was fundamental to define the perimeter of the activity and to limit the modeling scope to what was strictly necessary with respect to the intended use of the model.

As result of the trade-off, it was decided to concentrate the modeling effort in the System Analysis and Logical Architecture steps, with a possible extension toward the Physical Architecture.

- Cooperation among different disciplines was identified for each step, in particular:
- System, Ground Segment and Space Segment engineers for System analysis
- Ground Segment, Space Segment, sub-system and payload engineers for Logical Architecture modeling
- Subsystem, payload, hardware (HW) and software (SW) engineers for the Physical Architecture modeling

The EO System modeling activity started with the System analysis, aimed a defining the system functionalities and the interactions with external entities. Within the tool this activity is mainly driven by the identification of:

- System Context, Actors & Capabilities
- System Functional breakdown and dataflow
- System Functional chains and scenarios

The SAR EO System external actors identified and captured in the Contextual System Actors diagram are shown in the following figure.

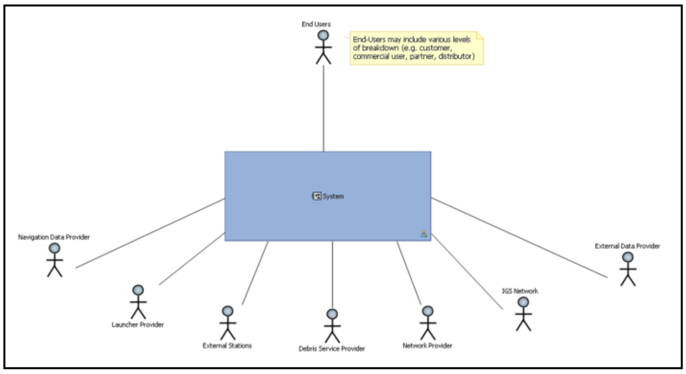

Fig. 4. System context

The primary mission of a SAR EO system is the provision of an End-to-End Service able to fast answer to the user needs, based on a flexible architecture generating and distributing that service in terms of SAR products. The End User represents in this model a variety of users that can access the system in order to make use of provided services and image products.

The identified system capabilities have been modeled in the tool and further decomposed into systems functions needed to accomplish the intended scope. The tool supports the definition and the visualization of System Functions in a hierarchical structure, highlighting with different colors the functions to be guaranteed by external actors (i.e. blue boxes) and the ones to be implemented by the System itself (green ones).

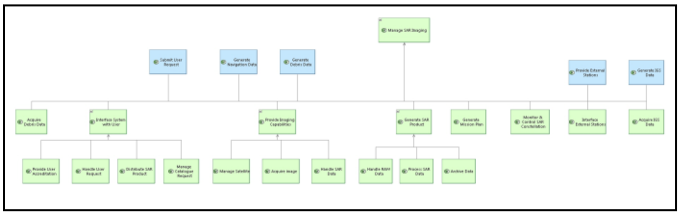

Fig. 5. Example of System Functions hierarchical breakdown

A point of strength of Capella tool is the possibility to represented the model through a variety of diagrams such as functional scenarios, functional data flow, functional chain description, exchange scenarios, mode and state diagrams, etc… Thus, the system engineers may define different diagrams to focus on their specific needs, the overall model consistency being guaranteed by the tool at any time.

The following figure shows an example of simplified System Architecture diagram for the modeled EO System.

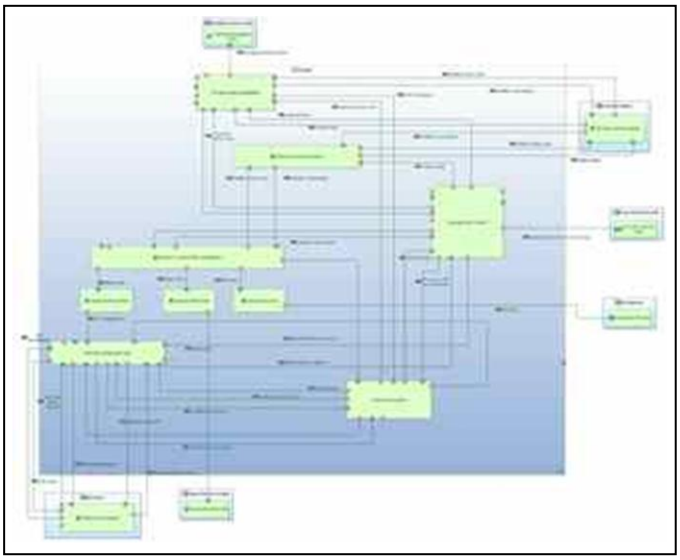

Fig. 6. System Architecture Diagram

Functional chains may be easily delineated in the model just selecting the involved functions and functional exchanges, and then represented in different graphical ways.

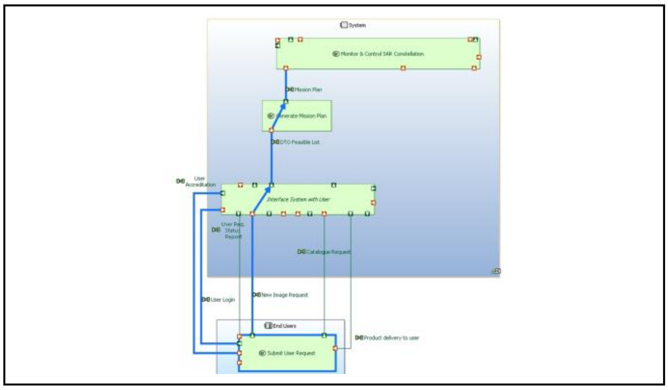

Fig. 7. Functional chain representation in the System Architecture diagram

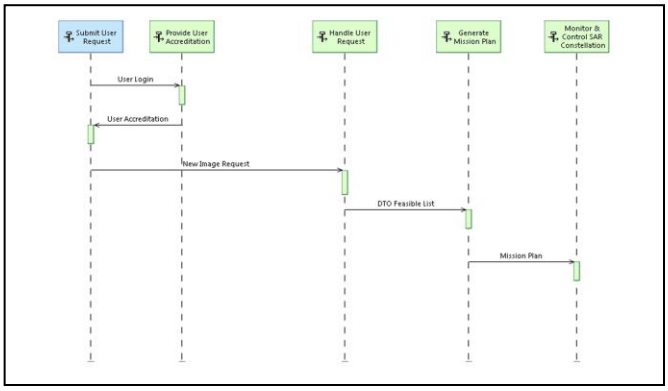

Fig. 8. Functional chain representation in the Function Scenario

Main outcome of the System analysis has been the functional description of system in its entirety, without allocating the system functions to Space or Ground systems, but modeling all functional exchanges among them (i.e. formalization of space to ground functional interfaces).

The Logical Architecture was aimed at refining the System analysis, by decomposing the system into logical components. The tool supports the automatic transition of the system functions into logical functions, to be further refined and allocated to logical components. At this stage the ground segment and space segment logical functions have been identified and allocated, still maintaining the architectural description independent from its technical implementation.

Functional breakdown, functional dataflow, functional chains, scenarios and interfaces have been modeled, bringing to a justified logical architecture design. A simplified example of the EO System Logical architecture modeling is shown in the figure.

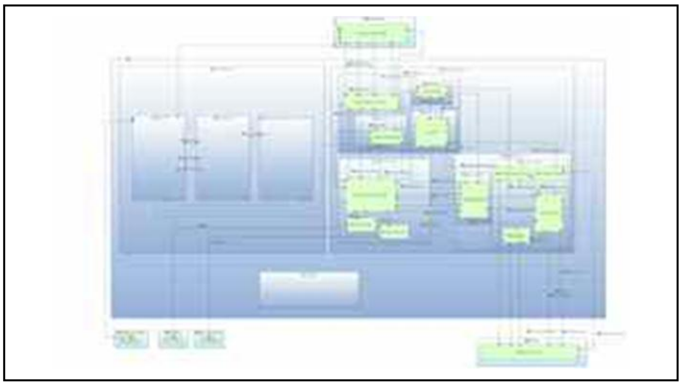

Fig. 9. Functional chain representation in the System Architecture diagram

The final step to be implemented in the pilot project was the Physical Architecture modeling, aimed at allocating the logical functions to physical functions and components. Different kinds of physical components may be identified and specified in the tool (e.g. HW, HW computer, firmware, SW, SW application, etc…). This modeling step was started for few subsystems of interest (e.g. satellite thermal control), leaving its complete implementation to future or separate projects.

In fact, since Arcadia method may be applied in a recursive way at each level of system breakdown, a subsystem of the current system may become a system in a new project/model, until single discipline subsystems or procurement items/COTS are identified. This approach has been used during the pilot project, after the completion of the logical architecture, to "extract" the ground segment architecture and to continue its modeling as a separate subsystem.

The advantage of this approach is that it allows focusing the attention of the engineering team to the subsystem of interest while maintaining the consistency with the overall system model. In fact, the tool provides the capability to:

- isolate and extract in an automatic way all the functions and interfaces to be inherited by the new model
- cross-check the consistency of the new model with the overall one, highlighting any divergences arisen after the first extraction.

Another main advantage in the use Capella is the possibility to set-up a link between textual requirements and the model. This may be implemented either using the tool intrinsic functionality to directly trace requirements or through the use of other traceability tools available in the tooled-up environment provided by Thales. In this way, it is possible to cross-validate the requirements against the architectural design solution at different levels, from the System down to the components level. In the frame of the pilot project, TAS-I has investigated how to set up a traceability link between DOORS requirement database (used for EO System Requirements formalization and flow down), the System model and the main documental outputs, to be automatically generated with proper tools available in the tooled-up environment.

B. *MBSE applied to a Spacecraft design*

In the last decade, the TAS research and development (R&D) collaborated with the corporate entities (THALES and Leonardo), with the customers (e.g. ESA and ASI) and with other space companies (e.g. in the ECSS technical memoranda WG) to analyze and define methods to support the spacecraft design and verification activities along the lifecycle. Examples are the ASI CEF&DBTE project[6], ESA Virtual Spacecraft Design initiative[7], the ESA MARVELS study [8] or the EC FP7 Use-it-wisely project [ref. SECESA paper]. The current approach follows a theoretical exploration of the main issues around the application of a complete model-based environment, associated with the application of some components of the overall methodology in different lifecycle phases.

This paragraph describes the application of Capella as one of the potential model-based tools applied to the design of a spacecraft applied to the early phases of design (namely phases A and B1 [ECSS ref.]). An overview of the association of Capella with team approach, connection with other system modeling tools and methods is then provided.

Spacecraft design is here intended as the definition of a space segment element (following ECSS definition [ref]), according to the customer specification, leading to a spacecraft architecture able to provide the breakdown in sub-components and allocate the relevant functions and interfaces to each component.

Current efforts in the space domain have been focused on the maintenance of the system consistency from a physical point of view, i.e. assuring a controlled exchange of data between different disciplines, keeping under control the resulting impact at system level of the mass, power and data volume (i.e. the system budgets), across the different phases.

Capella is not meant to provide these features, even if it is possible to enhance it through the development of viewpoints. However, it complements such existing tools and methods with additional features.

Following the ARCADIA method, the following list provides an example on how Capella may help the spacecraft definition phase:

1) *Operational Analysis*

a) Understand Customer rationale for Mission Requirements and eventually provide a formal analysis to improve or challenge the requirements and provide Engineering Change Notices

b) provide an overview of the mission, understanding the interaction between existing items (especially for spacecraft acting in an existing environment, or with a crew onboard)

2) *System Analysis*

a) Define boundaries between Space Segment (specifically the spacecraft under design), the Ground Segment and any Supporting Space Segments (e.g. to provide GNSS functions, data relay, etc.)

b) verify phase by phase that the system functions are in place, using functional chains

c) Define system modes and states, combining them with functions and functional chains

d) model relevant scenarios for the mission, especially if connected with specific requirements or to be provided as reference to different discipline specialists.

e) Understand Customer Mission and Spacecraft level requirements, allocating them to system functions and eventually provide a proper analysis to improve or challenge the requirements and provide Engineering Change Notices

3) Logical Architecture

a) allocate functions to subsystems and main logical components (e.g. the On-board computer, the AOCS sensors or the Power Storage)

b) Define Exchange Scenarios between subsystem to support the relevant specialists to understand the functional interfaces

c) Decompose the functional exchanges between Space and Ground Segment, and detail the functional exchanges between subsystem and main logical components

d) Define the main component exchanges. In early phases they can be simplified as "TM" or "Signal" or "Force", but in later phases they can be detailed, especially for data I/F's, e.g. as packets or with an associated data spec.

e) Derive Subsystem level Functional and Interface Requirements

4) Physical Architecture

a) Gather the relevant input from the disciplines specific analyses, components selection and architecture choices in order to provide a consistent physical architecture at system level

b) Verify that all the defined functions in previous levels are realized by a physical component

c) Define Physical Links as actual connections between components (e.g. a wire, a bus, an uplink connection, etc.) and allocate the related Component Exchanges

d) Define an Avionic Architecture, showing all the relevant connections to the Power Distribution Unit and to the On-board Computer

e) Allocate functions to Hardware or Software, and deploy the software to the relevant computing unit

f) Derive Equipment level Functional and Interface Requirements

The Spacecraft design is not only devoted to define and specify the components, but also to prepare the integration, validation and verification activities. In that sense all the aforementioned activities should be also coupled with the definition of scenarios and functional chains intended to be relevant for subsequent analysis and test activities.

A correct functional allocation between components, and also a clear definition of the main scenarios, are a good way to anticipate potential issues in the verification philosophy already in an early project phase.

The effectiveness of Capella usage of course shall be associated at least with:

- **Shared Approach in the team:** ARCADIA and related tooling (e.g. Capella) shall not be the "System Engineering" or "System Architect" toy. Every single stakeholder in the system definition may profit from a structured and consistent model.
- **Collaboration in the team:** the best approach is to let as much as possible all the project team members have a look in the model and being author of the part of which everyone is responsible (e.g. the avionic architecture should be managed by the On-board data handling S/S discipline, the Electrical Power Subsystem discipline, directed by the System discipline). Concentrating the work in the hand of a single person, could produce a bottleneck in the effectiveness of the tool usage, so that the model is not used <u>during</u> the design, but <u>after</u> the design.
- **Collaboration between different teams and different industrial partners:** in the case of complex industrial set-up (especially when the different subsystems are defined by different companies), a collaboration process and data exchange mean shall be defined as early as possible. TAS is researching efficient ways to improve this type of collaboration [9]
- **Connection with other SE tools:** Capella shall be used in parallel with other System-level tools, e.g. to manage the geometry, the exchange of technical parameters, the interface with discipline analysis tools. TAS is studying efficient ways to improve this types of interfaces [10]

IV. CONCLUSION

Several initiatives have been carried out in Thales Alenia Space in the last years to study and experiment innovative System engineering practices in order to transform the "traditional" way of working into a more cooperative tooled-up approach.

Continuous growing of system complexity and proven benefits deriving from wide collaboration between different engineering disciplines and specialties during the project lifetime brought to the development of concurrent engineering practices and, consequently, to the need to provide timely and structured access to shared information.

In this framework, Thales Alenia Space has taken benefit from Thales Group MBSE solutions, implementing the tooled-up approach in several space projects, and continuously fostering this approach for all new complex systems.

The advantages have been already experimented in terms of overall project consistency, team accessibility with different viewpoints, early identification of possible design issues and impact analysis, in case of requirement changes, automatically supported by the tool.

Nevertheless, in order to further benefit from this tooled-up approach additional improvements are needed to fully exploit the physical level thus supporting the lower level component detailed design and support the HW and SW implementation specification.

References

[1] INCOSE Systems Engineering Handbook, fourth edition, INCOSE-TP-2003-002-04-2015

[2] ISO/IEC/IEEE 15288 – Systems and software engineering – System life cycle processes

[3] ISO/IEC TR 24748 – Systems and software engineering – Life cycle managemet – Part 1, guide for life cycle management

[4] Pascal Roques, MBSE with the ARCADIA Method and the Capella Tool, hal-01258014 (https://hal.archives-ouvertes.fr/hal-01258014), January 2016

[5] Capella – https://www.polarsys.org/capella/index.html

[6] Portelli Claudio, Davighi Andrea, Del Vecchio Blanco Carlo, Basso Valter, Belvedere Giampiero, Rosazza Prin Paolo. ASI CEF+DBTE: Future applications of concurrent engineering methodology integrated with knowledge based economic analyses. 3rd International Workshop on System & Concurrent Engineering for Space Applications (SECESA 2008), 15-17 2008 Rome, Italy

[7] VSD website. https://www.vsd-project.org

[8] M. Pasquinelli, D. Gerbaz, J. Fuchs, V. Basso, S. Mazzini, L. Baracchi, S. Puri, L. Pace, M. Lassalle, J. Viitaniemi, Model-based approach for the verification enhancement across the lifecycle of a space system, 6th International Conference on Systems & Concurrent Engineering for Space Applications SECESA 2014 Stuttgart, 8-10 October 2014

[9] M. Pasquinelli, V. Basso, L. Rocci, M. Cencetti, C. Vizzi, S. T. Chiadò, Modelling and Collaboration across Organizations: issues and a solution, 7th Systems Engineering and Concurrent Engineering for Space Application Conference Proceedings, 4 October 2016, In press

[10] G. Garcia, X. Roser, Enhancing Integrated Design Model based process and engineering tool environment : towards an integration of functional analysis, operational analysis and knowledge capitalisation into in co-engineering practices, 7th Systems Engineering and Concurrent Engineering for Space Application Conference Proceedings, 4 October 2016, In press