System concurrent engineering of a Star Sensor

Geilson Loureiro[1]

Márcio Afonso Arimura Fialho[2]

Ana Paula de Sá Santos Rabello[3]

INPE - Brazilian Institute for Space Research

Av. dos Astronautas 1758, São José dos Campos, SP, Brazil; 12227-010

[1]geilson@lit.inpe.br; [2]maaf@dea.inpe.br; [3]anapaula@dea.inpe.br

Abstract: This paper presents the system concurrent engineering approach applied for the development of an autonomous star sensor. Traditional approaches focus solely on one aspect, that can be either the product, the development organization or the product operation. In those approaches the overall view of the inherent complexity in the development of a product, its life cycle processes and their performing organizations are not taken into consideration. Contrasting with those approaches, the system concurrent engineering performs, simultaneously, stakeholder analysis, requirements analysis, functional analysis and implementation architecture analysis for the product, its life cycle processes and their performing organization. From this, requirements and attributes are captured for the product and its life cycle processes organization, and the relationships between them are identified. Key conclusions are that the impact, traceability and hierarchy links promote the anticipation of the life cycle process requirements to the early stages of systems architecting. Late changes are avoided and development costs are significantly reduced, while satisfaction of stakeholders over the product life cycle is increased.

1 - Introduction

A star sensor (or star tracker) is an attitude sensor. Attitude sensors are instruments used aboard a spacecraft or an aircraft to gather information that can be used to calculate the spacecraft / aircraft attitude. In spacecraft engineering and in aeronautics, *attitude* means the spatial orientation of a body, being usually expressed by the yaw, roll and pitch angles (in case of aircrafts) or by a quaternion or an attitude matrix (in case of a spacecraft). The attitude gathered by attitude sensors in a spacecraft is compared to the desired attitude, and corrections are done if necessary, as depicted in Figure 1.

A star sensor uses the stars as references for attitude determination. Since the stars are very far away and maintain their apparent positions in the celestial sphere for many centuries (fastest moving star moves less than 11 arcsec/year) [1], a star sensor can provide an absolute reference for attitude determination. There are many types of star sensors. In this paper we focus on the system engineering aspect for a fixed head autonomous star sensor, i.e., a star sensor without moving parts capable of providing a complete attitude solution without the need of other attitude sensors.

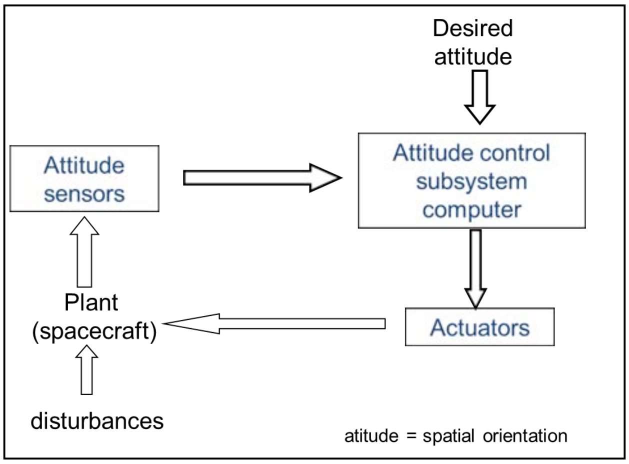

Figure 1 – A typical attitude control loop for a spacecraft.

This paper aims to present the system concurrent engineering approach applied for the development of the aforementioned star sensor/star tracker, here referred as the AST – autonomous star tracker. The approach used here and proposed by Loureiro [2, 3] is different from traditional systems engineering approach because it anticipates to the early stages of system architecting the product life cycle process requirements. It proposes to simultaneously develop, from the beginning, the product and its life cycle processes performing organizations. This approach has also been applied to the development plan of other systems, such as a mobile TT&C ground station for an UAV [4], a green car [5] and an EGSE for an On-Board computer [6], among others.

The paper is organized as following:

- Section 2 presents the opportunities for improvement especially in the practice of the traditional systems engineering and concurrent engineering approaches. Although the traditional concepts of systems engineering and concurrent engineering cover the entire life cycle of a product with its enabling organizations, the practice of these approaches are product focused and treat life cycle processes in isolation from each other.
- Section 3 presents the systems concurrent engineering approach framework and method. It describes the systems engineering and concurrent engineering for the integrated development of a product and its life cycle processes performing organization, highlighting the fact that product and organizations are systems engineered simultaneously. This section also details the system concurrent engineering method.

- Section 4 exemplifies the use of the system concurrent engineering approach for a star sensor. The exemplification does not include the complete set of stakeholders, requirements, functions and architectural elements. The example does not mean to be complete. It aims only to exemplify the steps provided in Section 3.
- Section 5 discusses the opportunities for improving the traditional systems engineering and concurrent engineering practice by using the system concurrent engineering approach.
- Section 6 concludes this paper.

2 – Traditional systems engineering and concurrent engineering

Being complex products, the development of a star sensor/star tracker is a multidisciplinary task, involving knowledge in the areas of Astronomy, Optics, Electronics, Mechanics, Software Engineering, to name a few. They must cope with extreme environments during launch (vibration) and operation in space (ionizing radiation, vacuum environment). Also, they must be able to provide accurate and dependable measurements for years, without any kind of preventive or corrective hardware maintenance, since once in space, maintenance is not possible (with very few exceptions). Being a multidisciplinary activity, the development of a star sensor requires the interaction of professionals from many different areas. By using a concurrent engineering approach from the beginning, many requirements and possible problems are anticipated. This leads in many opportunities to improve productivity during every project and product life cycle stages (early design, detailed design; manufacturing, AIT, operation, and disposal) if a concurrent engineering approach takes place from the beginning of the development effort.

According to Loureiro et al [4, 5, 6], traditional systems engineering approaches do not provide an overall view of the system during its various life cycle processes. Their main focus is on product operation, both for the product development and for the development organization that must be set up in order to assure that the product meets its operational requirements. This lack of view may increase costs, since the product will be subject to late changes and will not be developed considering the various requirements of its other life cycle processes and other life cycle process organizations. This is shown in Figure 2, that compares the traditional approaches to the approach proposed by Loureiro.

Loureiro states that even though concurrent engineering acknowledges the benefits of anticipating life cycle process requirements to the initial states of product development, modern standards such as EIA 632 [7] and NASA system engineering handbook [8] still system engineers products with operations in mind, lefting other life cycle requirements for the moment when the system/product architectures are defined. This paper presents a method, introduced by Loureiro, that takes into consideration these requirements early in the development of a product.

3 - The systems concurrent engineering approach

This section has been presented in a number of papers by Loureiro et al.[4, 5, 6]. It is presented here again (with some modifications) for reader’s convenience:

“Hitchins [9] states that complexity can be understood by what he calls complexity factors. They are variety, connectedness and disorder. Variety accounts for the number of different elements you have in a set. Regarding products, variety refers, for example, to the number of different parts a product may have, number of different functions it accomplishes, number of different requirements categories it is supposed to meet, number of different stakeholders it should satisfy. Connectedness refers to the relationships among elements. For example, how parts interact, how functions affect one another, how requirements conflict to each other, how value flow among stakeholders. Disorder refers to the level of tangling of those relationships. For example, is there a structure pattern of deploying stakeholder requirements through functional concept up to implementation architecture?

Figure 2 presents a framework to address complexity in product development – the total view framework evolved from [2]. It has three dimensions. Each dimension addresses one of the complexity factors mentioned above. The analysis dimension addresses the variety factor. Along the analysis dimension, it is deployed what must be analysed in order to develop a complex product. A systems engineering process consists of stakeholder analysis, requirements analysis, functional analysis and implementation or physical analysis. The integration dimension addresses the connectedness factor. It defines what must be integrated along an integrated product development process: product elements and organization elements. Organization here refers to the organizations that perform product life cycle processes. Product elements and organization elements are the system elements. The structure dimension addresses the disorder factor. According to Alexander [10] all structures evolve into a hierarchy. System breakdown structures are also represented in hierarchies.

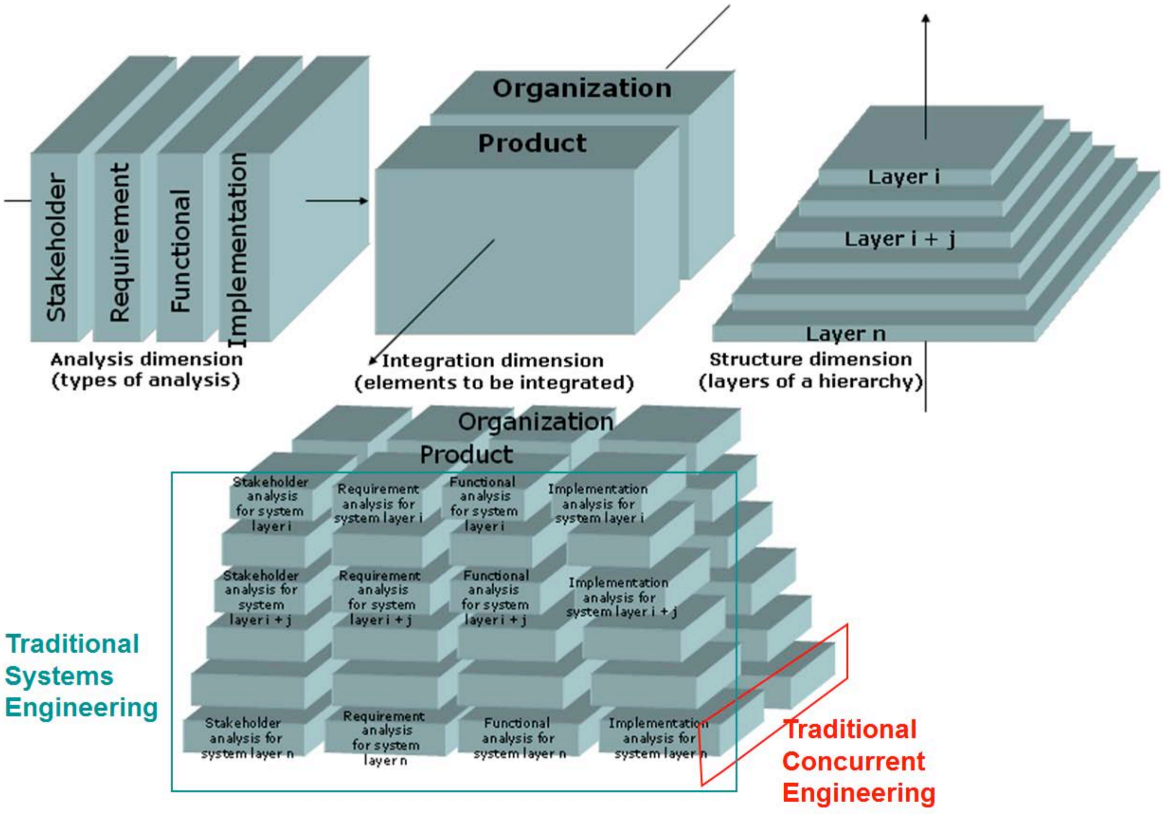

Figure 2 – A framework to address complexity in complex product developments – the total view framework

Figure 3 provides an overview of a method within the total view framework. The method is called concurrent structured analysis method evolved from [2]. Stakeholder analysis, requirements analysis, functional analysis and implementation (or physical) analysis is performed, simultaneously, for the product under development and its life cycle process performing organizations. The analysis processes are performed at each layer of the system breakdown structure. For example, if a car is the product under development, the analysis processes are performed at the car layer, at the powertrain layer, at the engine layer and so on.

Figure 4 details the concurrent structured analysis method showing how to incorporate the concurrent engineering concept in the systems engineering process. The method comprises four major steps. Step 1 refers to mission analysis, life cycle process and scenario analysis. Step 2 refers to product and organization stakeholder requirements and the system requirements derived from these stakeholder requirements. Step 3 refers to functional analysis with hazard and risk analysis, simultaneously, for both product and organization. Step 4 refers to implementation architecture derivation and analysis.

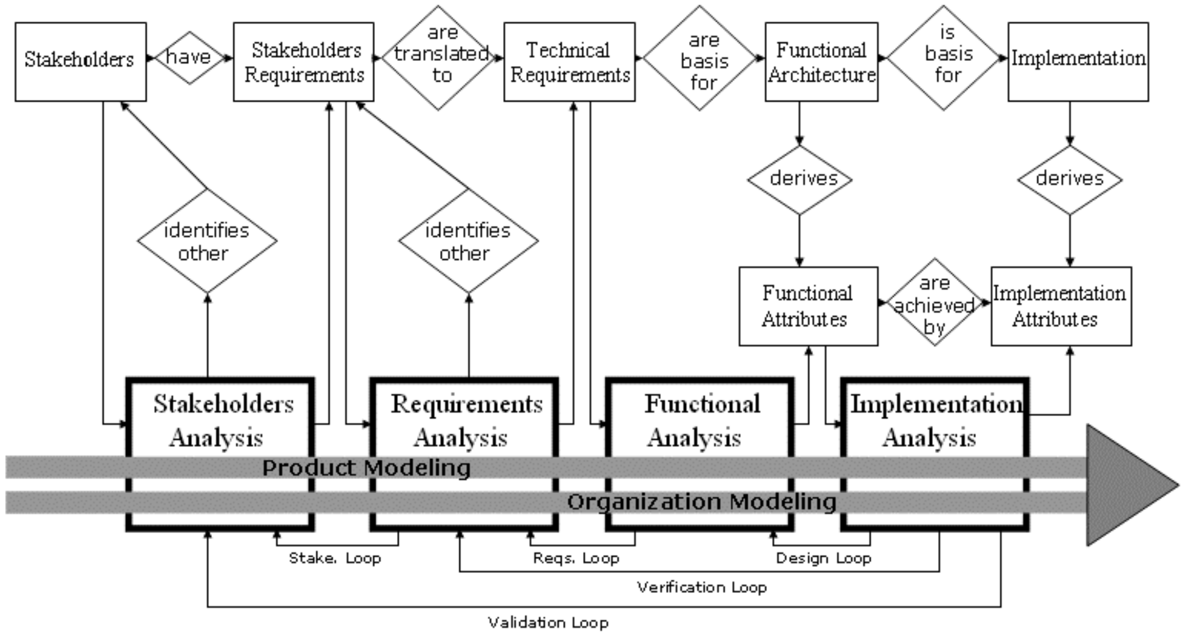

Figure 3 – A method within the total view framework – the concurrent structured analysis Method.

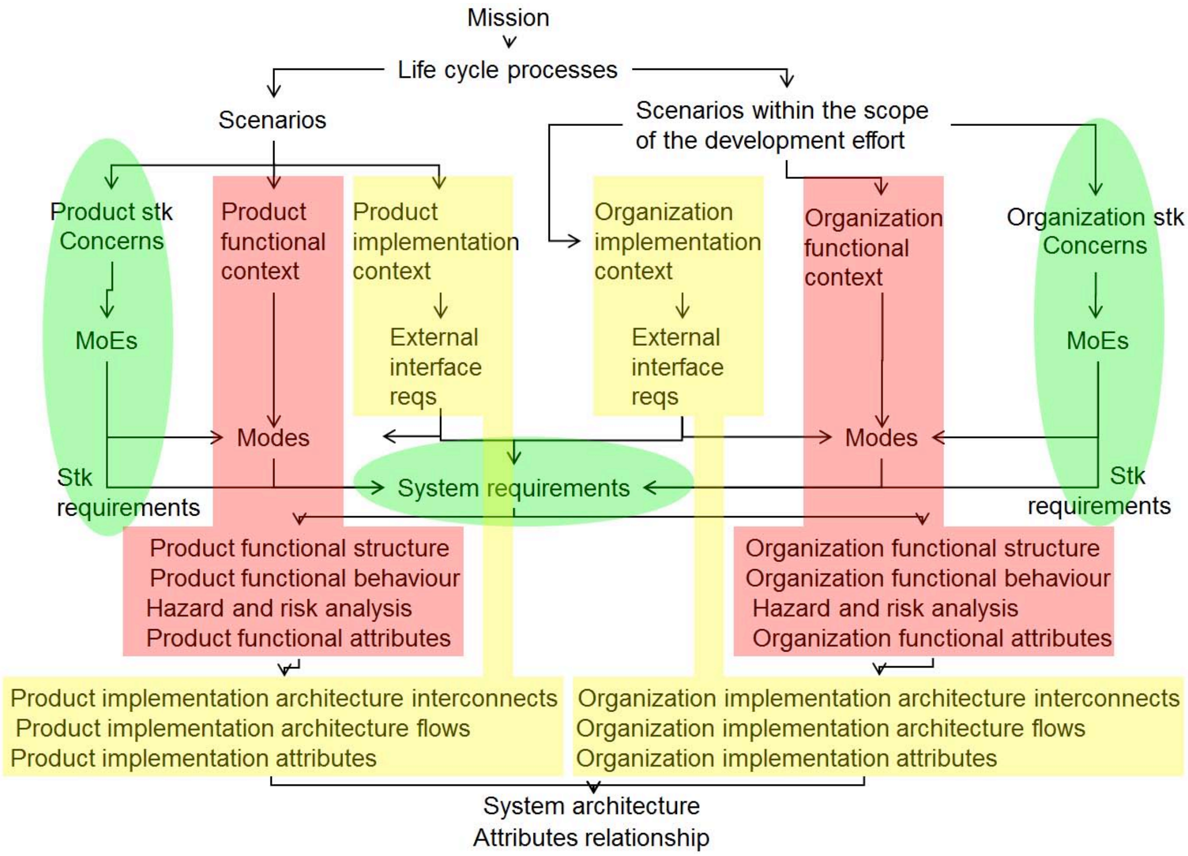

Figure 4 – The system concurrent engineering method in detail.

Step 1: *Identify the product mission, the product life cycle processes and their scenarios and, the scope of the development effort. Product mission refers to the product purpose or reason of being. Life cycle process scenarios are the alternatives in each process (for example, preventive or corrective maintenance) or the decomposition of a process (for example, advanced technology development, process engineering as components of the development process). The scope of the development effort consists of the life cycle processes or their scenarios that the development organization is also responsible for accomplishing. For example, EMBRAER is responsible for developing aircraft but is also responsible for providing maintenance services.*

Step 2: *Identify product stakeholders and their concerns for each product life cycle process scenario. Product stakeholders are the people who affect or are affected by the product during its life cycle. Product stakeholders are identified per life cycle process scenario. Identify organization stakeholders and their concerns for each process within the scope of the development effort. Organization stakeholders are the people who affect or are affected by the business of the organization in question. Organization stakeholders are identified per life cycle process scenario within the scope of the development effort. From stakeholder concerns, stakeholder requirements are identified and measures of effectiveness (MoEs) are derived. MoEs must measure how the system meets the stakeholder requirements. From stakeholder requirements, functions, performance and conditions are identified. The definition of what functions the system will perform, how well the system is going to perform such functions and under which conditions comprise the requirements analysis process. Requirement analysis transforms stakeholder requirements into system requirements. System requirements will be met not only by product elements but also by organization elements.*

Step 3: *Identify functional context for product at each life cycle process scenario and for organization at each life cycle process scenario within the scope of the development effort. Functional context defines the function performed by the system element and identifies the elements in the environment of the system. The environment of the system contains the elements outside the system function scope and that exchanges material, information and energy flows with the system. Those flows define logical interface requirements. Environment elements may have different relevant states. Sets of environment element states are called circumstances. The system must have different modes depending on the circumstances. Behaviour modelling is required to show under which conditions system mode and system state transition occurs. Functions are identified per mode. Functions are identified from outside in by identifying which responses the system is supposed to give to deal with each stimulus provided by the environment elements. For each function, performance requirements are identified. Circumstances, flows between the system and the environment and function failures are sources of hazards. Risk analysis is performed on each identified potential hazard and exception handling functions are also identified at this stage.*

Step 4: *Identify implementation architecture context for product at each life cycle process scenario and for organization at each life cycle process scenario within the scope of the development effort. Physical connections between the system and the environment elements define the physical external interface requirements. Physical parts are identified. Physical internal interfaces are defined by architecture connections and architecture flows among those parts. Allocation matrix relates physical parts and physical interfaces to the functions and functional flows."*

4 - The star sensor system concurrent engineering

This section illustrates the steps listed in Section 3 highlighting where the system concurrent engineering approach is different from traditional approaches, using an autonomous star sensor as a case study. The system concurrent engineering approach is stakeholder driven whereas traditional approaches are customer or user driven. In the various steps listed in Section 3, analyses are performed for each life cycle process scenario, simultaneously, for product and organization. Traditional approaches focus on product operation and development organization.

Steps 1 to 4 in section 3 must be run for all life cycle process scenarios, however, due to space constraints, stakeholders, requirements, functional and implementation architecture will be exemplified only for the development, AIT (Assembly Integration and Testing), and operation life cycle processes. Figure 5 presents the organization and product life cycle processes and scenarios for the AST.

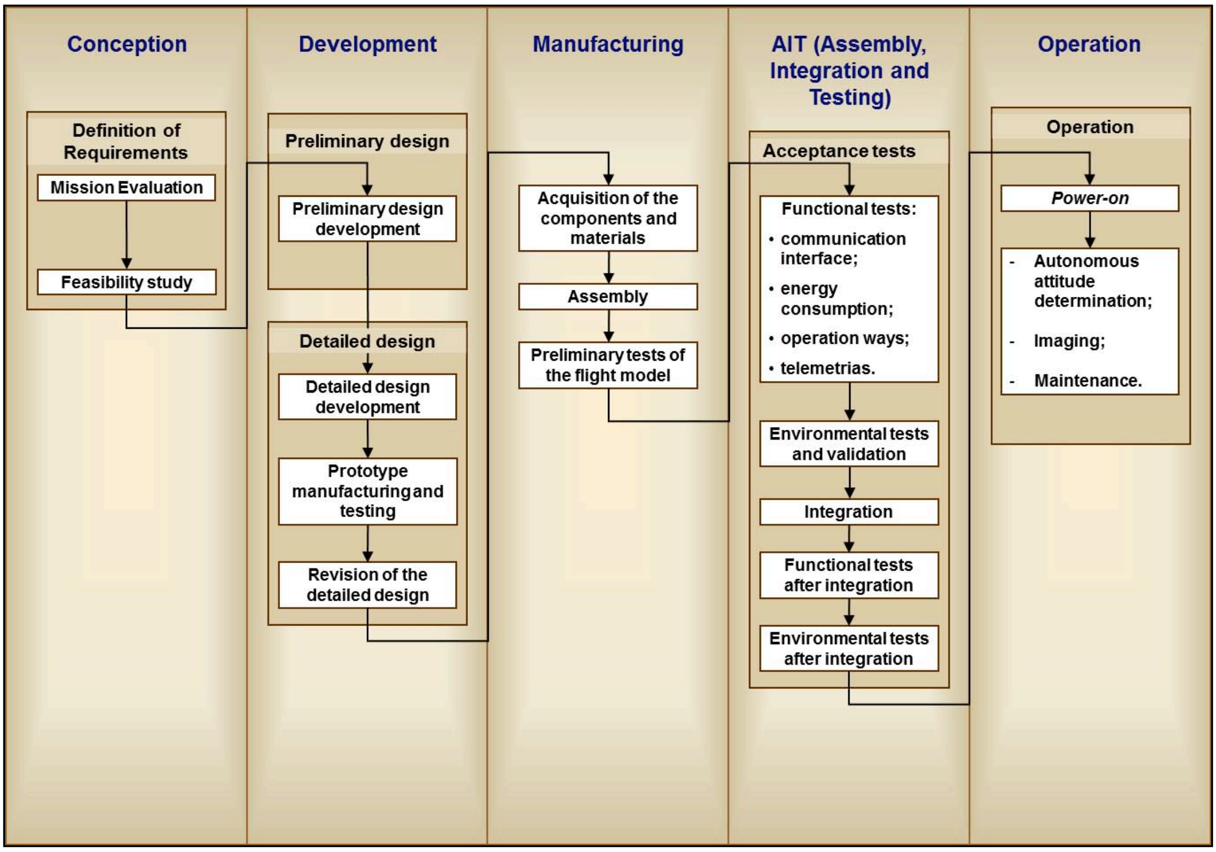

Figure 5 – Life cycle processes and scenarios of a star sensor.

After the identification of the organization and product scenarios (e.g.: development, manufacturing, operation), the stakeholders for each scenario are identified and their interests captured. Figures 6 - 9 show IDEF0 diagrams that have been used to identify key stakeholders for the aforementioned life cycle processes: development, AIT (organization perspective), AIT (product perspective) and product operation.

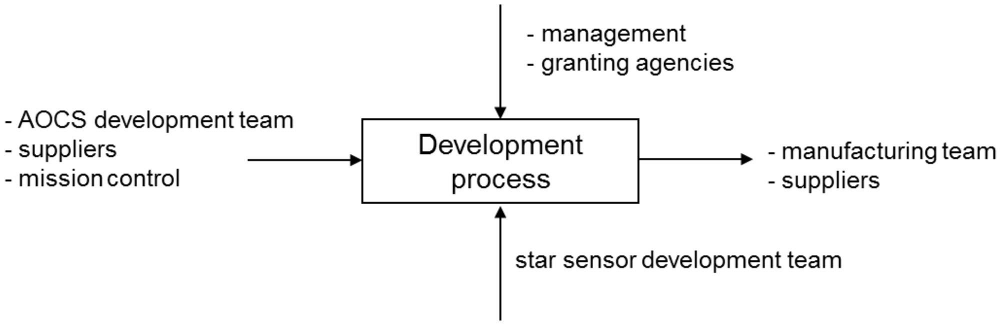

Figure 6 – Key stakeholders for the development process.

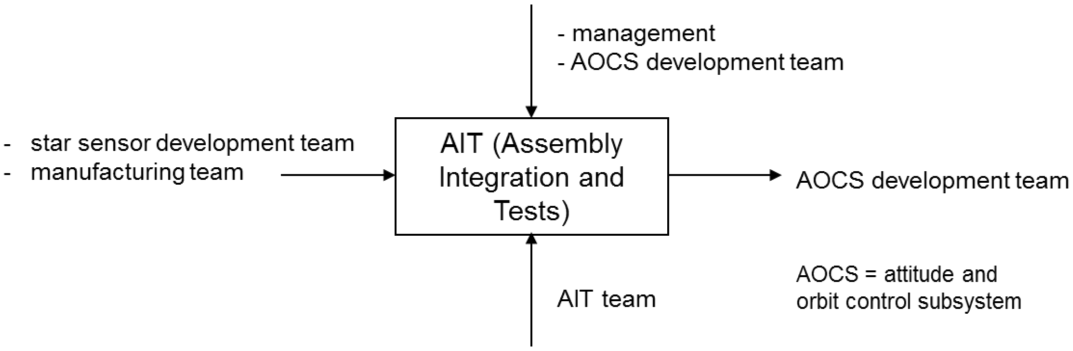

Figure 7 – Key stakeholders for AIT tasks (organizational perspective).

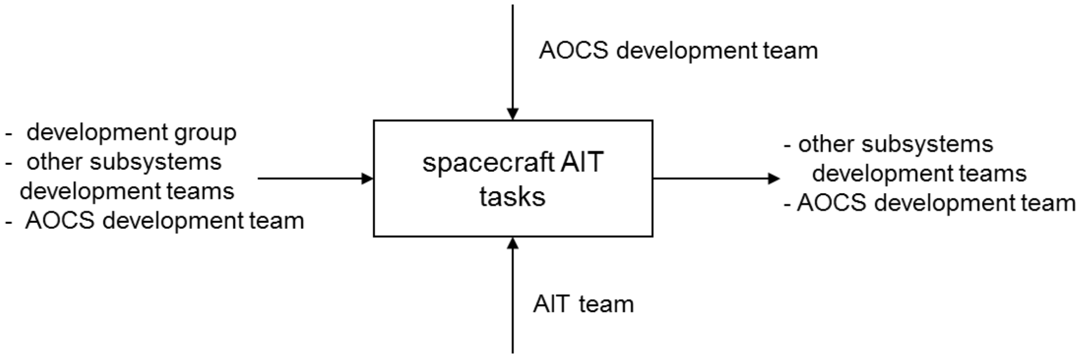

Figure 8 – Key stakeholders for AIT tasks (product perspective).

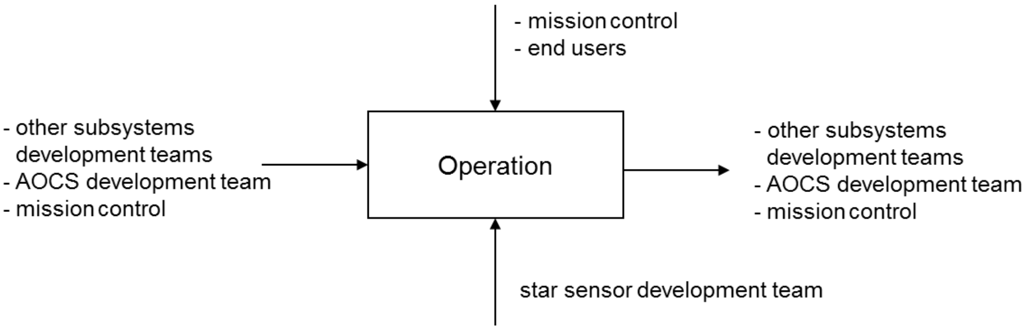

Figure 9 – Key stakeholders during product operation.

After the key stakeholders have been identified, their interests and concerns are investigated. These are collectively known as “stakeholder requirements”. As an example, Table 1 presents the stakeholders requirements for the development scenario, while Table 2 presents the stakeholders requirements for the product AIT scenario.

Stakeholder requirements also influence the scenarios that the system and the development organization will encounter. For example, the stakeholder requirement that the AST should also be able to function as an imaging camera creates the “imaging” operating mode in the “operation” life cycle process (see Figure 5). The AST operating in the “imaging mode” is a different scenario from the AST operating in the “autonomous attitude determination mode”, both happening in the “operation” life cycle process.

Table 1 – Organization stakeholders and their concerns for the development scenario.

Organization stakeholders in the development scenario	Stakeholder requirements
INPE management and granting agencies	- confirmation of the project budget in development scenario; - comply with the schedule; - use COTS (commercial of the shelf) components as far as possible; - use components qualified for space use.
AOCS project team	- meet star sensor requirements; - meet the schedule.
Development team	- meet star sensor requirements.
Manufacturing team	- receive detailed documentation for fabrication of qualification models and flight models of the AST; - receive materials.

Table 2 – Product stakeholders and their concerns for the AIT scenario.

Product stakeholders in the tests and integration in the system scenario	Stakeholder requirements
AOCS project team	- meet functional requirements; - compatibility of received and sent messages from the AST; - compatibility of electrical interfaces with other AOCS instruments; - comply with the mechanical requirements; - meet the environmental requirements (radiation, temperature, vibration,...); - compliance with EMI/EMC requirements.
Team of integration and tests	- references for optical alignment (example: alignment cube).

To derive systems requirements, MoEs (Measures of Effectiveness) are defined for each stakeholder concern. MoEs attempt to measure how well stakeholder requirements are fulfilled, and serve as a guideline in the definition of the system requirements. Figures 10 and 11 presents MoEs for two scenarios: development of the star sensor, and star sensor during operation, respectively.

From MoEs, the initial organization and product requirements can be derived. Some of these initial requirements for organization and product are presented in Tables 3 and 4. This is a very small subset of the total number of requirements for a project like the star sensor, whose total number tends to be in the number of hundreds. After functional analysis, architecture analysis, risk analysis and risk mitigation are performed, more requirements are aggregated to these lists.

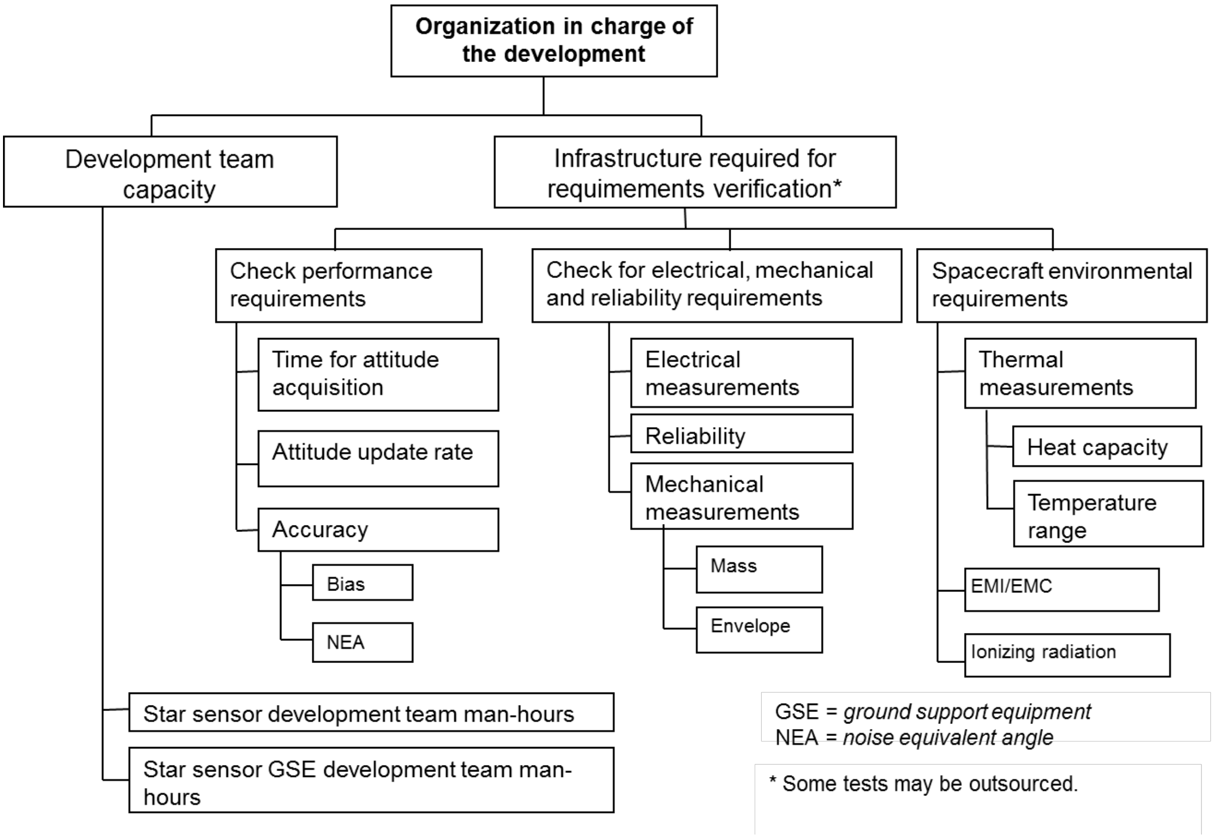

Figure 10 – MoEs for the product development scenario.

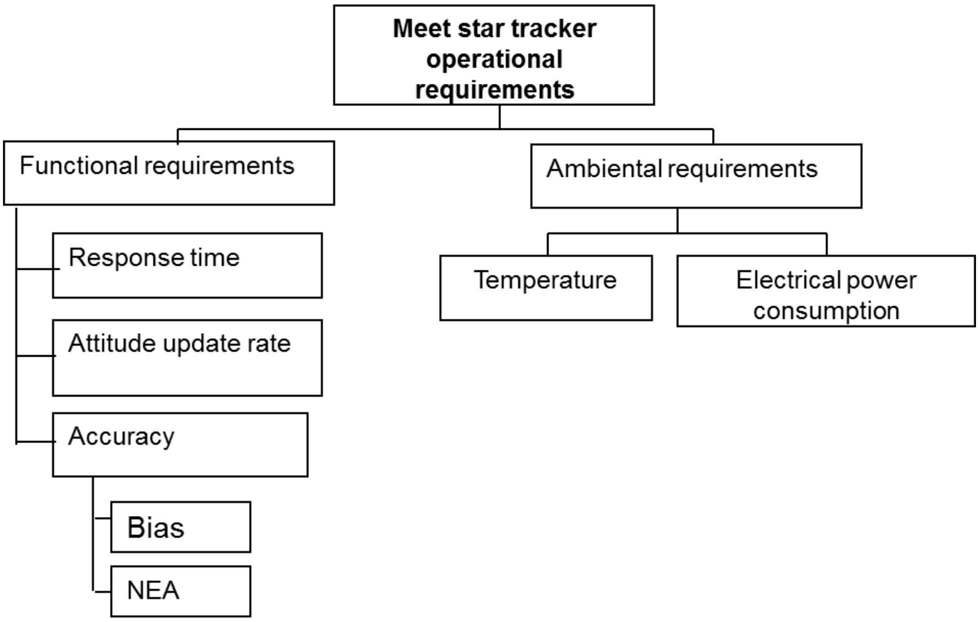

Figure 11 – MoEs for the product operation process.

Table 3 – Requirements list (subset) for the development organization

Item	**Scenario**	**Stakeholder**	**Stakeholder Requirements**	**Sub-item**	**Requirements for the Development organization**	**Type** Performance Functional Conditional	**Compliance** Mandatory Desirable	**Verifi-ability** Analysis Inspection Test
R0	Development	INPE management and granting agencies	schedule compliance	R0.1	Development team must conclude the project within xxx months.	C	M	I
			keep within budget	R0.2	Development team must be able to finish the project with a budget of $ xxx	C	M	I
R2	Development	Manufacturing team	receive detailed documentation	R2.1	Development team shall provide detailed documentation for manufacturing the qualification and flight models.	C	M	I

Table 4 – Requirements list (subset) for the AST product

Item	**Scenario**	**Stakeholder**	**Stakeholder Requirements**	**Sub-item**	**System requirements**	**Type** Performance Functional Conditional	**Compliance** Mandatory Desirable	**Verifi-ability** Analysis Inspection Test
R1	Operation	Mission Control center	correct handling of received telecommands	R1.1	AST shall check telecommand messages before executing them.	F	M	I, T
			send telemetries to the ground correctly	R1.3	Telemetry messages shall be protected by check/ECC fields.	F	M	I, T
			Obtain temperature in critical points in the star tracker.	R1.4	AST shall make available temperature telemetries for some key internal points.	F	D	I, T
R2	Operation	Other spacecraft subsystems teams.	meet spacecraft mass budget	R2.1	AST total mass shall be no greater than 5 kg.	C	M	I, T
			meet spacecraft volume budget	R2.2	Excluding the sun baffle, the AST shall fit inside a 20cm x 20cm x 40cm envelope.	C	M	I, T
			meet the spacecraft power budget	R2.3	The mean power consumption for the AST shall be no greater than 12 W, considering any 10 second interval.	C	M	I, T
				R2.4	The peak power consumption shall be no greater than 24 W.	C	M	I, T
			Be reliable	R2.5	The AST reliability shall be greater than 96% for a mission duration of 3 years in LEO orbit (h < 900 km).	C	M	A

Figures 12 and 13 present two context diagrams: one of them for the development organization during the development phase and the other for the product during operation. These context diagrams also present some circumstances that the development team may face during project development, and that the star tracker may encounter during its operation. Circumstances are the possible states that elements which relate to the product or organization may assume. To handle various different circumstances the system and organization may need to have additional operating modes and functions. For instance, a star tracker may operate with its usual algorithms when it has a clear view of the sky, but switch to a modified algorithm to cope with additional noise, when it becomes partially blinded by the Sun or the Earth.

The system must have different modes according to the circumstances. To uncover in which conditions system mode and system state transition occurs, behaviour modelling is employed. Functions are identified per mode. Functions are identified from the outside in by identifying which responses the system is supposed to give to deal with each stimulus provided by the environment elements. For each function, performance elements are identified. Circumstances, flows between the system and the environment and function failures are sources of hazards. Risk analysis is performed on each identified potential hazard and exception handling functions are also identified at this stage.

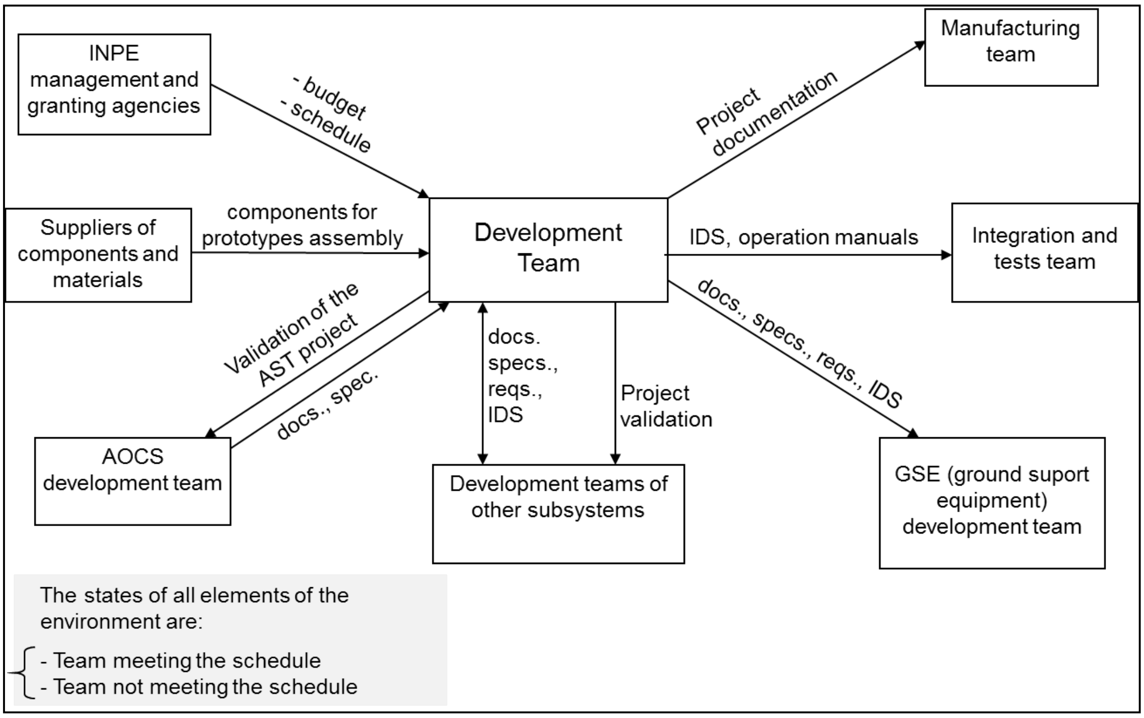

Figure 12 – A context diagram for the development team, showing material and information exchanges with other teams/entities.

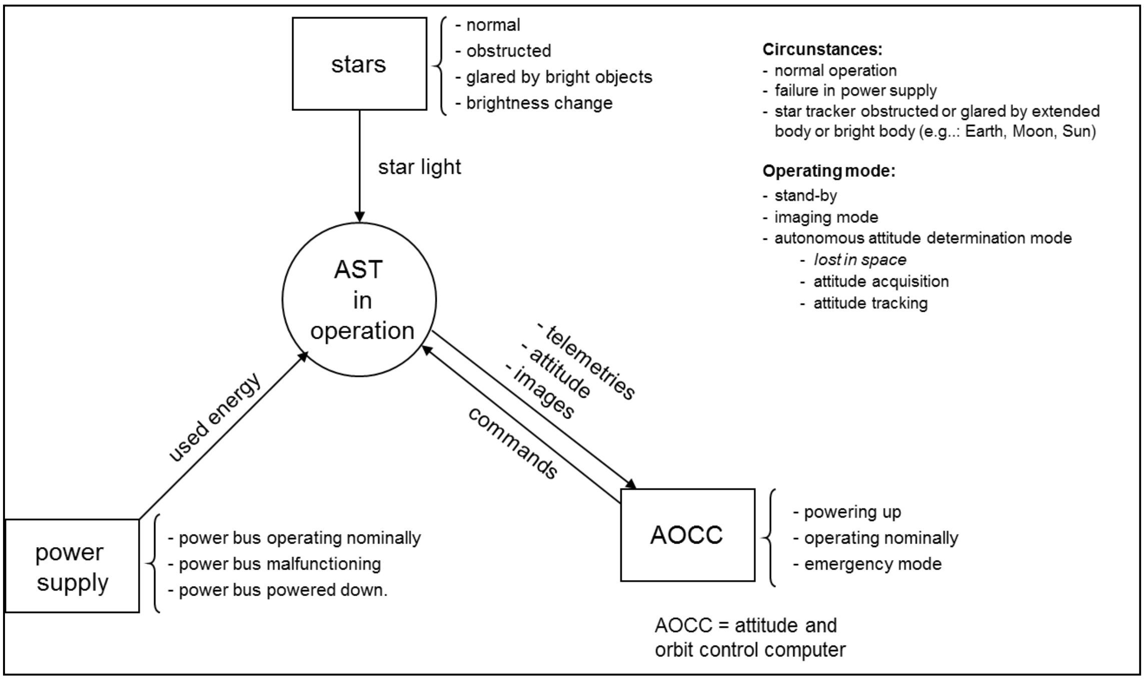

Figure 13 – Functional context diagram for the product during operation.

Figure 14 presents the external connections between the star tracker and the elements in its environment.

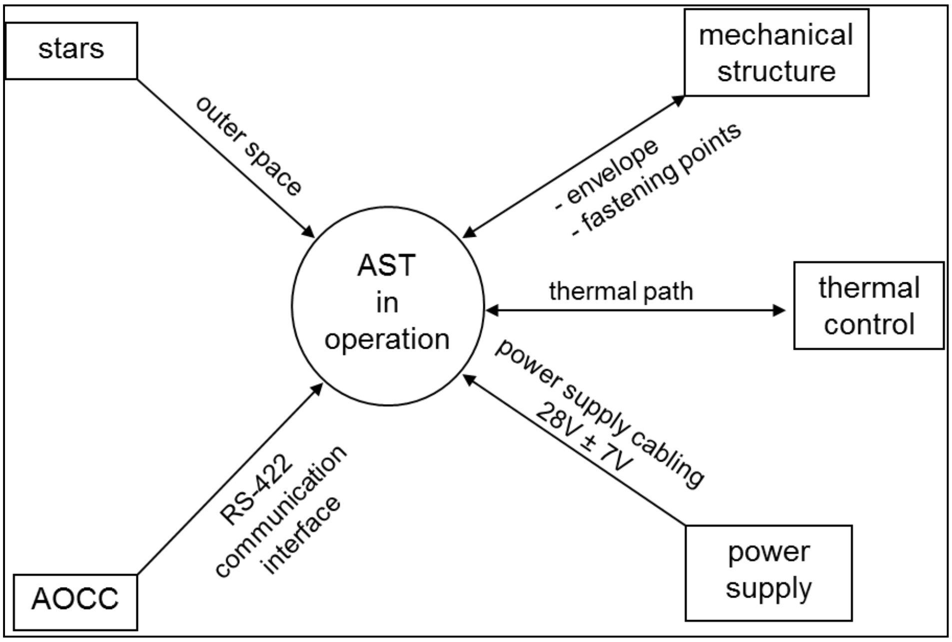

Figure 14 – Product architecture context during operation.

By knowing the product context and how it relates to its environment, a functional decomposition for the product can be performed, as shown in Figure 15. The same happens for the development organization (Figure 18).

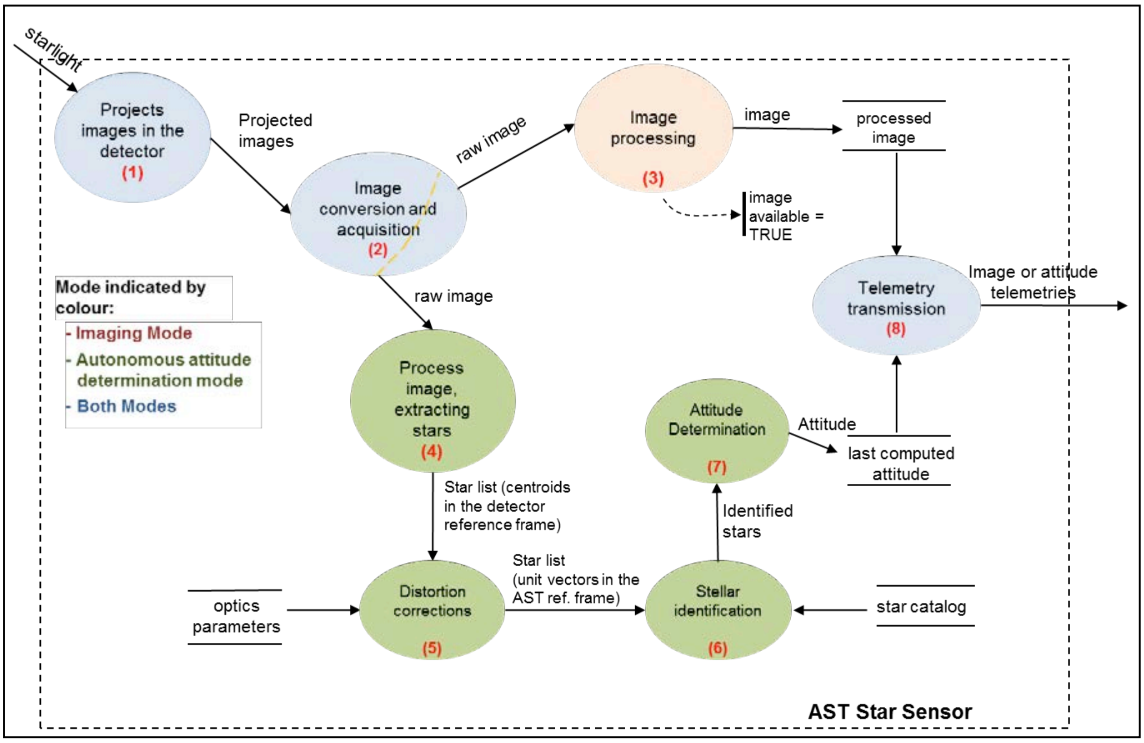

Figure 15 – Functional decomposition for the AST.

After the functional decomposition has been performed an architecture for the product can be proposed (Figure 16) and each function found in the functional decomposition can be associated with physical elements of the system (Figure 17). This association is expressed through an allocation matrix that relates physical parts and physical interfaces to the identified functions and functional flows.

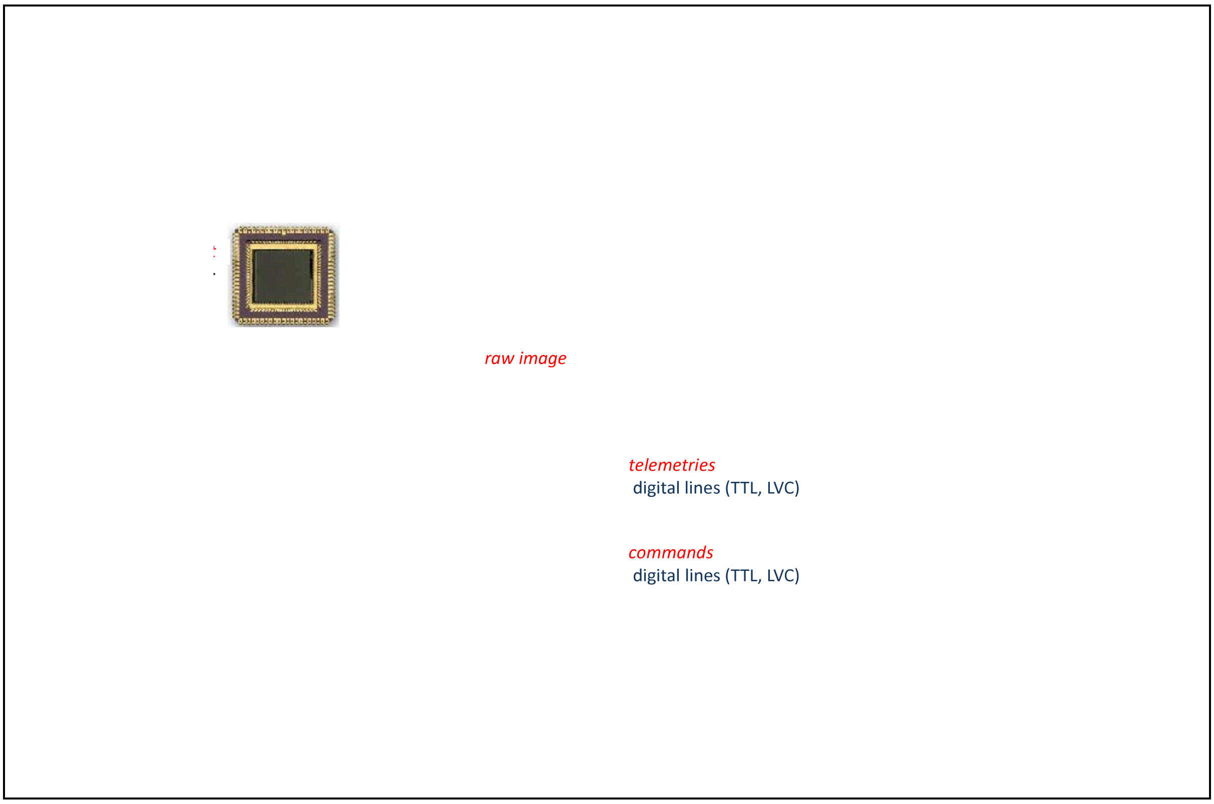

Figure 16 – Proposed architecture for the AST, showing information fluxes and physical interconnection mediums.

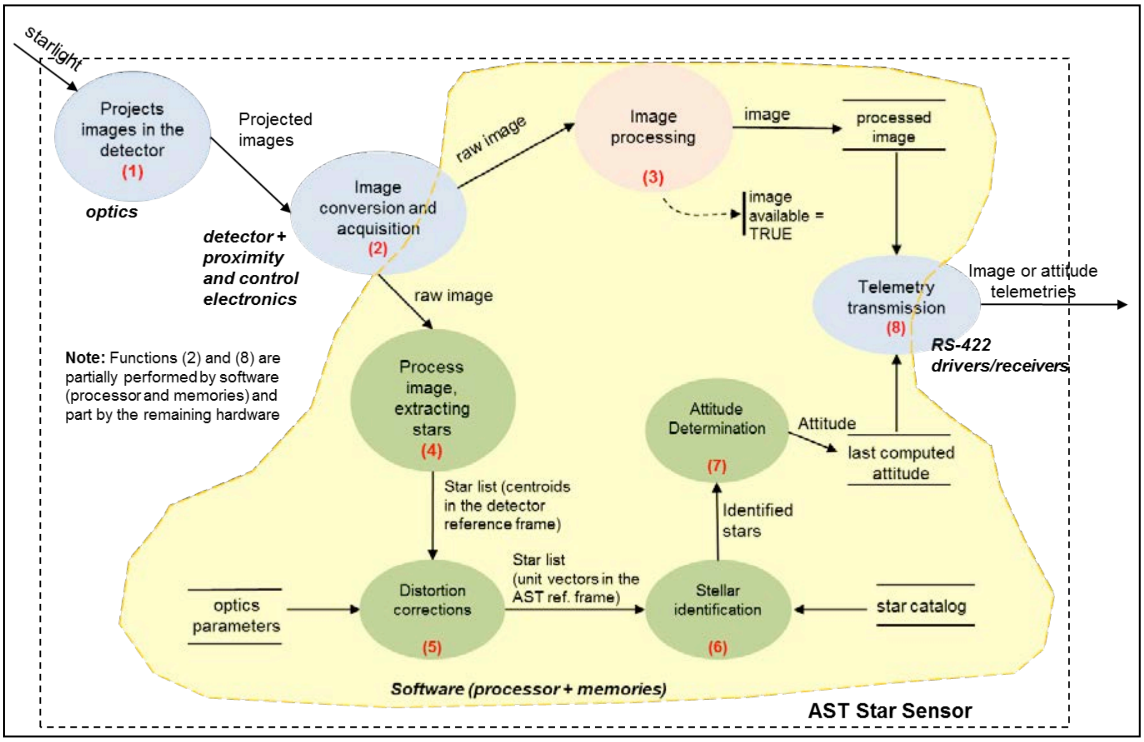

Figure 17 – Allocation of functions to physical elements for the AST.

The same steps done for the AST product can be performed for an organization, as shown in Figures 18 - 20.

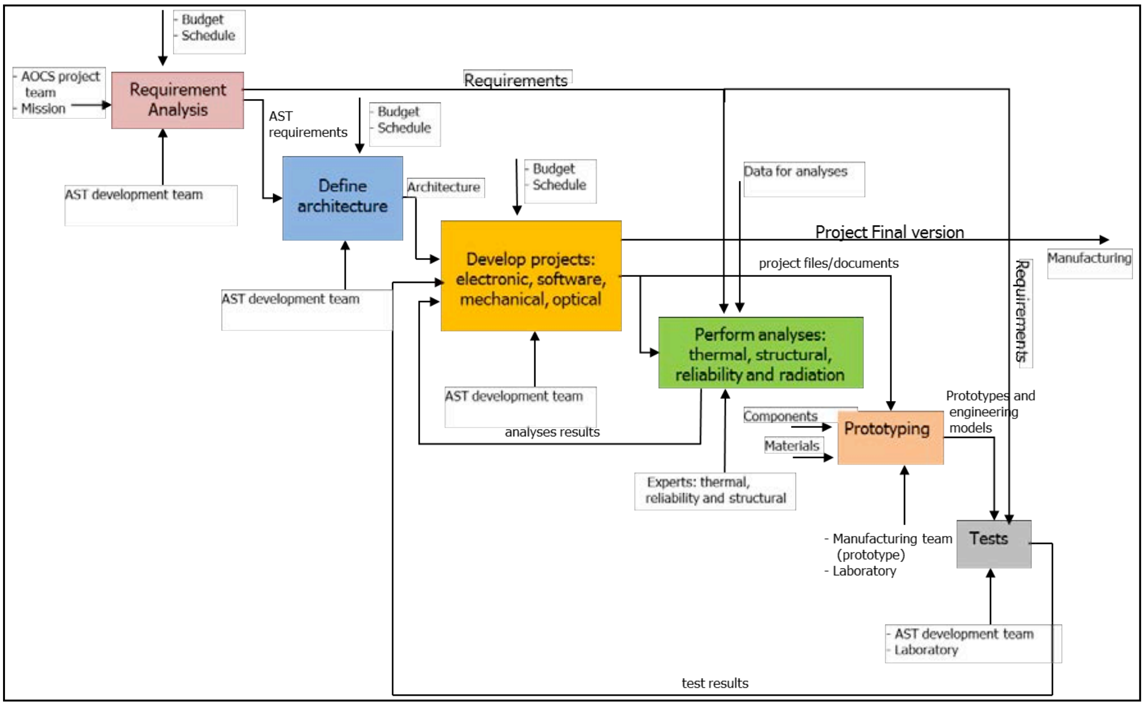

Figure 18 – Functional decomposition for the organization “development team”.

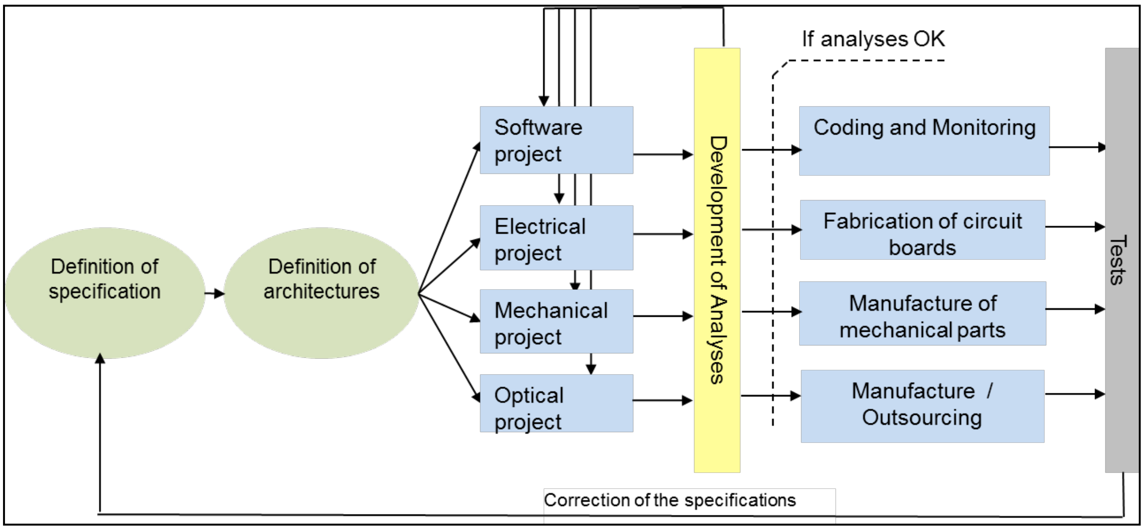

Figure 19 – Functional decomposition for the organization “development team” for the detailed design scenario in the development life cycle process.

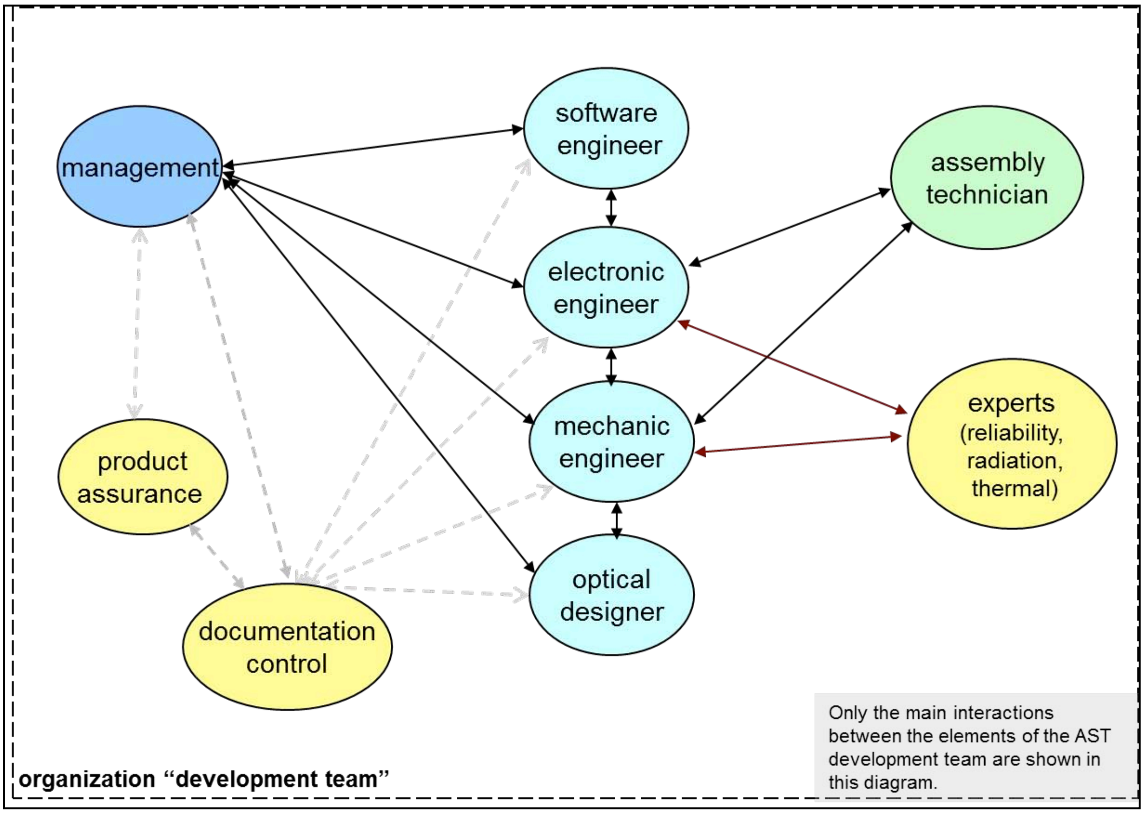

Figure 20 – development team architecture

5 - Discussion

The system concurrent engineering approach:

- considers, as the highest level of abstraction, the system's life not the system itself;
- acknowledges that the system solution comprises product and organization elements;
- performs stakeholder, requirements, functional and implementation analysis, simultaneously, for product and organization elements.

The traditional systems engineering approach:

- focuses on customer and users instead of using a general approach for taking into consideration a broader set of stakeholders;
- considers the product and its mission as the highest level of abstraction [11];
- focuses on developing the product elements of the solution;
- performs the systems engineering process for the product and for the development organization.

Traditional concurrent engineering approach:

- treats life cycle processes in isolation from one another [12];
- is applied at a very low level of abstraction;

Organizations that develop a product may also perform other product life cycle processes. In this case, the development of the organizations performing these life cycle processes is also within the scope of the development effort, Therefore, this approach allows for the integrated development of the product and the organizations performing its life cycle processes. For life cycle processes outside the scope of the development effort, the system concurrent engineering process is performed in order to derive requirements that must be accomplished by the product in order to contemplate life cycle process requirements.

Traditional systems engineering approach capture requirements from the product in order to be accomplished by the life cycle process performing organizations [13].

Taking into consideration, from the outset, the the life cycle process requirements, avoids late changes, and therefore, increases the productivity of such life cycle process performing organizations. It allows the life cycle process to be performed faster and cheaper for equivalent product quality.

6 – Conclusion

This paper used the example of a complex product development case, an autonomous star sensor, in order to present an application of the system concurrent approach to complex system development. The paper described the approach, compared and contrasted with traditional systems engineering approach and presented an example of application of the approach. The paper shows that concurrent development of product and organization elements of a system solution is possible with current and well established modelling notations. The paper qlso suggests that the concurrent architecting of product and organization elements may bring advantages in terms of early identification of potential interactions between product and organization attributes. Knowing these interactions in anticipation allows us to avoid late product changes, therefore increasing life cycle process productivity.

7 – Acknowledgement

The authors would like to thank the support of CAPES (Brazilian Coordination for Improving Superior Education Personell) for their support in sponsoring their participation in the 22nd International Sysmposium of the INCOSE.

References

[1] SIMBAD. Centre de Données astronomiques de Strasbourg. "SIMBAD Query Result: V* V2500 Oph -- Variable of BY Dra type". Retrieved March 14, 2012. http://simbad.u-strasbg.fr/simbad/sim-id?Ident=V*+V2500+Oph

[2] Loureiro G. A systems engineering and concurrent engineering framework for the integrated development of complex products. Ph.D. thesis, Loughborough University, 1999.

[3] Loureiro G. Lessons learned in 12 years of space systems concurrent engineering. 61st International Astronautical Congress, Prague, CZ. International Astronautical Federation, 2010.

[4] Loureiro G. et. al. System concurrent engineering of a mobile TT&C ground station for an unmanned aerial vehicle. Product (IGDP), v. 8, p. 1-10, 2010.

[5] Loureiro G. et. al. System concurrent engineering to develop a green car. New World Situation: New Directions in Concurrent Engineering, 2010, Part 7, 569-581, DOI: 10.1007/978-0-85729-024-3_53

[6] Loureiro G. et. al. Systems Concurrent Engineering of an Electrical Ground Support Equipment for an On-Board Computer. New World Situation: New Directions in Concurrent Engineering, 2010, Part 7, 513-526, DOI: 10.1007/978-0-85729-024-3_49

[7] Eectronic Industries Alliance. EIA-632: process for engineering a system. Arligton, 1997.

[8] National Aeronautics and Space Administration. NASA systems engineering handbook. Washington, 2007.

[9] Hitchins DK. Getting to grips with complexity. In: proceedings of the 2nd annual conference of the INCOSE – UK chapter, 1996.

[10]Alexander C. Notes on the synthesis of the form. Harvard university press. Cambridge, USA, 1964.

[11] European Cooperation for Space Standardization. ECSS-E-ST-10C: space engineering: system engineering general requirements. Noordwijk: ESA, 2009.

[12] Huang GQ. Design for x: concurrent engineering imperatives. London: Chapman Hall, 1996.

[13] Institute of Electrical and Electronics Engineers. Systems engineering: application and management of the systems engineering process. New York: IEEE, 2005.