"REALISATION, EXTENSION AND UTILISATION OF THE SYSTEM FACTORY OF SPACE SYSTEMS"

Carlos Redondo [(1)], Marina García [(1)], Tiago Jorge [(1)], Clément Goujon [(2)], Stephan Jahnke [(3)], Jean-Baptiste Bernaudin [(4)], Marcel Verhoef [(5)], Alberto González [(5)], Elena Alaña [(1)]

(1) GMV Aerospace and Defence, E-mail: carlos.redondo.aparicio@gmv.com, marina.garcia.d@gmv.com, tiago.jorge@gmv.com, ealana@gmv.com
(2) Thales Alenia Space Italy, E-mail: clement.goujon@thalesaleniaspace.com
(3) OHB System AG, E-mail: stephan.jahnke@ohb.de
(4) Airbus Defence and Space SAS, E-mail: jean-baptiste.bernaudin@airbus.com
(5) European Space Agency, E-mail: marcel.verhoef@esa.int, alberto.gonzalez.fernandez@esa.int

MBSE2022 Objectives: T-2, T-3.

Abstract: Model-Based System Engineering (MBSE) is an enabler of the digital continuity. Nevertheless, its deployment in space systems is not straightforward due to difficulty of the stakeholders to interact when using different technologies. One of the key elements that would ensure the seamless exchange of engineering data and facilitate this interoperability within the MBSE community is the definition of a Systems Engineering supporting infrastructure, called *System Factory*. This System Factory supports the System Engineers in executing the tasks described in the ECSS-E-ST-10 [1] standard when adopting MBSE. The specification of such a System Factory, following the ARCADIA method [2], was presented in MBSE2021 conference, including the results of Operational Analysis, System Need Analysis and Logical Architecture [3]. In this paper, we present several Physical Architectures of the System Factory as well as the refinements and extensions performed at both Logical and Physical levels, motivated by the results of various parallel ESA R&D activities, benchmarking against external initiatives (e.g. INCOSE), and addressing the notion of extended enterprise (data exchange across the supply chain). Finally, the utilisation of the System Factory is illustrated through one relevant System Engineering Use Case.

Keywords: Capella, Data Hub, Digital Continuity, Extended Enterprise, MBSE, Ontology, System Factory.

1. INTRODUCTION

The architecture of the System Factory is defined considering the main exchange scenarios produced during the space system development process within an organisation or company, including also the interface with external stakeholders (i.e. Local System Factory).

Its *Logical Architecture* represents the way engineering will be done when adopting MBSE independently of any underlying technology. It presents how the factory works (i.e. functionalities, exchanges) and the decomposition in its constituents' parts (i.e. logical components).

Although there is not a unique logical solution, the importance of this high-level abstract architecture lies in the fact that it has been designed to represent a reference point for all the stakeholders to implement their Physical Architectures. To achieve this, its specification considers and harmonises the usage scenarios and needs of the following Large System Integrators: *Airbus Defence and Space, Thales Alenia Space and OHB*. Recently, the model has been updated to also capture the physical architecture as used within ESA [4].

As the architecture at Logical level is independent from any technology or implementation, it allows the materialisation of different Physical Architectures depending on the implementation, technical and technological constraints and choices of each company.

Currently, three main areas are being analysed. These areas are presented in this paper:

1. **Realisation of the System Factory**: The Logical Architecture of the System Factory is realised by the Physical Architecture that represents a concrete physical solution based on existing tools. Several Physical Architectures have been modelled according to different views, which are presented in section 2.
2. **Extension of the System Factory**: Following an Agile process, the specification of the System Factory, especially at Logical and Physical levels, is incrementally improved through dedicated sprints in which issues are detected and implemented. The main activities performed towards the achievement of this objective include harmonising the semantics used, optimising the architecture by, for example, getting rid of functionality duplication, and consolidating the Physical Architectures. These activities are carried out considering new inputs from the involved Large System Integrators.

   Another source of extension of the System Factory comes from the integration of the results of MBSE applications in ESA projects and ESA R&D activities and external initiatives, with the aim of producing a fully aligned architecture. This integration effort has a major impact on the Logical and Physical levels. A table specifying these R&D activities, their integration status and their impact on the System Factory is provided in section 3.
3. **Utilisation of the System Factory**: The ultimate goal of the System Factory is to prove that it is useful and practical to be implemented in the development of space systems. To reach this goal, a use case is exercised based on the exchange of technical budgets (namely mass and power) within a company's Extended Enterprise (EE) considering the usage of the System Factory. Results of the first iteration of this activity are presented in section 4.

2. SYSTEM FACTORY REALISATION

Although the Logical Architecture of the System Factory provides a detailed overview of its main functionalities, and the roles involved, it is the Physical Architecture which details how these functionalities are eventually realised by a concrete set of Physical Components, which in the context of the System Factory represent the required Systems Engineering toolset. These Physical Components are traced back to the Logical Architecture at leaf Logical Component level through a *realisation* link.

While the Logical Architecture of the System Factory is unique and relatively stable over time, the Physical Architecture is organization-dependent and expected to be much more dynamic as new tools and features become available, enabling users to digitalise more and more their systems engineering processes. A total of six Physical Architectures have been produced, which can be grouped per specific viewpoints, defined in the following subsections.

2.1. Organization-driven Physical Architecture

Within this category, a total of four Physical Architectures have been modelled: one for each of the LSIs involved in the study (*Airbus, Thales Alenia Space and OHB)* and one for ESA.

They have been captured through Physical Architecture Blank (PAB) diagrams, identifying not only the tools that constitute each Physical Architecture but also their exchanges. A dedicated PAB with the classification of all the tools contained in these four PABs has also been created, grouping them by nature (e.g. MBSE and Architectural Tools, Requirements Tools, System Analysis and Simulation Tools…).

These architectures show the adoption of MBSE in each organisation, i.e. the organisations' MBSE infrastructure. The capability to export model elements and metrics related to them enable to set indicators on adoption of MBSE tools or resistance of non-MBSE tools to change plans.

2.2. Ecosystem-driven Physical Architecture: Extended Enterprise

A novel concept that has been introduced to the System Factory as part of its extension is that of *Extended Enterprise*, which SEBoK[1] defines as *"wider organization representing all associated entities - customers, employees, suppliers, distributors, etc. - who directly or indirectly, formally or informally, collaborate in the design, development, production, and delivery of a product to the end user*".

In particular, the Extended Enterprise considered is that of ESA, which contracts LSIs for the development of space systems. Equipment suppliers (i.e. one level lower than the LSIs) have not been considered yet in the current model.

[1] https://www.sebokwiki.org/wiki/Extended_Enterprise_(glossary)

The resulting Physical Architecture depicts the main exchanges between ESA and the LSIs, as well the tools that make these exchanges effective. This analysis is based on multiple space missions extracted from [4].

2.3. Maturity-driven Physical Architecture

Another viewpoint that has resulted of interest for the depiction of the Physical Architecture is that related to the maturity of the tools employed. While the organization-driven Physical Architecture classifies the tools per organization, and the ecosystem-driven contextualises it within an organization's environment, the maturity-driven Physical Architecture adds an extra layer of information related to the maturity of some of the tools that build up the Physical Architecture.

For this purpose, an R&D Physical Architecture PAB has been created, incorporating tools that are currently under development, as well as their exchanges with already identified tools. The metric selected for describing the tools maturity is the Technology Readiness Level (TRL). Note that only tools in development (referred to as R&D tools) are to be assigned with a TRL value, excluding commercial tools as the maturity of these is not of interest and already well known by the MBSE community.

Examples of tools included in this R&D PAB include TeePee [5] and GSEF (Ground Segment Engineering Framework) [6]. Tools that have been found to exchange information with them have also been captured in the PAB, although not assigned with a TRL as explained above. New tools may be added when more R&D activities are analysed and their results effectively integrated (see section 3).

3. SYSTEM FACTORY EXTENSION

One of the main sources identified as driving the need for the extension of the System Factory is the consideration of parallel ESA R&D activities. To ensure the maximum possible alignment between these activities and the System Factory, their results are being integrated progressively as they become available. Note that some of these activities are currently on-going, and therefore their integration is expected to happen in the near future. Having set the context and the need for the consideration of these activities, Table 1 depicts 1) the considered R&D projects, 2) their integration status within the System Factory and 3) their main impact on the architecture.

Table 1: R&D activities analysed

R&D Project	Integration Status	Impact
MODEX	On-going	Logical Architecture
TeePee4Space	On-going	Logical & Physical Architectures
Others (see §3.3)	Not started	Not analysed yet

Next subsections summarise these R&D activities and the results integrated in the System Factory.

3.1. MODEX

In the MODEX (Model Exchange for Software Engineering) study [7] a detailed analysis on the ECSS-E-ST-40C standard [8] and OSRA [9] processes is performed, putting the focus on the data and models (so-called *Work Products* or *Exchange Items*) produced internally but more importantly externally to the core SW development process, the roles that produce them, the exchanges between those roles and the supporting data exchange formats. As a result, the study describes the implementation of the software development process when a model-based approach is adopted, in relationship with the ECSS-E-ST-40C standard. The outcome of this activity was a preliminary Software Factory Functional Architecture modelled in Capella, which focused on the identification of the main interfaces with the System Factory, considered as a black box. These interfaces are the target of the MODEX activity integration, putting the focus on the System and Software Factories exchanges.

A dedicated folder within the System Factory Logical Architecture is created for the modelling of the Software Factory at Logical level, transitioning from the functional one. System/Software Factories interfaces have been captured through dedicated Logical Architecture Blank (LAB) diagrams, as shown in Figure 1.

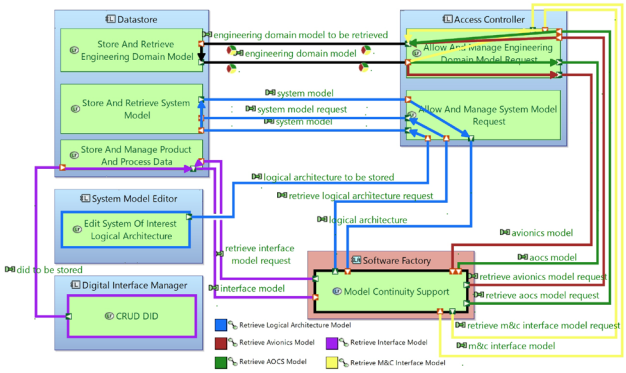

Figure 1: Interface between the System Factory and the Software Factory functionality "Model Continuity Support"

This analysis enables the identification of interface gaps between the Software and System Factories. Such gaps apply to Software Factory functionalities *Code Development, Integration and Build Support* and *V&V Support*, whose associated exchange items are not currently supported by any existing System Factory functionality, as these refer to code-like artefacts (e.g. *OBSW binary*, *Component Code, Execution Platform Code*). The extension of the System Factory Logical Architecture driven by these gaps is precisely the target of this analysis and is currently on-going task.

3.2. TEEPEE4SPACE

TeePee4Space project [5] was performed as part of the OSIP Model-Based System Engineering Campaign as an early technology development. The goal of this project is to perform several analyses related to the physical architecture of the system of interest (e.g. Product Breakdown Structure, Mass Analysis, Power Consumption Analysis, etc.) on a heterogeneous and distributed set of models (e.g. COMET, Cameo, Capella, Excel…) showing the benefits of digital continuity at an extended enterprise level to systems engineers. For this end TeePee tool has been developed, being a platform which handles heterogeneous models performing these analyses based on the imported models data.

Three activities are identified which fall within the scope of the System Factory. They are currently on-going:

1. Validation of the System Factory architecture for TeePee4Space particular use cases by augmenting the scope of both Customer and Payload Supplier roles, identifying potential shortcuts and gaps in the current System Factory Logical Architecture. These use cases refer to embed technical budgets exchanges (namely mass and power) within the Extended Enterprise in the System Factory Logical Architecture. Dedicated diagrams at both Logical and Physical levels will be created for this purpose. A first iteration of this activity is presented in section 4 as a practical example of how the System Factory can be used and exploited for a specific and relevant systems engineering use case.
2. Addition of the TeePee tool in the System Factory Physical Architecture. As introduced in section 2.3, an R&D Physical Architecture PAB has been created, which precisely includes TeePee tool as well as additional tools which exchange information with it as per TeePee4Space defined use cases (namely COMET, Cameo and Excel). The main goal of this activity is derived from section 2.3 in which R&D tools are to be monitored and captured in a specific PAB, being TeePee tool the initiator of this need.
3. Impact of the addition of TeePee tool in the Gap Analysis performed in [10] between the current systems engineering state-of-the-art toolset and the System Factory Logical Architecture. The main goal of this activity is to identify if any of the highlighted gaps would be covered with the implementation and deployment of TeePee in the development of space systems. This specific activity has not yet been started and therefore no further reference has been made in the present paper.

3.3. OTHER R&D ACTIVITIES

While the work for the integration of both MODEX and TeePee4Space projects results have already started, showcasing in this paper some preliminary results, integration of other projects results is pending and scheduled for the near future as they are materialised and made available. We expect to present a more mature model at the time of the MBSE 2022 workshop.

These include Model-Based Engineering Hub (MBEH), Space System Ontology (SSO), Model-Based System Engineering for AIV (MBSE4AIV), Product Assurance as a Service (PASaaS) and Digital Engineering Hub Pathfinder (DEHP) activities.

4. SYSTEM FACTORY UTILISATION

4.1. EXTENDED ENTERPRISE SCOPE

One of the key objectives of the System Factory and its associated Logical Architecture is to be used and considered as a reference for organizations and tool vendors across the supply chain aiming at digitalising their Systems Engineering processes, not limiting its use to ESA or LSIs but also to Equipment Supplier organisations, among others, as they are all elements of the Extended Enterprise involved in the development of Space Systems.

This reutilisation objective sets the scope for the System Factory in the sense that it shall be possible to reuse it regardless of the organisation nature, be it ESA, an LSI or a lower-level supplier organisation. Therefore, the System Factory is based on a three level environment: Customer, System of Interest Contractor and Supplier.

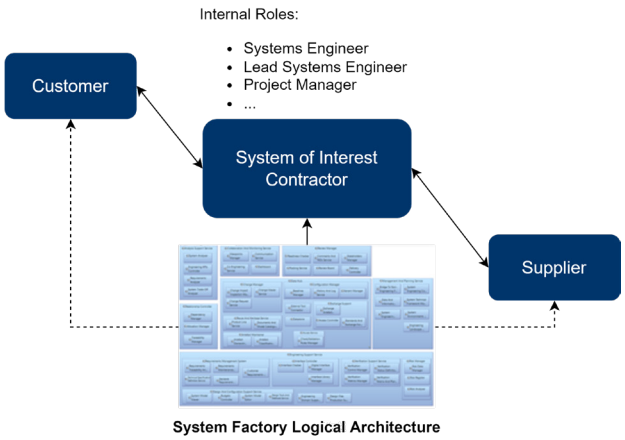

Figure 2: System Factory scope within the Extended Enterprise

The System Factory is then centred at the System of Interest Contractor level, with external roles being the Customer and the Supplier. Note that the System of Interest Contractor could be any organisation involved in the development of a Space System along the supply chain, which fulfils the objective initially set allowing *inserting* the System Factory Logical Architecture at any level of the Supply Chain based on the Supplier ↔ System of Interest Contractor ↔ Supplier environment.

4.2. TEEPEE4SPACE: A SYSTEMS ENGINEERING USE CASE

One of the identified activities showcasing a high synergy between TeePee4Space project and the System Factory is the embedding of the use cases described in the former into the architecture of the latter. This activity totally aligns with one of the main objectives of the System Factory, which is to prove itself useful when facing the development of space systems by prescribing which methods and tools are available for the performance of a given engineering process. In particular, the selected use case is the exchange of technical budgets (mass / power) within a system integrator organisation which implements the System Factory. The embedding of this use case takes advantage of the Logical Architecture. However, a dedicated Logical Architecture Blank diagram is created for the sake of clarity to identify the functionalities required by the use case – see Figure 3.

Figure 3 depicts a subset of the System Factory Logical Architecture showing how external (i.e. Customer and Payload Supplier) and internal (i.e. Systems Engineer) roles interact with the System Contractor System Factory. A total of six functional chains related to the TeePee4Space use case have been identified and listed on the bottom-right side of the diagram following a colour code scheme:

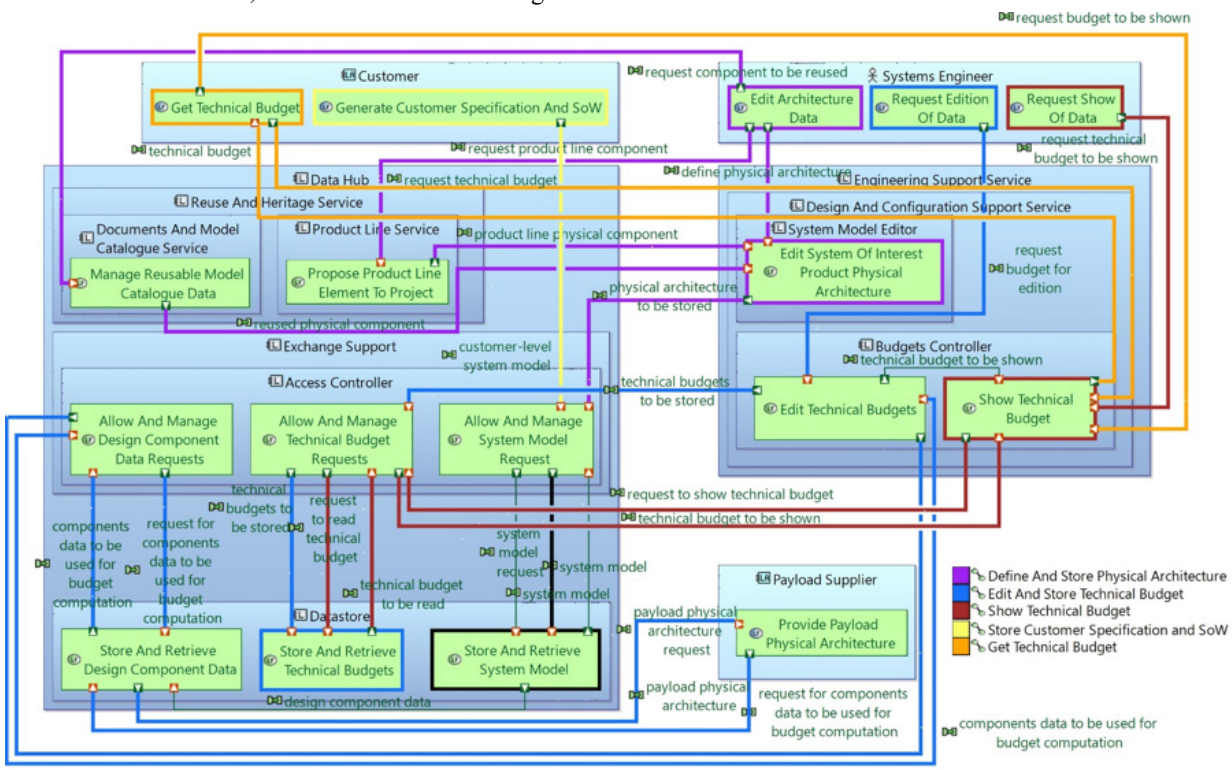

Figure 3: First iteration of the TeePee4Space use case integration within the System Factory Logical Architecture

- **Define and Store Physical Architecture**: Illustrates how the Systems Engineer makes use of the System Factory to define and maintain the system of interest physical architecture.
- **Edit and Store Technical Budget**: Illustrates how the Systems Engineer makes use of the System Factory to build up a technical budget based on selected physical components, including those to be provided by the Payload Supplier, and stores it.
- **Show Technical Budget**: Illustrates how the Systems Engineer makes use of the System Factory to request that an already stored technical budget is shown.
- **Store Customer Specification and SoW**: Illustrates how the System Factory manages the reception and storage of specifications and statement of work from the Customer organisation.
- **Get Technical Budget**: Illustrates how the System Factory handles a request from the Customer organisation to be shown a technical budget and to import it.

A dedicated Physical Architecture Blank diagram is created with Physical Components that realise the Logical Components required in the use case. A Physical Component *realizes* one or more Logical Components, and a Logical Component is *realized by* one or more Physical Components.

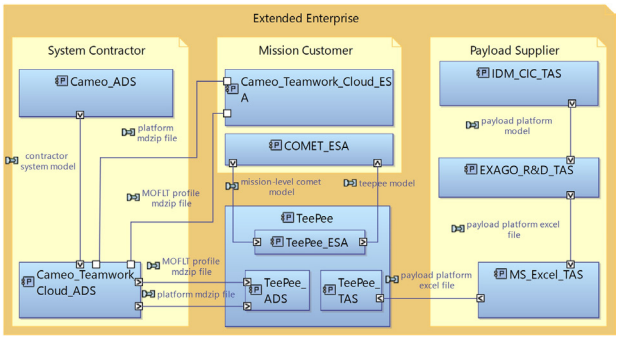

Figure 4: First iteration of the TeePee4Space use case integration within the System Factory Physical Architecture

Table 2 shows the traceability between the Logical and Physical Architectures used to depict TeePee4Space use case. Note that only Physical Components implemented by the System Contractor are listed since the System Factory Logical Architecture depicted in Figure 2 is centered on the System Contractor organization.

Table 2: Realization links between the Logical and Physical Architecture for TeePee4Space use case

Logical Component	Physical Component
Documents and Model Catalogue Service	Cameo Systems Modeler
Product Line Service	Cameo Systems Modeler
Access Controller	Cameo Teamwork Cloud
Datastore	Cameo Teamwork Cloud
System Model Editor	Cameo Systems Modeler
Budgets Controller	TeePee

Similar tables to Table 2 could be derived for the Mission Customer and Payload Supplier organisations and the corresponding System Factory subset that they would use for the derivation of their deliverables (e.g. how the Mission Customer uses the System Factory to ultimately generate a Mission-level model, and how the Payload Supplier makes use of the System Factory to ultimately provide the Payload Specification). Similar Logical Architecture Blank diagrams to that displayed in Figure 3 could be elaborated since as explained in section 4.1 the System Factory is centred on the System of Interest Contractor and this can be any organisation along the supply chain.

5. CONCLUSION

The Logical Architecture of the System Factory shall serve as a reference to the MBSE community to implement their Physical Architectures and enable interoperability in the development of space systems as illustrated in the TeePee4Space use case.

The Logical level provides information on which are the exchanges that take place, the functionalities involved and their allocation, whereas the representation at Physical level illustrates how the Logical Architecture is implemented in an organisation. Both architectures are evolving, e.g. they shall consider results from ESA R&D activities, the Physical Architectures depend on available MBSE tools, etc. As part of this continuous improvement process, stakeholders are encouraged to provide feedback on the system factory definition.

The System Factory model is freely available in the ESA's European Space Software Repository [11]. HTML documentation is also provided so that installation and use of Capella is not ultimately needed.

6. REFERENCES

[1] "Space Engineering - Space engineering general requirements", ECSS-E-ST-10C Rev.1.

[2] ARCADIA: https://www.eclipse.org/capella/arcadia.html"

[3] "Specification and architecture of a System Factory for space systems", MBSE 2021.

[4] "MBSE at ESA: State of MBSE in ESA Missions and Activities", Jamie Whitehouse on behalf of ESA TEC-S/MBSE, MBSE2021 Conference.

[5] "TeePee4Space Final Report", V076L00T00-008, V1.

[6] "ADGE – Advanced Digital Ground Segment Engineering – Statement of Work".

[7] "Relationship Between Process And Software Factory", MODEX study, GMV 22788/19 V6/20.

[8] "Space engineering – Software", ECSS-E-ST-40C.

[9] "OSRA - Onboard Software Reference Architecture", https://essr.esa.int/project/osra-onboard-software-reference-architecture.

[10] "Consolidated Gap Analysis", SASyF study, GMV 23459/21V4/22.

[11] ESSR: https://essr.esa.int/project/specification-and-architecture-of-a-system-factory-sasyf

7. ACKNOWLEDEMENTS

This work was supported by the "SASyF Utilisation for ESA Infrastructure" project, ESA contract 4000136894/21/NL/AS/kk. GMV acts as prime contractor in this activity with Thales Alenia Space Italy, Airbus Defence and Space SAS and OHB System AG as subcontractors.