From Model Based Systems Engineering to Model Based System Realization: Role and Relevance of IVTV Plan

Vincent Chapurlat, Eric Bonjour

From Model Based Systems Engineering to Model Based System Realization: role and relevance of IVTV Plan

V.Chapurlat[1], E.Bonjour[2]

1) Ecole des Mines d'Alès - LGI2P - Parc Scientifique G.Besse, 30035 Nîmes cedex 1

Vincent.Chapurlat@mines-ales.fr

2) ENSGSI - 8 rue Bastien Lepage BP647, 54010 Nancy Cedex

Eric.Bonjour@u-lorraine.fr

Abstract. The IVTV Plan (Integration, Verification, Transition and Validation of the system before its Qualification) is developed and validated during the design stage. It details all the activities, resources, requirements, means, etc. requested during the realization stage so it is the hyphen between these two crucial stages in system life cycle. It is today necessary to help companies to better transfer detailed design models towards realization for many reasons discussed in this paper. Mainly, IVTV plan remains difficult to be exploited. This article proposes a first step towards a Model-Based Realization Plan, that is, a meta-model that represents the links between models that comes from Model-Based System Engineering and information required in the IVTV plan.

Keywords. System Engineering, System Design, System Realization, Integration, Verification, Transition, Validation, Plan, meta model

1 Introduction

Systems Engineering (SE) [1][2][3][4][5] is an engineering approach covering the whole life cycle of a system as schematized in Figure 1 and considered as a model based approach [6] *e.g.* requirements, functional, physical, operational scenarios, or configuration models.

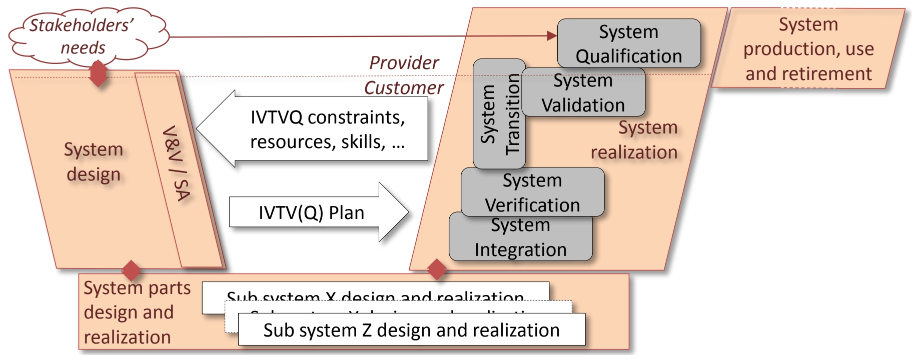
Stakeholders' needs
Provider
Customer
System Qualification
System production, use and retirement
System Validation
System Transition
System realization
IVTVQ constraints, resources, skills, …
System design
V&V / SA
IVTV(Q) Plan
System Verification
System Integration
System parts design and realization
Sub system X design and realization
Sub system Z design and realization

Figure 1: From Design to Realization, IVTV plan role and position

The IVTV Plan (Integration, Verification, Transition, and Validation prior to the Qualification of the system not considered here) is developed and validated during the design stage. It details all the IVTV activities, resources, requirements, means, etc. requested for the realization stage of a system as expected by all stakeholders. So, this plan operates belt transmission between teams, activities and processes concerning system design and system realization. It gathers the information necessary 1) to describe the subsystems and components that must be integrated, 2) how to proceed to assemble these elements to get the 'right' system and to converge step by step and in confidence towards the 'right' then the ‘good’ system, and 3) predict and anticipate risks and shortcomings inherent in achieving integration e.g. by defining and evaluating possible alternatives. In the mind as in the practices of design and realization team members, this plan is still often present in the form of documents generally prepared from templates facilitating, writing as reading and interpreting the plan. In this case there is no real continuum of models from the upstream design activities and IVTV activities. Causes of this rupture are multiple. First the type and nature of the expected product (single exemplar, for a small or medium to large series, software-intensive system, technical / socio-technical, etc.), the culture and practices of the company in charge of all or part of the realization (on site or in factory), etc. can be of course considered. Second design models are built by using various Design Specific Modelling Languages (DSML). These ones are generally defined by meta models highlighting at least SE core concepts and relations e.g. requirement, function, component and interface [7]. However, concepts and relations requested for the elaboration of IVTV plan are generally insufficiently detailed and linked with these SE core concepts [8][9][10]. Third, some of design models, even if they have to be adapted or transformed prior to any use, can be useful for facilitating work and assuming the relation between design and realization stages e.g. allowing integration team members to share test bench results having to be associated to a given set of requirements defined by design team members. Last, it should also be noted the significant lack of tools to use wisely the models mentioned above, or adapt / change without loss or effort or undue delays so that they become truly useful and usable. This paper aims to propose an IVTV Plan meta model to link more closely design and realization activities by 1) irrigating the latter with models issued from the former, eventually by using model transformation rules and techniques, 2) facilitating the sharing of information between the two stages, and 3) supporting IVTV project preparation and management dependently from the defined and validated plan.

2 IVTV Plan meta model

2.1 IVTV definitions and needs

Processes promoted by Systems Engineering standards [1] or reference document [11] give details about the activities to be done all along the system life-cycle. The position, the role and the relevance of IVTV Plan (detailed in [3]) is discussed below for facilitating the interactions between these main processes as illustrated in Figure 2 and then for reaching the proposed objectives.

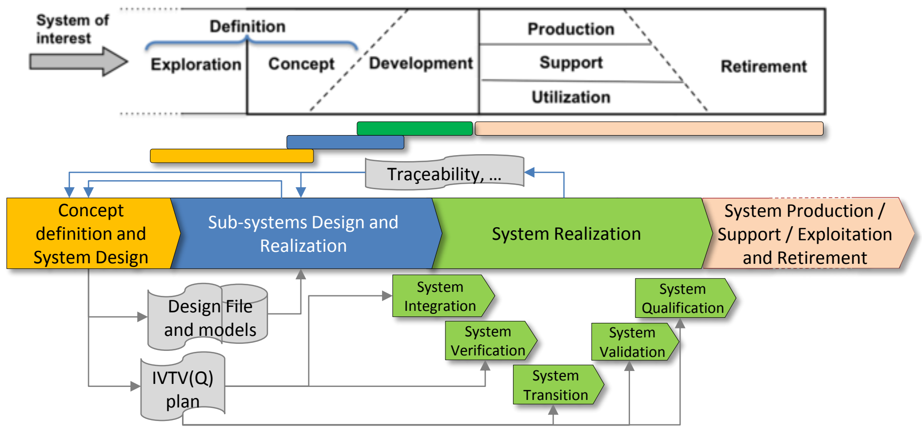

Figure 2: System life-cycle and SE processes: IVTV plan position

We adopt the following definitions:

• **Integration**: [1] define integration process as *a process that combines system elements (implemented elements) to form complete or partial system configurations in order to create a product specified in the system requirements*. Its purpose is to prepare the system of interest (SOI) for final validation and transition either for use or for production. Integration consists of progressively taking delivery from implemented elements (*i.e.* components and sus-systems of the SOI), assembling theses ones as architected during design, and to check correctness of static and dynamic aspects of interfaces between them.

- **Verification:** [1] defines the verification as the *confirmation, through the provision of objective evidence, that specified (system) requirements have been fulfilled.* The verification process aims to ensure the system has been built correctly and ready to be validated with an acceptable level of risk.
- **Transition**: [3] defines transition process as the process *putting into operation the verified system with its useful enabling systems to demonstrate its ability to provide operational services expected by stakeholders*.
- **Validation:** [1] defines the validation as the *confirmation, through the provision of objective evidence, that the stakeholders' requirements for a specific intended use or application have been fulfilled*. The validation process aims to ensure *the system satisfies the customer and user needs as stated and agreed* [11] and ready to be qualified or exploited.

Various needs have to be taken into consideration when performing IVTV processes. Let us first mention needs common to all processes:

- Define a global IVTV strategy as soon as possible during design phase considering IVTV constraints from teams, and subsequent possible alternatives of associated plans (verification, validation, integration and transition),
- Define and schedule activities to be carried out considering system and stakeholders' requirements, and detailing tasks and operations within these activities,
- Define, forecast, reserve, prioritize, optimize and plan requested resources use: human resources (availability, level of skills, need training or employment, etc.), material resources (test benches, tools, etc.), organizational resources (rules and policies, procedures, agreements, technical documents, etc.), methodological (inspection, demonstration, simulation, test) and software resources to ensure the activities, their management and logistics,
- Manage risk (technical, human, organisation, environmental, etc.),
- Define metrics, indicators (management as technical) *e.g.* TRL/IRL and effective measures *e.g.* to evaluate risky situations occurrences or risk level,
- Manage interfaces (technical, logical, organisational) from the system, sub systems and components,
- Organise and manage traceability (percentage of detected defaults, teams workload, anomalies, requested modifications or evolutions of the product, etc.), return of experiment and of data reference models,
- Estimate costs of each activity taking into account various metrics *e.g.* related to the set of requirements,
- Manage requirements, constraints (normative, reference, linked to the contract with customer, etc.),
- Set up configurations of the target product under test and of its environment (reference configuration, configurations reachable, dysfunction configurations, etc.),
- Manage contributors and enabling systems IVTV along with target product IVTV,
- Dispose of tools supporting plan model building, checking, assessing in order to reach a consensus between team members against proposed plan, alternatives, etc.

More specifically, integration requires 1) to assume that delivered components are a) delivered in time or can be emulated by other and equivalent components in case of delay, and b) to respect the requirements in particular in terms of interfaces, and 2) to evaluate step by step the behaviour of the resulting assembly by applying various techniques. Verification requires 1) to apply verification techniques (test, audit, etc.), tools (simulators, emulators, test bench, etc.) or methodologies relevant for justifying and demonstrating requirements (functional as non-functional) defined in design phase are fulfilled, and 2) to trace the verification results even incomplete. Transition requires 1) to have available the operational environment *i.e.* validated enabling systems, associated documents and procedures, 2) to be able to put final users in training situations, and 3) to provide solutions in response to specific expectations from the customer. Finally validation requires 1) to have and to be able to use all deliverables, facts, tooled environments, etc. coming from previous activities (integration, verification and transition) and 2) to dispose of a validation environment corresponding as much as possible to the operational environment in which the future system has to work providing services and evolving.

Information (requirements, activities schedule, means, skills, resources, etc.) requested to cover these needs are defined in design stage gathered in the IVTV plan defined too during this stage. However, this plan is elaborated as a (set of) document(s) more or less easy to write, to understand in time and to interpret without ambiguities and sometimes huge efforts. The IVTV Plan Meta Model (IVTV PMM) presented below formalizes, merges and makes available in a coherent manner various concepts and relations requested and handled by the four concerned domains: design (SE core concepts), project management, risk management and IVTV (extensible to Qualification) concepts.

2.2 IVTV PMM

This meta model is conform to EMF notation and the Ecore metamodel[1]. The core SE concepts retained here are the function, component, interface, requirement and operational scenario (or use case) represented in Figure 3.

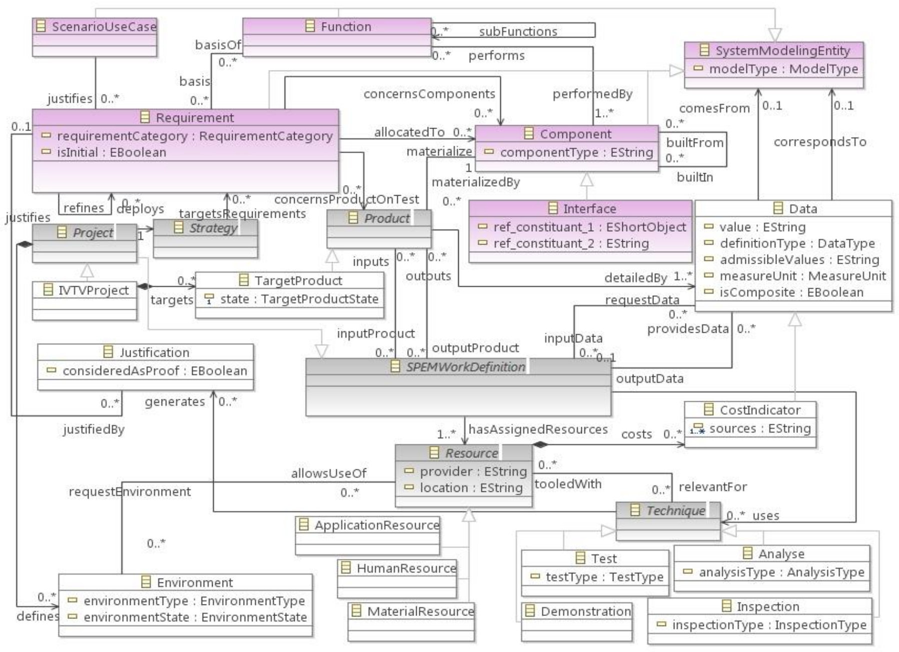

Figure 3: Merging core SE, project management and IVTV concepts (partial view)

[1] Eclipse Modeling Framework, available online at: http://www.eclipse.org/modeling/emf/

In a second way, the SPEM standard [12] proposed by OMG is used to describe project, process, activity, resource and other concepts and relations related to project management domain. A risk model inspired from [13] is used for covering risks management domain (technical, financial as managerial). Last, IVTV concepts and relations are defined taking into account the needs listed before *e.g.* IVTV strategy, product, technique, result, or report as schematized briefly in Figure 4.
Once merged, the result is a set of 77 concepts fully interoperable with various existing SE, project and risks management principles and even tools. It is then permit to progress step by step, in confidence during design stage and taking into account in a common approach design maturity level, project feasibility and risk evidence when performing the next activities:

- To define the IVTV strategy and the various alternatives of IVTV plans, preparation, determination and scheduling of activities, retained resources, needed enabling systems to design and realize, risk level and possible impacts and vulnerability of such plans or resources, etc. Teams' members can then share and dispose of all expected data, information and knowledge about system of interest, project, resources profiles and availability, etc. [14][15].

To check consistency and conformity but also relevance of the modelled plans by using appropriate techniques *e.g.* [16][17] and by modelling and considering global constraints, best practices, rules of thumb or policies.

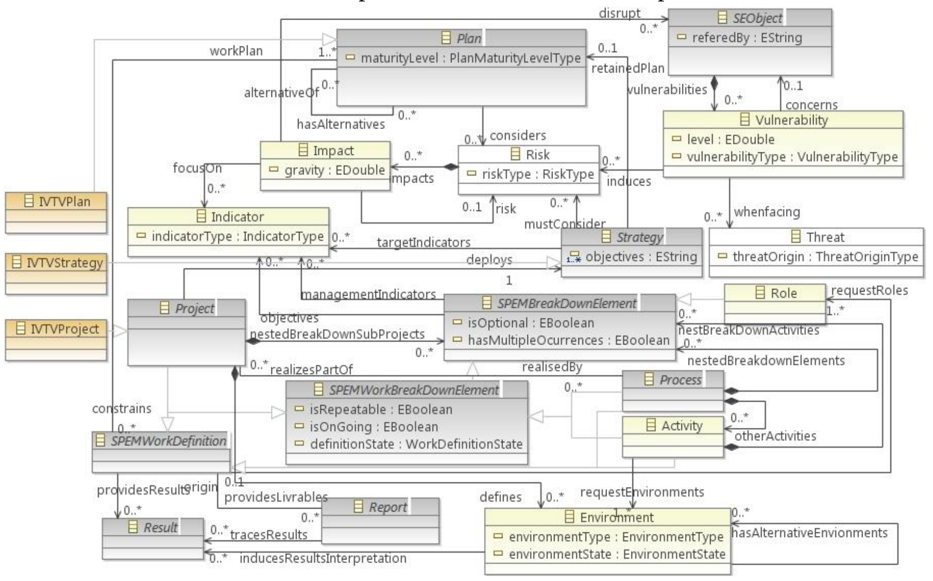

Figure 4: Merging core Risk management and IVTV concepts (partial view)

- To simulate and to assess IVTV plans alternatives in order to compare them, then to optimise and facilitate validation of the final IVTV plan to be performed.

- To manage execution of this plan and to share in time information resulting from IVTV activities without ambiguities or doubts because reported directly in design system models.

3 Conclusion and perspectives

The IVTV PMM presented in the previous part forms the basis of a new DSML for Systems Engineering assuming a part of the expected continuum of models between design and realization stages. It is possible to talk about Model Based Realization System principles. For this, at least another contribution is now expected.
Indeed, it is necessary to conceptualize and develop (with a great attention to DSML interoperability problematic *i.e.* to be and stay conform and compliant in order to become able to check the consistency of resulting models at least) a tooled approach supporting design model transformation in order to extract from these models specific business models used for supporting or facilitating realization as it is proposed for instance [18] in the case of transforming SysML models [19] in MODELICA [20]. This is one of the related works currently under development.

4 Acknowledgements

The authors wish to thank the members of the Working Group IVVQ of the Technical Committee Safety, Security and Validation of Systems (SV2S) of Association Française d'Ingénierie Système (AFIS) for the various opportunities to discuss advices and ways of working.

5 References

1. ISO/IEC 2008, IEEE Standards 15288.2008 – Systems engineering – System life cycle processes (2nd edition), February 2008
2. INCOSE, System Engineering (SE) Handbook Working Group, System Engineering Handbook, A Guide For System Life Cycle Processes And Activities Version 3.2.1, INCOSE TP 2003 002 03.2., 2011
3. DCIS, Découvrir et comprendre l'Ingénierie Système, Collection AFIS sous la direction de Serge Fiorèse, Jean-Pierre Meinadier, CEPADUES Editions, ISBN : 97802036493.005.6 , Avril 2012 [in French]
4. Blanchard, B.S., and W.J. Fabrycky. 2011. Systems Engineering and Analysis, 5th ed. Prentice-Hall International series in Industrial and Systems Engineering. Englewood Cliffs, NJ, USA: Prentice-Hall.
5. Faisandier, A. 2012. Systems Architecture and Design. Belberaud, France: Sinergy'Com.
6. INCOSE, Survey of Model-Based Systems Engineering (MBSE) Methodologies, Model Based Systems Engineering (MBSE) Initiative from International Council on Systems Engineering (INCOSE), 10 June 2008

7. CORE V6.0, Vitech corporation, http://www.vitechcorp.com/products/index.html, 2010
8. N.C. Braspenning, J.M. van de Mortel-Fronczak, J.E. Rooda, 2006, A Model-based Integration and Testing Method to Reduce System Development Effort, Electronic Notes in Theoretical Computer Science, 164(4), 2006, pp. 13-28
9. Luna S., Lopes A., Yan See Tao H., Zapata F., Pineda R., Integration, Verification, Validation, Test, and Evaluation (IVVT&E) Framework for System of Systems (SoS). Procedia Computer Science, 20, 2013, pp. 298-305
10. L.C. van Ruijven, Ontology for Systems Engineering, Procedia Computer Science, 16, 2013, pp. 383-392
11. BKCASE Project, System Engineering Book of Knowledge, SEBoK v1.2, http://www.sebokwiki.org/ (last visit 2013-04), 2013
12. OMG, Software & Systems Process Engineering Meta-Model Specification, V2.0, 2008
13. MADS MOSAR, Gestion des risques : Méthode MADS-MOSAR II Manuel de mise en oeuvre, Pierre Périlhon, 2007 [in French]
14. Forsberg, K., H. Mooz, H. Cotterman. 2005. Visualizing Project Management, 3rd Ed. Hoboken, NJ: J. Wiley & Sons.
15. Project Management Institute, A Guide to the Project Management Body of Knowledge (PMBOK® Guide)—Fifth Edition, 2013. ISBN13: 9781935589679
16. B. Combemale, X. Cregut, P.-L. Garoche, X. Thirioux, and F. Vernadat, “A Property-Driven Approach to Formal Verification of Process Models,” in *Enterprise Information Systems*, vol. 12, Eds. Springer Berlin Heidelberg, 2009, pp. 286–300.
17. V. Chapurlat, UPSL-SE: A model verification framework for Systems Engineering, *Comput. Ind.*, vol. 64, no. 5, pp. 581–597, 2013
18. W.Schamai, P.Fritzson, C.Paredis, A.Po, Towards Unified System Modeling and Simulation with ModelicaML: Modeling of Executable Behavior Using Graphical Notations, Proceedings 7th Modelica Conference, Como, Italy, Sep. 20-22, 2009
19. System Modelling Language SysML (http://www.sysml.org/), 2010
20. MODELICA - A Unified Object-Oriented Language for Physical Systems Modeling Language Specification Version 3.2 (see http://www.modelica.org/), March 24th, 2010